

UNIVERSITÄT BREMEN  
Fachbereich 3  
Computer Graphics and Virtual Reality

Doktorarbeit

---

# Immersive Medical VR Training Simulators with Haptic Feedback

---

MAXIMILIAN KALUSCHKE

Datum: 6. Mai 2024

**Erstprüfer:** Prof. Dr. Gabriel Zachmann

**Zweitprüfer:** Prof. Dr. Miguel A. Otaduy

**Kolloquium:** 11.07.2024

Copyright © 2024 by *Maximilian Kaluschke*

All rights reserved.

E-mail: mxkl@uni-bremen.de, maxi.kalu@gmail.com

Website: <https://cgvr.informatik.uni-bremen.de/research/dentalsim>  
<https://cgvr.informatik.uni-bremen.de/research/hips>

This document was generated using X<sub>Y</sub>TEX (PDF version 1.7).

The text and equations are typeset in 11pt Garamond.

## ABSTRACT

---

Virtual reality and haptic feedback technologies are at the forefront of a revolution in medical training, particularly within the realms of orthopedic and dental surgery. These new technologies enable the creation of virtual simulators that offer a risk-free environment for medical trainees to hone their skills, presenting a solution to the ethical dilemmas associated with traditional patient-based training methods. The challenge, however, lies in crafting simulators that are not only immersive, but also realistic and effective in transferring learned skills from virtual to real-world scenarios.

This dissertation unveils a groundbreaking modular system for virtual reality (VR)-based, haptic-enabled physics simulation designed to meet these challenges. It integrates continuous, realistic 6 degrees-of-freedom force feedback with material removal capabilities, ensuring an authentic interaction with virtual anatomical structures and tools. Central to this system are novel algorithms for collision detection, force rendering, and a volumetric representation, which collectively elevate the realism and performance of VR haptic simulators to unprecedented levels. These algorithms were implemented in a library that can be used with a variety of game-engines, haptic devices, and virtual tools.

Practical applications of this novel simulation library are demonstrated through the development of two state-of-the-art medical training simulators: one for total hip arthroplasty and another for dental procedures, focusing on root canal treatment and caries removal. These simulators are enhanced with innovative features, including automated VR registration, sound synthesis based on material removal, VR zoom capabilities, and accurate eye tracking, all of which contribute to a more immersive training experience. Through rigorous testing, these simulators have been shown to have a significant positive impact on learning, enabling students to seamlessly transfer skills learned in a virtual setting to real-life medical procedures.

The hip surgery simulator was used by 18 experts in orthopedics, collectively rating the current state of the system as being useful and recommended the application for medical students and even residents to train with. The dental simulator was evaluated twice with a larger group of dental students (40 and 30), whose real-world skill was assessed before and after training by independent experts. The results revealed a significant learning effect on their actual skill, due to the training in my VR simulator. The first study also provided insightful data on how specific technological aspects of VR in head-mounted displays influence learning effectiveness and the transferability of training. My findings indicate that hand-tool alignment and stereopsis are critical in improving learning outcomes and useability of skills. Additionally, the second study has offered new perspectives on dental students' training behaviors, especially regarding the use of indirect vision and its correlation with performance and learning. This was also made possible by the discovery of novel metrics for indirect vision in complex bi-manual tasks.

By pushing the boundaries of VR and haptic technology in medical training, this dissertation contributes significantly to the research field, offering advanced tools that promise to transform how medical professionals are trained. The developed simulators not only make training more efficient and effective but also open up new avenues for research in medical education, ultimately aiming to enhance patient care and treatment outcomes.

## ZUSAMMENFASSUNG

---

Virtuelle Realität und haptisches Feedback sind Technologien die medizinischen Ausbildung, insbesondere in den Bereichen der Orthopädie und Zahnmedizin, revolutionieren werden. Diese neuartigen Technologien ermöglichen die Erschaffung virtueller Trainingssimulatoren, welche eine risikofreie Umgebung zu der Verbesserung der Fähigkeiten von medizinische Auszubildende. Damit präsentieren solche Simulatoren eine Lösung für die ethischen Dilemmas, die häufig mit patientenbasierten Ausbildungsmethoden verbunden sind. Die Herausforderung besteht jedoch darin, Simulatoren zu entwickeln, die nicht nur immersiv, sondern auch realistisch und effektiv in der Übertragung von gelernten Fähigkeiten von virtuellen auf reale Szenarien sind.

Diese Dissertation präsentiert ein bahnbrechendes modulares System für VR-basierte Physiksimulation mit haptischem Feedback, das entwickelt wurde, um diesen Herausforderungen zu begegnen. Es integriert kontinuierliche, realistische Kraftrückkopplung mit sechs Freiheitsgraden und Materialabtrag, welche eine authentische Interaktion mit virtuellen anatomischen Strukturen und Werkzeugen gewährleistet. Zentral für dieses System sind neuartige Algorithmen für Kollisionserkennung, Kraftrendering und eine volumetrische Darstellung, welche gemeinsam den Realismus und die Leistung von VR-haptischen Simulatoren auf ein beispielloses Niveau verhelfen. Diese Algorithmen wurden in eine Bibliothek implementiert, welche im Zusammenhang mit einer Vielzahl von Game-Engines, Haptikgeräten und virtuellen Werkzeugen verwendet werden kann.

Praktische Anwendungen dieser neuartigen Simulationsbibliothek wurden durch die Entwicklung von zwei modernen medizinischen Ausbildungssimulatoren demonstriert: einem für die totale Hüftendoprothese und einem anderen für zahnmedizinische Eingriffe, mit Fokus auf Wurzelbehandlung und Kariesentfernung. Diese Simulatoren werden durch weitere innovative Funktionen, wie automatischer VR-Registrierung, Klangsynthese basierend auf Materialentfernung, VR-Zoom und genauem Eye-Tracking, verbessert. Diese Zusatzfunktionen tragen zu einem noch immersiveren Trainingserlebnis bei. Bei rigorosen Tests haben diese Simulatoren einen signifikanten Lerneffekt gezeigt, der es den Studenten ermöglicht, die in einer virtuellen Umgebung erlernten Fähigkeiten nahtlos auf reale medizinische Verfahren zu übertragen.

Der Hüftchirurgiesimulator wurden von 18 Orthopäden erprobt, welche das System im aktuellen Zustand kollektiv als bereits nützlich bewerten. Außerdem gaben die Experten an, dass sie es Medizinstudenten und sogar Assistenzärzten empfehlen mit dem Simulator zu üben. Der Zahnmedizinsimulator wurde innerhalb von zwei Experimenten mit einer größeren Anzahl von Zahnmedizinstudenten (40 und 30) evaluiert. Die tatsächlichen Fähigkeiten der Zahnmedizinstudenten wurden dafür vor und nach dem Training durch unabhängige Experten beurteilt. Die Datenauswertung hat ergeben, dass eine signifikanter Lernfortschritt, gemessen an tatsächlichen Fähigkeiten, festgestellt wurde, welcher auf das Training mit meinem VR-Simulator zurückzuführen ist. Das erste Experiment lieferte außerdem aufschlussreiche Daten darüber, wie spezifische technologische Aspekte von VR in Head-Mounted Displays die Lerneffektivität und die Übertragbarkeit der Ausbildung beeinflussen. Meine Ergebnisse zeigen, dass die Überlagerung von realen und virtuellen Werkzeugen, sowie Stereoskopie, entscheidend dafür sind, um den Lerneffekt und die Nutzbarkeit von VR-Simulatoren zu verbessern. Darüber hinaus hat das zweite Experiment neue Perspektiven auf das Trainingsverhalten von Zahnmedizinstudenten eröffnet, insbesondere in Bezug auf die Nutzung von indirekter Sicht und deren Zusammenhang mit Leistung und Lerneffektivität. Diese Erkenntnisse wurden auch ermöglicht, durch die Konzeption von neuartigen Metriken für indirekte Sicht in komplexen, beidhändigen Tätigkeiten.

Durch das Erweitern der Möglichkeiten von VR- und haptischer Technologie in der medizinischen Ausbildung leistet diese Dissertation einen bedeutenden Beitrag zum Forschungsfeld und bietet fortschrittliche Werkzeuge, die ermöglichen, die Art und Weise, wie medizinische Fachkräfte ausgebildet werden, zu revolutionieren. Die entwickelten Simulatoren machen die Ausbildung nicht nur effizienter und effektiver, sondern eröffnen auch neue Forschungsfelder in der medizinischen Bildung, mit dem letztendlichen Ziel, die Patientenversorgung und Behandlungsergebnisse zu verbessern.



## ACKNOWLEDGEMENTS

---

This dissertation would not have been possible without the help of many people.

First and foremost, I want to thank my supervisor Prof. Dr. Gabriel Zachmann, for guiding me and teaching me the ways of science and research. He gave me the opportunity to do my Ph.D. and always took the time if I had something to talk about. He also managed to create a fruitful and pleasant working environment within our workgroup, despite all the challenges. I am also thankful for the vast amount of patience, as finishing this dissertation seemed to have taken me nearly forever.

I am also very thankful to all my colleagues at the CGVR lab, especially Dr. Rene Weller, for countless discussions that helped me tremendously. Even though my research focus did not overlap to a large degree with many of my colleagues, we still had fruitful discussions and they helped me grow, professionally and personally.

This thesis would also never have happened if it weren't for my dad's repeated encouragement to finish my doctorate and to occasionally think about my own interests, which I am thankful for. I also want to thank the rest of my family and my girlfriend for their love and support.



# CONTENTS

---

<b>1</b>	<b>The Need for Virtual Training</b>	<b>3</b>
1.1	My Contributions . . . . .	5
1.2	Research Context of My Publications . . . . .	7
1.3	My Publications . . . . .	8
<b>2</b>	<b>Related Work</b>	<b>11</b>
2.1	Kinaesthetic Haptic Rendering Methods . . . . .	11
2.2	VR-Based Haptic Training Simulators . . . . .	12
	<hr/>	
	<b>I METHODS</b>	<b>15</b>
	<hr/>	
<b>3</b>	<b>System Overview</b>	<b>17</b>
3.1	Haptics Connect Virtuality and Reality . . . . .	18
3.2	Physical Simulation . . . . .	18
3.3	Displaying the Simulation State . . . . .	18
3.4	Software Architecture Overview . . . . .	19
<b>4</b>	<b>Rigid-Body Simulation</b>	<b>25</b>
4.1	The Description of a Rigid Body . . . . .	25
4.2	Computing the Distribution of Mass using Inner Spheres . . . . .	26
4.3	The Integration of Motion . . . . .	26
<b>5</b>	<b>Collision Detection</b>	<b>31</b>
5.1	Enforcing Intersection-Free Operation . . . . .	32
5.2	Detecting Points of Contact . . . . .	36
5.3	Resolving Contacts . . . . .	42
<b>6</b>	<b>Modular Haptic Rendering</b>	<b>53</b>
6.1	Defining Coordinate Spaces and Transformations . . . . .	53
6.2	Automated Registration of Haptic- and VR-Space . . . . .	58
6.3	Mediating the Link Between User and Virtual Tool . . . . .	60
6.4	The Force Rendering Algorithm . . . . .	63
<b>7</b>	<b>Modeling 3D Materials</b>	<b>67</b>
7.1	Virtual Power Tools . . . . .	67
7.2	Defining A Local Cutting Model . . . . .	68
7.3	Representation of Removable Material . . . . .	70
7.4	Fast Continuous Removal of Material . . . . .	73
7.5	Variations on the Material Removal Algorithm . . . . .	74
7.6	The Visualization Pipeline . . . . .	77
	<hr/>	
	<b>II APPLICATION</b>	<b>91</b>
	<hr/>	
<b>8</b>	<b>Immersive Simulators</b>	<b>93</b>
8.1	Bi-Manual VR Dental Simulator . . . . .	93
8.2	Large Forces for Hip Surgery Training Simulator . . . . .	103

9	Evaluating Simulators in Practice	113
9.1	Teaching Dental Surgery . . . . .	113
9.2	Simulating Hip Surgery . . . . .	117
10	Conclusion	125
10.1	Future Work . . . . .	125

III APPENDIX

	Bibliography	129
A	Variation of Clustering Algorithm	137
B	Math Notation	139
	List of Figures	142
	List of Tables	144
	List of Code	146
	Acronyms	147

IV PUBLICATIONS

P1	A Volumetric Penetration Measure for 6-DOF Haptic Rendering of Streaming Point Clouds	150
P2	Immersive Anatomy Atlas — Empirical Study Investigating the Usability of a Virtual Reality Environment as a Learning Tool for Anatomy	158
P3	Realistic Haptic Feedback for Material Removal in Medical Simulations	168
P4	Review of Haptic Rendering Techniques for Hip Surgery Training	176
P5	A cadaver-based biomechanical model of acetabulum reaming for surgical virtual reality training simulators	188
P6	Immersive Anatomy Atlas: Learning Factual Medical Knowledge in a Virtual Reality Environment	202
P7	A Shared Haptic Virtual Environment for Dental Surgical Skill Training	212
P8	The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training	220
P9	Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study	228
P10	The effect of 3D stereopsis and hand-tool alignment on learning effectiveness and skill transfer of a VR-based simulator for dental training	246
P11	Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks	272



**OUTLINE**

1.1	My Contributions . . . . .	5
1.2	Research Context of My Publications . . . . .	7
1.3	My Publications . . . . .	8

## THE NEED FOR VIRTUAL TRAINING

THE need for orthopedic care is on the rise as life expectancy continues to increase globally. This is particularly evident in Germany, where the population is aging [The World Bank Group, 2024; Statistisches Bundesamt, 2022]. In 2002, the prevalence of arthritis was found to be 9% in people over the age of 20, 17% in 37 year-olds and over 90% in 65 year old people [Theiler, 2002]. Therefore, most people will be affected by arthritis during their life-time. Consequently, the number of total arthroplasties has seen a steady increase from 2005 to 2019 (see Figure 1.1) (with a drop in 2020 and 2021 due to the COVID-19 pandemic). In 2021, musculoskeletal procedures were by far the most often performed with over 4 million (see Figure 1.2), with similar trends in prior years. When looking at specific surgical procedures done, total hip and knee arthroplasties are among the most performed procedures. They make up the 4<sup>th</sup> and 8<sup>th</sup> most common procedures with around 233 k and 172 k performed in 2021 in Germany (see Figure 1.3). Given this trend and currently no plausible reasons why it would not continue, training of orthopedic surgeons is an important task that can improve the general population's health.

Hip replacement is incredibly important for elderly people, since mortality due to pulmonary embolism [Nauffal et al., 2012] and all-cause mortality [Dhingra et al., 2021] is significantly higher for people with recent immobilization. In fact, the mortality following a hip fracture is heavily influenced by the delay of hip replacement surgery, where a delay of just one week were significantly associated with higher mortality rates [Dhingra et al., 2021]. Therefore, higher efficiency in hip arthroplasties could save lives. In fact, hip arthroplasty is shown to be the most cost effective way to extend a person's life expectancy and quality of life after hip fracture [Parker et al., 1992]. Even when analyzing the cost from a health-care provider's point of view, a similar conclusion can be found. A recent systematic review shows that all cost effectiveness analyses find total hip arthroplasty to be a cost-effective intervention for hip osteoarthritis [Agarwal et al., 2021].

Unfortunately, hip arthroplasties are extremely difficult to learn and to perform correctly. Deviations of the prosthesis positioning negatively impact the procedures' success rate, with more possible dislocations, greater wear and reduced range of motion [Bosker et al., 2007]. It is important to note that the success of orthopedic surgery depends heavily on the surgeon's experience with the specific procedure. This relationship can be observed in a wide variety of procedures, such as tooth implantation [Sendyk et al., 2017; Mordechai et al., 2022] and total hip arthroplasties [Ceylan et al., 2020]. In fact, a recent systematic review found that most studies reported that among many kinds of measurements improved with surgeons' experience, such as shorter operative times and lower complication

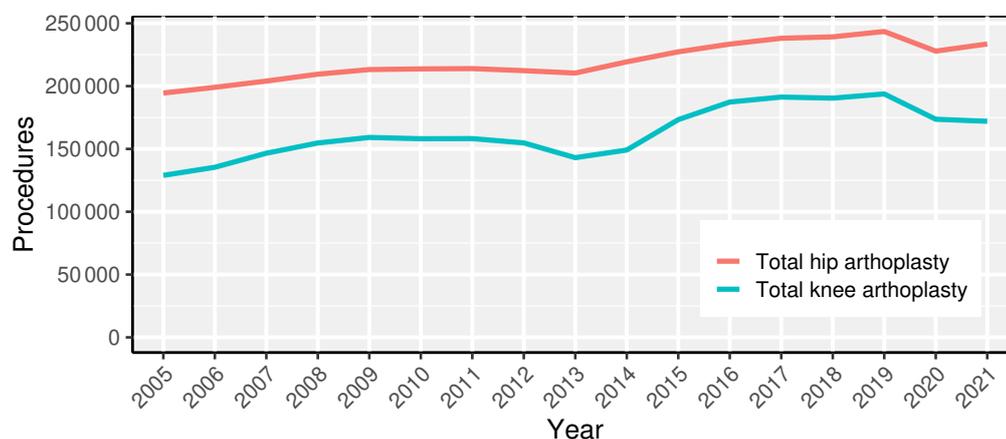
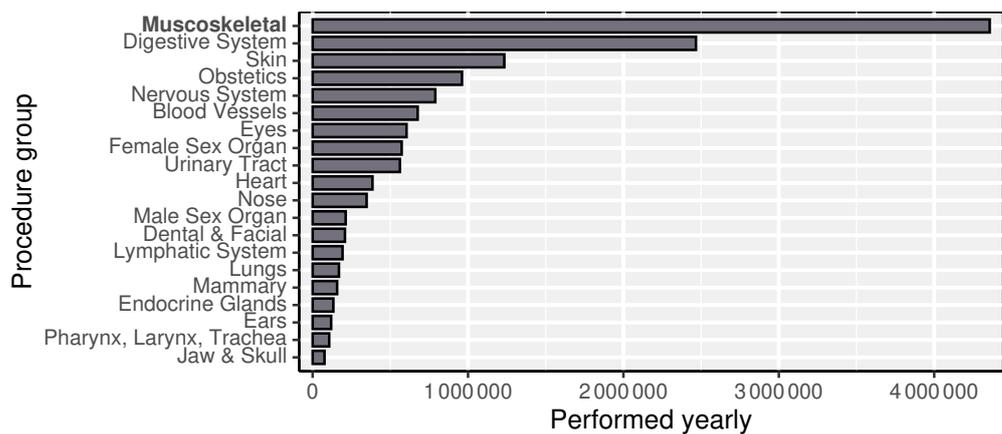


Figure 1.1: Orthopedic procedures over time. The amount of specific procedures performed in Germany, from 2005 to 2021 [Statistisches Bundesamt, 2023]. Orthopedic procedures, such as total hip and knee arthroplasty are steadily increasing in demand over time. Contrary to that teeth-related procedures did not increase in recent years.

Figure 1.2: Procedure frequencies, grouped by kind of surgery. The amount of surgical procedures performed, grouped by kind of surgery, in Germany during the year 2021 [Statistisches Bundesamt, 2023]. Musculoskeletal surgeries are by far the most common, those include orthopedic procedures such as knee and hip arthroplasties.



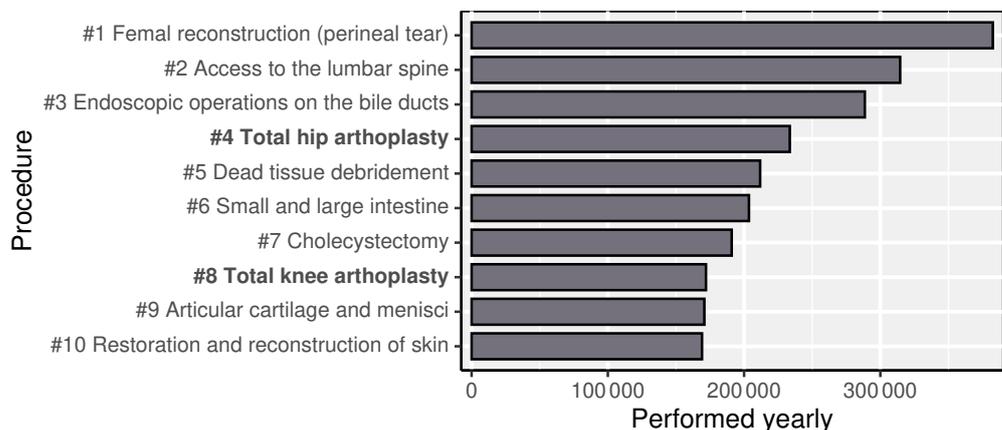
rates [Go et al., 2020]. Consequently, patients and the medical community are interested in increasing surgeons’ procedure-specific practice before they become responsible for surgical outcomes of patients.

The medical community has developed two main approaches for new surgeons to gain practice in surgeries:

1. Assisting a veteran surgeon’s in live procedure (real organs)<sup>1</sup>
2. Practice inside simulated environments (fake organs)
  - a) Real tools
    - i. Physical dummies with real tools
    - ii. Organ donations with real tools
  - b) Haptically rendered tools
    - i. Monitor-based computer setup
    - ii. VR-based computer setup

In first approach, new surgeons attend veteran surgeons’ operations, during which they start out simply observing the procedure. If they are lucky, there will be some verbal explanation by the veteran surgeon during the procedure, in case they have excess capacities. As the new surgeons’ experience builds up over multiple assisted procedures, they get more responsibilities over time, such as doing certain tasks under supervision of the veteran. The advantage is, the procedure is completely real and on live patients, therefore learned practical progress is guaranteed to translate to the surgeon’s expertise. However, the involvement of the new surgeon is very limited in the beginning and progress needs to be slowed down to ensure the patient’s safety.

Figure 1.3: Top 10 most common surgical procedures. The amount of surgical procedures performed, grouped by specific procedures, in Germany during the year 2021 [Statistisches Bundesamt, 2023]. Musculoskeletal surgeries such as hip and knee arthroplasties make up the 4<sup>th</sup> and 8<sup>th</sup> most common procedures.



The second approach consists of simulating a procedure, which lets surgeons practice in a simulated but safe environment, since no patient's health is put at risk. Traditionally, simulators consist of operating on physical dummies with real tools (approach 2(a)i). Even though using the real tools helps surgeons familiarize themselves with the tools and how they are handled, the other side of the interaction, the bone, is often approximated by plastics that exhibits inaccurate material properties, such as hardness, density and friction. Additionally, the material is only usable once, as an operation causes irreversible modification of the physical dummy. This waste of dummies incurs a high operational cost, which could be avoided by using reusable virtual simulators [Perry et al., 2015]. However, using physical dummies is still very popular in Dentistry in today. This is mostly because teeth dummies are incredibly small and therefore relatively inexpensive compared to larger bones such as pelvis and femur. However, even in dentistry, the operational costs of practicing root-canal access opening surgery is considerable, at around \$10 per plastic tooth. Depending on the requirements for realistic details such as internal anatomy with roots and accurate internal colors, these prices vary. Simple procedures such as caries removal requires only simple teeth models and are therefore relatively cheap to practice with plastic dummies. On the other hand, the real tools also experience wear when they are used, especially diamond burs for dentistry need to be exchanged regularly when they have been in use, adding on to the considerable operational costs. Approach 2(a)ii consists of simulating the bone part by a body parts of organ donors. This makes the approach very realistic, however the costs are much higher, as the amount of available real bones from organ donors is out of everyone's control, which makes them quiet rare.

Approaches 2b simulate both sides, bones as well as tools. Tools are typically rendered using a haptic device that can deliver kinaesthetic haptic feedback that can be experience as external forces that act upon the virtual tools based on virtual collisions and contacts. Due to the low operational costs, simulators can even be used for veteran surgeons to refresh their practical skills, if they have been on a long time-out<sup>2</sup>. In fact, there is also another considerable benefit to VR training simulators: they can be configured to provide trainees with a variety of cases to practice on [Badash et al., 2016]. In contrast, tooth model manufacturers have a limited amount of variations<sup>3</sup>.

A good hybrid approach to learning could consists of using simulations as the first learning experience for new surgeons, and to do accelerated traditional observing and assisting during live operations after a relatively short time learning on simulations. The time duration could be initially chosen based on statistical analysis of a learning plateau, based on in-simulator performance metrics<sup>4</sup>, and later on tuned by experts' feedback. This way, we accelerate the new surgeons' learning progress in the early stages of learning, as the progress in a simulator is going to be faster than simply observing during a procedure, which also takes time away from other, more advanced surgeons' learning opportunity. Additionally, the time spent observing during procedures could be minimized and the new surgeon can take on the roll of assisting earlier than before.

## 1.1 MY CONTRIBUTIONS

The core component of virtual simulators with haptic feedback is the force rendering algorithm, which computes the forces that attached haptic devices should display to the user. As the user is simultaneously exerting external forces on the haptic devices, the user and haptic devices are linked in a tight closed feedback loop. The force rendering algorithm should, *at all times*, communicate *realistic*, *reliable* and *overlap-averse* forces to each haptic device. To more precisely define the desired properties:

- *at all times*: Ideally, any change in the user input leads to an immediate proportional output change. However in reality, the algorithm should minimize the delay of the

<sup>2</sup>Orthopedic surgeons told us that they have to relearn how to perform procedures if they haven't done them in a long time, for example due to an extended sickness or if they were working in a different position.

<sup>3</sup>In fact, most manufacturers sell only a single variant (e.g. Acidental, Nissin Dental Products INC. and Kilgore International, Inc.. All manufacturers that I checked only had single variants of a specific tooth model for endodontics, likely because they are already difficult to produce (costs range between \$10–35).

<sup>4</sup>Our root-canal access opening simulator showed a learning plateau after 3-4 trials (each running around 10-30 minutes).

output change, typically 1 ms is considered sufficiently fast.

- *realistic*: The computed forces create tool-obstacle interactions, that users perceive as physically plausible.
- *reliable*: The computed forces change continuously and do not contain artifacts, such as uncontrolled vibration, pop-through or sticking.
- *overlap-averse*: The computed forces minimize the overlap between tools and obstacles. The algorithm is able to produce torques that a 6 degrees-of-freedom (DOF) output device can render to resolve contacts far from the center of mass by rotations.

Consequently, the force rendering algorithm has high demands for the quality of the forces, while also having strict requirements on the computation time.

In this thesis, I will present a force rendering algorithm that meets these tight requirements for arbitrary watertight solid objects. The algorithm computes forces and torques that are physically plausible, which was formally evaluated over three user studies with over 80 medical experts, as well as informally by many people.

The force rendering algorithm supports material removal at the same haptic frequency, which is parametrized precisely by the state of the simulation at that time. More precisely, the material removal adapts to the tool contact configuration and external user force. Additionally, turbulent forces and torques that emerge during hip reaming from fast frictional collisions of the reamer and the pelvis can be simulated. Additionally, a sound algorithm produces convincing drilling sounds that emerge during real material removal during dental procedures, also based on the simulation configuration.

The algorithm operates at a very high frequency of 1.6–3 kHz, even in challenging contact situations with hundreds of simultaneous contacts, including material removal at the same rate.

The reliability was demonstrated during several demos and user-studies on various haptic devices, including a KUKA iiwa with 137 N and a Haption Virtuouse with 70 N, rendered with a high stiffness of  $12\,000 \frac{\text{N}}{\text{m}}$ , which is the device's maximal stiffness; the angular stiffness was set to 24 Nm/rad, though it could even be increased without becoming unstable. Haptic rendering on these high-force and high-stiffness devices is challenging and shows the reliability of the force rendering algorithm. The experiments so far have shown that the haptic rendering stiffness is only limited by the hardware and not further limited by the force rendering algorithm.

The obstacle surface can be visualized by an asynchronous visualization system that generates a triangle mesh based on the simulation's state at interactive rates of 3–10 Hz.

The algorithm was integrated in surgical simulators for orthopedic surgery and dental surgery. Both systems were formally evaluated by over 80 medical practitioners during three user studies, as well as informally by other experts. The dental surgery simulator was developed in collaboration with experts of Dentistry and evaluated by 70 dental students and 6 experienced dentists with overall positive feedback. Additionally, we have shown that the use of the simulator results in significant learn gains for students learning the root canal access opening (RCAO).

The orthopedic surgery that was simulation is the total hip arthroplasty, and was evaluated by 14 experts of orthopedics, with varying degrees of experience, including two heads of surgery. The experts overwhelmingly agreed that the simulator is a helpful tool to learn the hip replacement procedure and reported that they recommend practicing with the simulator to medical students and resident surgeons, and some even for attending surgeons.

In addition, my algorithm is modular by design, which makes it easy to incorporate different rigid-body simulation techniques. I have for example used penalty-, impulse- and position-based techniques in my system and could switch between them with ease.

## 1.2 RESEARCH CONTEXT OF MY PUBLICATIONS

I have started my research that is related to this document in the first half of 2017, when I began working on the government-funded research project “HIPS”. During this time, I have firstly published a conference paper [1] on penalty-based haptic rendering of streaming point clouds. The tool was represented by a spherepacking, whereas the environment was a set of points from a depth camera. At the same time, I have developed an immersive anatomy atlas that works with off-the-shelf VR equipment, based on which I have coauthored two journals papers: [2] in 2018, and [6] in 2021. Both evaluated the learning effectiveness of the anatomy atlas with students, showing improvement effectiveness compared to traditional anatomy learning with books.

For “HIPS” I have first developed a haptic rendering method that combines continuous feedback with continuous material removal, which was published as a conference paper in 2020 [3]. The biometric data that was auxiliary while developing this method was gathered during experiments, which were published in a journal paper in 2020 [5]. Simultaneously, I have used the same algorithms that I developed for “HIPS” to put together a VR dental surgery simulator. This simulator was used to construct 3 user studies, which I wrote about in 3 conference papers ([7], [8] and [11]) and an extended journal paper in 2023 [10]. The introduced method was well suited for simple tools, however, the follow-up project “DynamicHIPS”, which I also worked on, was dealing with more complicated tools. For example, the bone saw is used to cut off the femur head, during which one creates a slim slit, in which the blade is constrained, while only parts of the blade can actually cut. Thus, I improved my method by making the sphere properties more diverse and combining a rigid-body simulation with my continuous collision detection and material removal. These new algorithms have not been published yet, but they are described in this document. During the development of my improved algorithms, I have done a literature review, published as a conference paper in 2020 [4], which was given the best paper award. Additionally, I have conducted a user-study to find out the most ideal haptic rendering method for rendering during hip surgery. The results of the user-study were presented in a journal paper in 2023 [9], and guided me during the development of the new improved algorithms.

## 1.3 MY PUBLICATIONS

The following is a complete list of full papers (11), which I have published since I began working at the University of Bremen, divided into publications where I am first author (7), and those in which I am a coauthor (4), each in chronological order.

**First author**

- P1.** Kaluschke, M., Weller, R., and Zachmann, G. (2017). **A volumetric penetration measure for 6-DOF haptic rendering of streaming point clouds.** In *2017 IEEE World Haptics Conference (WHC)*, pages 511–516
- P3.** Kaluschke, M., Weller, R., Hammer, N., Pelliccia, L., Lorenz, M., and Zachmann, G. (2020). **Realistic Haptic Feedback for Material Removal in Medical Simulations.** In *2020 IEEE Haptics Symposium (HAPTICS)*, pages 920–926
- P7.** Kaluschke, M., Yin, M. S., Haddawy, P., Srimaneekarn, N., Saikaew, P., and Zachmann, G. (2021). **A Shared Haptic Virtual Environment for Dental Surgical Skill Training.** In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 347–352, Lisbon, Portugal. IEEE
- P8.** Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2022). **The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training.** In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, pages 449–455, Rochester, MN, USA. IEEE
- P9.** Lorenz, M., Hoffmann, A., Kaluschke, M., Ziadeh, T., Pillen, N., Kusserow, M., Perret, J., Knopp, S., Dettmann, A., Klimant, P., Zachmann, G., and Bullinger, A. C. (2023). **Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study.** *Scientific Reports*, 13(1):11230
- P10.** Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2023). **The Effect of 3D Stereopsis and Hand-Tool Alignment on Learning Effectiveness and Skill Transfer of a VR-based Simulator for Dental Training.** *PLOS ONE*, 18(10):e0291389
- P11.** Kaluschke, M., Weller, R., Yin, M. S., Hosp, B. W., Kulapichitr, F., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2024). **Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks.** In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*

**Coauthor**

- P2.** Weyhe, D., Uslar, V., Weyhe, F., Kaluschke, M., and Zachmann, G. (2018). **Immersive Anatomy Atlas — Empirical Study Investigating the Usability of a Virtual Reality Environment as a Learning Tool for Anatomy.** *Frontiers in surgery*, 5(73):73
- P4.** Ziadeh, T., Perret, J., Kaluschke, M., Knopp, S., and Lorenz, M. (2020). **Review of Haptic Rendering Techniques for Hip Surgery Training.** In *EuroVR 2020 Application, Exhibition & Demo Track: Proceedings of the Virtual EuroVR Conference*, number 381 in VTT Technology, pages 29–39. VTT Technical Research Centre of Finland
- P5.** Pelliccia, L., Lorenz, M., Heyde, C.-E., Kaluschke, M., Klimant, P., Knopp, S., Schleifenbaum, S., Rotsch, C., Weller, R., Werner, M., Zachmann, G., Zajonz, D., and Hammer, N. (2020). **A Cadaver-Based Biomechanical Model of Acetabulum Reaming for Surgical Virtual Reality Training Simulators.** *Scientific Reports*, 10(1):14545

- P6.** Gloy, K., Weyhe, P., Nerenz, E., Kaluschke, M., Uslar, V., Zachmann, G., and Weyhe, D. (2022). **Immersive Anatomy Atlas: Learning Factual Medical Knowledge in a Virtual Reality Environment.** *Anatomical Sciences Education*, 15(2):360–368

The complete manuscripts, including my individual contributions, can be found at the very end of this document (see [Publications](#)). If a chapter was heavily influenced by a publication, I will mention the specific publication on the left of a new chapter.

## RELATED PUBLICATIONS

<b>P4.</b>	<b>Review of Haptic Rendering Techniques for Hip Surgery Training . . . . .</b>	<b>176</b>
------------	---	------------

## OUTLINE

2.1	Kinaesthetic Haptic Rendering Methods . . . . .	11
2.1.1	Constraint-Based Methods . . . . .	11
2.1.2	Penalty-Based Methods . . . . .	11
2.1.3	Velocity- and Position-Based Methods . . . . .	12
2.2	VR-Based Haptic Training Simulators . . . . .	12
2.2.1	Dental Surgery Training . . . . .	13
2.2.2	Simulation of Orthopaedics . . . . .	14

## RELATED WORK

---

*“It weeps, it bleeds, and each new day a gash  
Is added to her  
wounds.”* — William Shakespeare

**H**APTIC rendering is the core technology that enables VR simulators to teach psychomotor skills. The way to derive the forces that are being rendered by the haptic device is an ongoing research topic for several decades. In fact, most haptics research focuses on novel active devices that can be worn or that are in other ways portable. Although those kinds of device can be more easily integrated into a VR scenario with walking, this is not my focus. Instead, I will present an overview of only desktop kinaesthetic haptic rendering, and how medical simulator have been built and used with those.

### 2.1 KINAESTHETIC HAPTIC RENDERING METHODS

The most basic idea is direct rendering, where simply all forces that act on the virtual tool during a simulation are sent to the haptic device. This approach has many issues, which led to the development of more sophisticated methods.

#### 2.1.1 *Constraint-Based Methods*

The first improved rendering algorithm was introduced in 1995: the god-object method by [Zilles and Salisbury, 1995]. The method keeps track of a second point (where the first is directly attached to the haptic device), which moves along the surface, constrained by the surface geometry. The surface position is derived by solving constraints imposed by the contacting triangles. This method assumed a point tool, which makes it only compatible with 3 DOF input. [Ortega et al., 2007] later extended the method to work with 6 DOF. However, the constraints are linearized here, and the simulation frequency is still much lower than haptic rates, which might result in less stable rendering at high stiffnesses. [Xu and Barbic, 2014] and [Ge Yu et al., 2015] later improved upon this method by solving constraints without linearization, although contact counts still need to be low to achieve haptic rates. These methods compute an intersection-free pose on the surface, which minimizes the difference to the haptic device pose.

[Chan et al., 2011] have used a volumetric representation for the static object (signed distance field (SDF)) with constraint-based contact solving. The tool is a collection of points on the tool surface (called point-shell). They also support material removal, though they do not provide any details on it. Looking at the thesis of [Chan, 2014], it shows that the cutting geometry of the virtual tool is limited to spheres. Moreover, the chosen object representations present problems for material removal simulation. A point-shell leaves the possibility to miss contacts due to holes, and the voxelized material removal will naturally behave discretized, whereas continuous removal is desirable.

#### 2.1.2 *Penalty-Based Methods*

There have also been simpler approaches, which do not need explicit constraint solving. Instead, rely on physically-based simulation, where the haptic interaction and contacts are formulated simply as another physical interaction inside the simulation.

[McNeely et al., 1999a] introduced a penalty-based method that simulates a point-shell tool, with static normals, in a static SDF environment (voxmap pointshell (VPS)). Contacts

result in penalty forces, which indirectly enforce soft constraints of contacts. [Arbabtafti et al., 2008] have implemented another VPS-based simulation, which additionally supports material removal. Their material removal assumes a sphere-tool, which will remove overlapping voxels in a discrete manner. Their simulation loops runs at only 100 Hz, which in combination with discrete material removal, results in noticeable discretization artifacts.

[McNeely et al., 1999a] introduced virtual coupling (VC), designed to improve the rendering stability. VC introduces a 6D virtual spring-damper between the real haptic device and the virtual tool. The forces relative to the tool are applied to it (thus moving it towards the device pose), and the forces relative to the device are rendered to it. The advantage is that the tool movement is now simulated, and thus, can be parameterized in all sorts of ways to keep the simulation stable. [Wan and McNeely, 2003a] later improved upon this method, by requiring only a 6D spring (without damper), resulting in less parameters that need tuning<sup>1</sup>. However, this method strictly works for voxels, as one parametrizes based on voxel dimensions, whereas VC is more general and could be used with any kind of simulation.

The two mentioned methods essentially decouple the haptic rendering from the simulation, which makes the implementation details of the simulation less important, as those forces are not directly rendered. In fact, a simulation needs not produce any forces to generate haptic feedback.

<sup>1</sup>Due to the coupling of simulation with real world (haptic device, human arm), where the latter are hard to predict, most developers revert to manual parameter tuning which can take a considerable amount of time. This is especially cumbersome if one has multiple scenarios (such as a set of virtual tools or haptic devices), in which case this time cost will multiply.

### 2.1.3 Velocity- and Position-Based Methods

This is exactly what impulse-based methods (originally introduced by [Mirtich and Canny, 1995]), and methods based on position-based dynamics (PBD) (introduced by [Müller et al., 2007]) make use of. Such approaches have been used for haptic rendering, e.g. [Wang et al., 2012] use an impulse-based simulation, based on the method by [Guendelman et al., 2003]. They compute the rendered force (although the rest of the simulation solves contacts on the velocity level) and material removal directly on the surface mesh. Although working directly on the surface mesh is likely done to not lose accuracy by using an approximation (such as voxels), it has the opposite effect. Since the polygon count that can be simulated in this way at haptic rates is several limited, one has to choose a very coarse mesh, leading to poor accuracy. Additionally, computing material removal on a surface mesh limits the possible removal geometry, i.e. no topology change to the original surface mesh can be performed.

Volumetric approaches do not suffer from these limitations. [Berndt et al., 2017] have used a particle-based representation with PBD-based simulation to implement soft-tissue cutting. The simulation results look graphically convincing, including some degree of physically-plausible deformation, without FEM simulation. However, the cutting process is not physically-based and happens instantaneous as objects overlap, which is not natural. The presented run-times are not fast enough for haptic simulation. Moreover, it is questionable how quality force feedback can be generated during instantaneous cutting.

## 2.2 VR-BASED HAPTIC TRAINING SIMULATORS

The utilization of VR technologies in training for medical applications has been explored with success across countless fields, such as laparoscopy [Alaker et al., 2016; Seymour, 2008], endoscopy [Ahmed et al., 2010], suturing and intestinal surgery [Satava, 2008], among others. Haptic feedback technologies have enabled the development of virtual simulators for the training complex psychomotor skills. Such precise motor skills and navigation by haptic sense are necessary in disciplines like orthopedics and more notably in dental education<sup>2</sup>, where the mastery of fine motor skills under indirect vision<sup>3</sup> is crucial.

<sup>2</sup>Again, many more fields employ, or could benefit from, haptic feedback in virtual training simulators.

<sup>3</sup>Looking at a reflection of an object while manipulating it is a difficult and foreign task to most people, thus, it has to be repeatedly practiced to become naturally integrated.

2.2.1 *Dental Surgery Training*

Consequently, there is a vast amount of VR-based training simulators for Dentistry, which have been presented in recent time. [Koopman et al., 2010] have developed the Simodont simulator, which has been shown multiple times to be an effective tool to help dental students surgical procedures. For example, [Eve et al., 2014] have shown that it is a feasible assessment tool, as they found significant performance differences for caries removal task between novices and experts, measured inside the simulator.

In research conducted by [Chu et al., 2023], seventy-two dental students were divided into control and experimental groups through random assignment. Both cohorts engaged in training sessions utilizing the Simodont system and participated in an initial assessment involving mirror operations. The experimental group utilized a training tool named Mirrosistant, while the control group adhered to conventional training methodologies, which included the use of dentognathic models, complete dentitions, a dental mirror, and manual instruction. The study revealed a preference among students for the traditional training method, which involves practicing with plastic teeth and a real handpiece, over the virtual simulation system. The researchers noted that the virtual system fell short in delivering a realistic experience of manipulation and failed to accurately represent the constrained operational space typical of oral procedures.

[Pohlenz et al., 2010] extended the Voxel-Man simulator for dental apicectomies (later extended for other types of sculpting tasks), and evaluated fifty-three students' subjective impression. The majority of participants (fifty-one) recommended training using virtual simulators as an additional modality in dental education.

[Kinoshita et al., 2016] developed a training simulator for dental implant surgery to let users experience the force magnitude necessary during osteotomy in posterior mandibular bone. The force is programmed based on 1D experimental data on the necessary force to advance, based on the current depth, as cortical bone is only present near the other end of the bone. All participants in the study could feel the cortical bone, and the following perforation as well, as the resistance force dropped to zero.

In a review on dental education using simulators by [Perry et al., 2015], the authors noted that one of the earliest VR simulators was DentSim [Buchanan, 2001], which blends the traditional mannequin-based training with 3D visuals, which has been shown multiple times to improve learning.

In a recent review on simulator use in dental education by [Li et al., 2021], the authors investigated nine available dental simulators (including Simodont, Voxel-Man and DentSim). They have found that none of the available simulators are immersive, i.e. none are using a head-mounted display (HMD) or other immersive display; most use a 2D display with 3D glasses, or 2D display with phantom head. It is unclear whether they output other sensory information, such as sound. Both of these issues are solved in my dental VR simulator. Currently available VR dental simulators are generally lacking, in following aspects, according to [Li et al., 2021]:

- Hardware-related issues
  1. Bad stereo vision or low resolution display
  2. No fixed physical finger rest
  3. Lack of bi-manual operation
- Software-related issues
  5. Force feedback is not realistic enough
  6. Simulation of soft tissue not realistic enough
  7. Training content is insufficient

## 8. Outcome not accurately quantified

My simulator can remedy hardware issues [Items 1 to 3](#) and software-issues [Items 5 and 8](#), and with further development issues [Items 6 and 7](#) could be rectified too.

2.2.2 *Simulation of Orthopaedics*

Another field that could benefit from VR training is orthopedics, related to hip and knee surgery. In this area, high forces need to be applied accurately, and under bad visibility, i.e. a surgeon's orientation is accomplished by haptic sensation, to a large extent. However, there is less research on the development and feasibility of such simulators, compared to dentistry. One reason for this, could be the challenge of delivering appropriately high forces, since orthopaedics has the highest mean force (210 N) compared to other surgeries [[Golahmadi et al., 2021](#)]. In fact, delivering  $> 50$  N is difficult, as most devices, even expensive ones, are limited in this regard. For example, a Phantom Premium High Force can deliver a maximum of 37.5 N. [[Hulin et al., 2008](#)] have presented such a haptic device, which is based on a KUKA LWR, which supports up to 150 N and 25 N m for haptic rendering. This system was later used by [[Sagardia and Hulin, 2016](#)] to test a novel penalty constraint-hybrid simulation method. The benchmark is not rigorous, which makes it unclear how well it works. Moreover, the deviation between unconstrained and constrained object seems to be limited. Similarly, the stiffness was limited to a maximum of  $4 \frac{\text{kN}}{\text{m}}$ .

Orthopaedic simulators instead often rely on low-force devices to deliver the haptic feedback. For example, [[Faieghi et al., 2020](#)] present a bone machining simulator that uses the VPS algorithm on a Phantom Omni (3 DOF, 3 N peak force). [[Wang et al., 2015](#)] use the same low-force device and same basic algorithm to simulate pedicle screw insertion surgery. [[Chan et al., 2016](#)] present a rendering method that uses the same VPS object representations, but optimizes a close surface pose that comply with all constraints; they did not mention what haptic device was used for development and evaluation, though their project web site mentions compatibility for the two 3 DOF low-force devices (Phantom Omni and Omega.6, 3 DOF and 12 N peak force).

Interestingly, despite the low fidelity force feedback that is generally used, a recent meta-analysis found that VR-based training results in better objective performance and faster time-to-completion, compared to traditional training methods [[Polce et al., 2020](#)]. Similarly, another meta-analysis by [[Aïm et al., 2016](#)] found that VR training leads to improvement in technical skills in orthopaedic surgery; however, the transfer from VR to reality was not investigated in nearly all included studies. It should be noted that most of the included studies focused on shoulder, knee surgery and other orthopaedic surgeries, which generally require less force than hip surgery<sup>4</sup>.

In conclusion, there are many simulators that are designed for surgical tasks, including dentistry and orthopaedics. However, the algorithms that are used to generate the force feedback is limited to the VPS algorithm and algorithms that are based on it. The haptic devices that are used often only support low force and 3 DOF. Immersive VR integration is generally not supported.

<sup>4</sup>During spine surgery, forces do not exceed 15 N [[Ortmaier et al., 2006](#)]; knee surgery stays below 50 N and 0.5 N m [[Chami et al., 2008](#)].

**Part I**

**Methods**

RELATED PUBLICATIONS

**P3.** Realistic Haptic Feedback for Material Removal in Medical Simulations . . . 168

OUTLINE

3.1 Haptics Connect Virtuality and Reality . . . . . 18  
3.2 Physical Simulation . . . . . 18  
3.3 Displaying the Simulation State . . . . . 18  
3.4 Software Architecture Overview . . . . . 19

## SYSTEM OVERVIEW

*“Look like the innocent flower, But be the serpent under’t.”*

— William Shakespeare

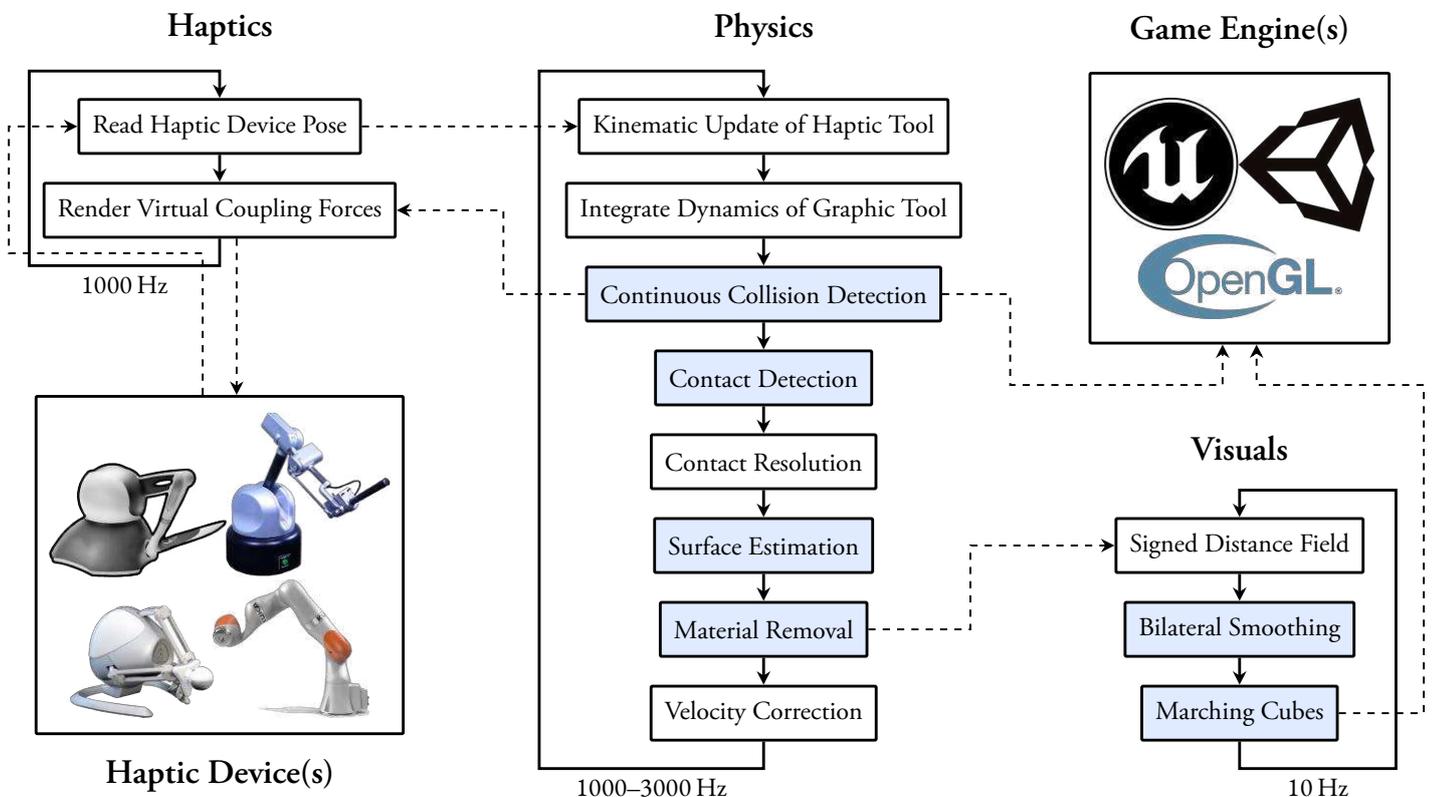
**F**ORCE rendering algorithms are often tightly linked to a large system of algorithms. In this chapter, I will give an overview of the individual algorithms that are linked with the force rendering algorithm and that make up a simulation system.

My simulation system consists of three separate, asynchronous threads:

1. **Haptics:** Handle the communication with the physical haptic device.
2. **Physics:** Simulate external forces acting on bodies through user interaction and collisions or contacts.<sup>1</sup>
3. **Visuals:** Keeps an up-to-date version of the modifiable obstacle (such as the pelvic bone or a tooth).

<sup>1</sup>The collision detection is sometimes performed in a fourth separate thread, in case the frequency regularly drops below 1 kHz.

In Figure 3.1 I have outlined the tasks that each thread has to perform repeatedly, and which thread frequency is expected for correct operation. Note that this is a simplified view and omits many important details of the algorithms. All important details will be given in separate chapters, such that readers could reimplement the whole system based on this document. I have implemented the presented the whole system from scratch in C++ and CUDA. It utilizes the inner sphere tree (IST) presented in [Weller and Zachmann, 2009a] as a form of volumetric representation of the involved bodies.



**Figure 3.1: System algorithmic overview.** The system consists of three asynchronous threads that operate at very different frequencies. Execution order of tasks is marked by solid lines. Inter-thread communication is marked by dotted lines. Blue colored tasks utilize the graphics processing unit (GPU).

## 3.1 HAPTICS CONNECT VIRTUALITY AND REALITY

The *Haptics* thread has the simple task of handling the communication with physical haptic device (details in Chapter 6). A haptic device typically outputs the haptic handle pose which it currently is in. It typically expects either a 3 or 6 DOF force as an input to render to the user by actuating the force on the haptic handle via motors. Depending on the haptic device that you use and its corresponding application programming interface (API), the haptic device might as well expect a haptic handle pose as an input to apply forces towards that position.

A virtual tool is attached at the haptic handle and can therefore be moved in space by the user by naturally moving the haptic device handle. The changes read by the *Haptics* thread and will control the virtual tool pose, either directly or more typically indirectly. Indirect tool control achieved here with virtual coupling [Colgate et al., 1995; McNeely et al., 1999a].

The basis for the force output is essentially computed in a separate thread. The task that this thread has to perform mostly boils down to a constant number of matrix multiplications per frame. Therefore, it is usually trivial to achieve a sufficient frequency of 1 kHz for the *Haptics*. This decouples the *Haptics* thread from the varying computational complexity of different contact situations that could happen during the physics simulation. The *Haptics* thread's work is designed to be constant and small, as the stability of the haptic device operation is tightly linked to the delay of appropriate feedback<sup>2</sup>.

<sup>2</sup>Note that the *Haptics* thread alone can only simulate a tool locked in a specific pose, as all the simulated movement is computed in the *Physics* thread. Therefore, poor performance of the *Physics* thread is similarly detrimental to the user experience.

## 3.2 PHYSICAL SIMULATION

The *Physics* thread is tasked with simulating the interactions of the virtual tool with (1) the force that the user is exerting and (2) the force that the virtual world imposes on the tool, such as from collisions and contacts with other bodies, sometimes other external forces such as gravity or air friction or noticeable inertia are desired.

The *Physics* considers two instances of the virtual tool, called *Haptic Tool* and *Graphic Tool*<sup>3</sup>. The *Haptic Tool* is moved through a kinematic translation and rotation to the virtual tool as if it is directly attached to the haptic device handle. Therefore, velocities are computed by backtracking positions and orientations in a small time-frame. Additionally, collisions of the *Haptic Tool* are not tracked as no external forces can act on it. Rather, it is a virtual representation of the real, physical haptic device handle, on which real physical forces already act, and the resulting translational and rotational changes are already tracked. As such, the *Haptic Tool* can be regarded as an auxiliary body and can be hidden from the user's view by simply not rendering a mesh in its place.

On the other hand, the *Graphic Tool* is a body with fully simulated dynamics. The user interaction is modeled through a torsional spring that is attached at the center of mass of the *Haptic Tool* and *Graphic Tool* (details in Section 6.3).

## 3.3 DISPLAYING THE SIMULATION STATE

The task of the *Visuals* thread is to translate the changes of a modifiable body in the form of a mesh that can be rendered by the game engine. To generate a mesh, the IST representation of the modifiable body is converted to a SDF by sampling (details in Section 7.6.1). Afterwards, the SDF is smoothed by a bilateral filter (details in Section 7.6.1.1). The smoothing step can be necessary to deliver a smooth and natural appearance, for example of human teeth, as the high-frequency bumps of the inner spheres is smoothed out. Afterwards, the marching cubes algorithm is used to generate a surface mesh, based on the smoothed SDF.

<sup>3</sup>I chose the terms *Haptic Tool* and *Graphic Tool*, which were introduced in [Wang et al., 2013], although there are also the terms *Haptic Interaction Point* and *Surface Object*, however I feel the former better foster the intuition for both *things* to be instances of the tool, with the same geometry but a different role in the physics simulation.

All three computation steps of the *Visuals* thread make use of spatio-temporal coherence to minimize the workload for consecutive computations. Additionally, steps 2 and 3 are computed on the GPU to accelerate them further. The achieved performance is sufficient to update the surface mesh at a frequency of 3–30 Hz depending on how many spheres are modified (see details of material removal in [Chapter 7](#)) per frame. The game engine is not bound by the frequency of the mesh updates, as the computations are asynchronous, just the updates to the computed vertices<sup>4</sup> are done from within the game thread and do not incur a significant slow-down.

<sup>4</sup>The surface mesh is described by vertices, face indices, as well as per vertex normals and colors.

### 3.4 SOFTWARE ARCHITECTURE OVERVIEW

Since the simulation is to be used in VR simulators, which are typically created using a modern game engine (GE), we require the simulation be controlled via a library that can be accessed from various GEs. Modern game-engines, such as the Unreal Engine (UE) 4 and 5, allows intuitive level design, flow control and VR rendering, which simplifies simulator and game development. Unreal Engine can natively drive visuals, sounds and controller input. Thereby, I extend its capabilities to also drive haptics with my library.

Many GEs allow the integration of third-party C++ libraries. UE and Godot allow the inclusion of plugins (or modules), which in turn can integrate external C++ code. Some game-engines, such as Unity 3D, do not allow C++ libraries to be called, but only allow functions that use plain C types. Since I want to support as many GEs as possible, I implemented my simulation library as a C++ library with C-style headers, although some parts are compiled using CUDA. In [Figure 3.2](#), I visualized the library architecture, including the UE plugin. The UE plugin contains only a small portion of the functionality and needs to be reimplemented for other GEs. The ISTHaptic library wraps all the functions in ISTHapticCUDA, such that the GE does not need to directly call CUDA code. At the time of writing, most GEs can not integrate binaries that were compiled by CUDA's `nvcc`. This phenomenon is not well documented, presumably because integrating CUDA code in a GE is not common practice.

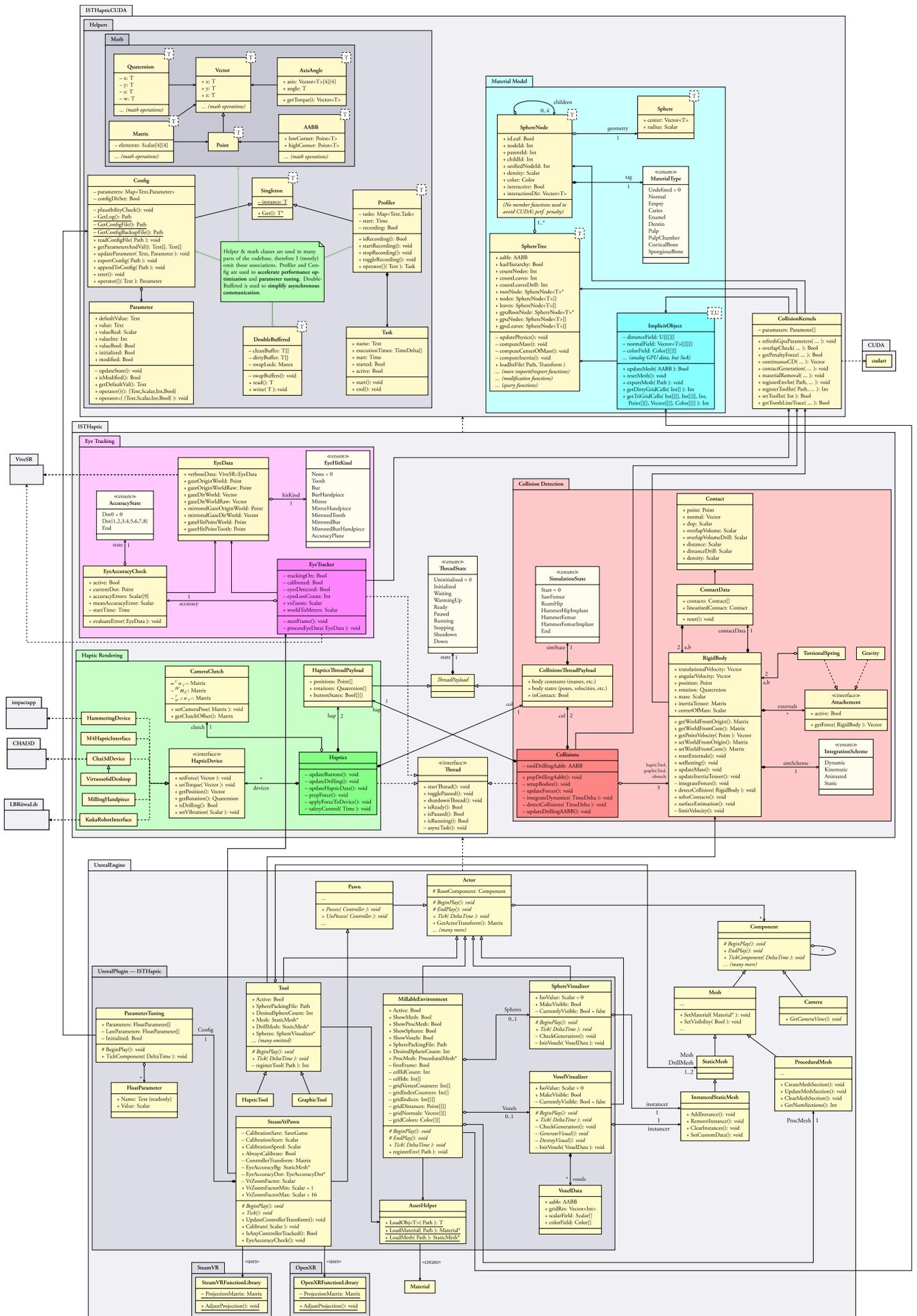
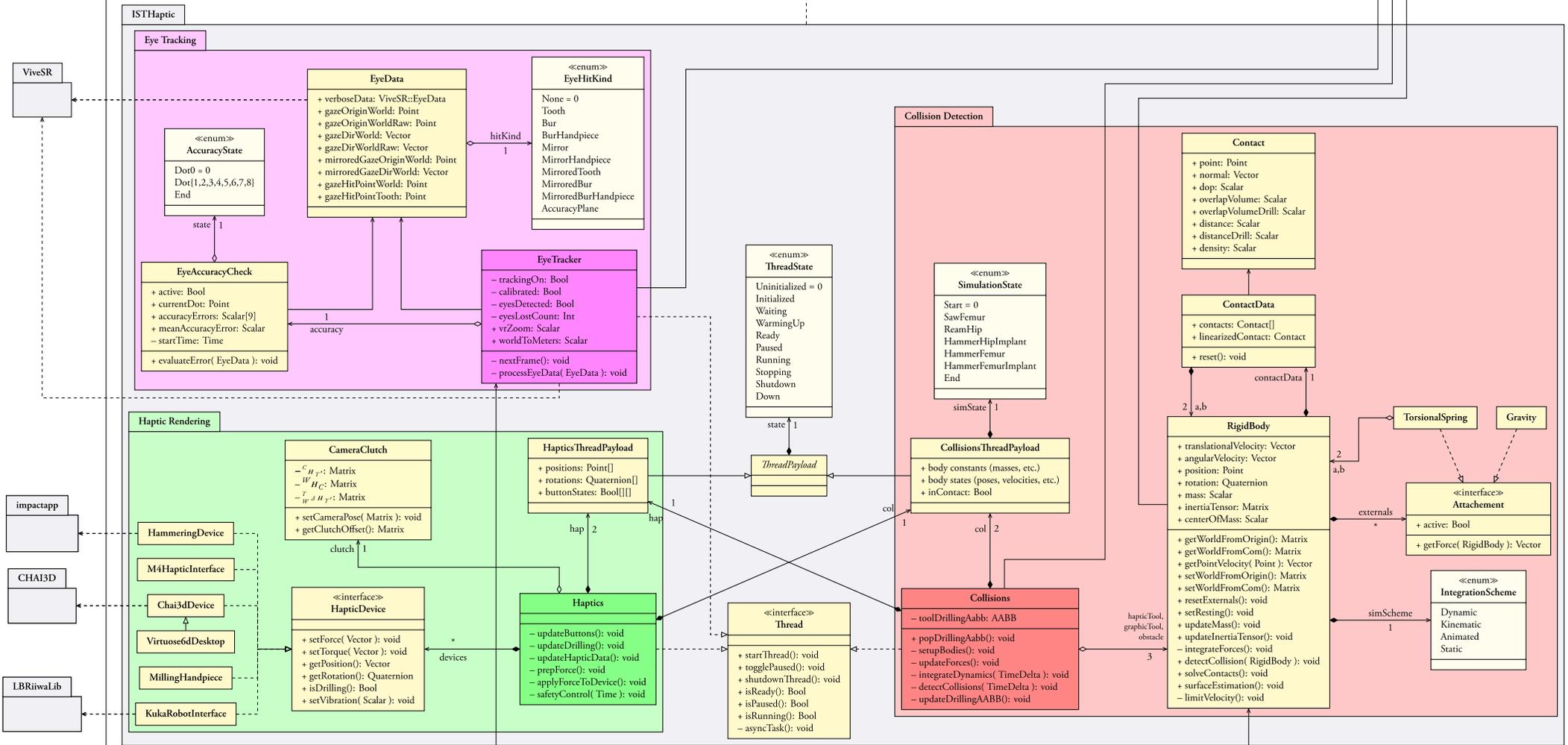
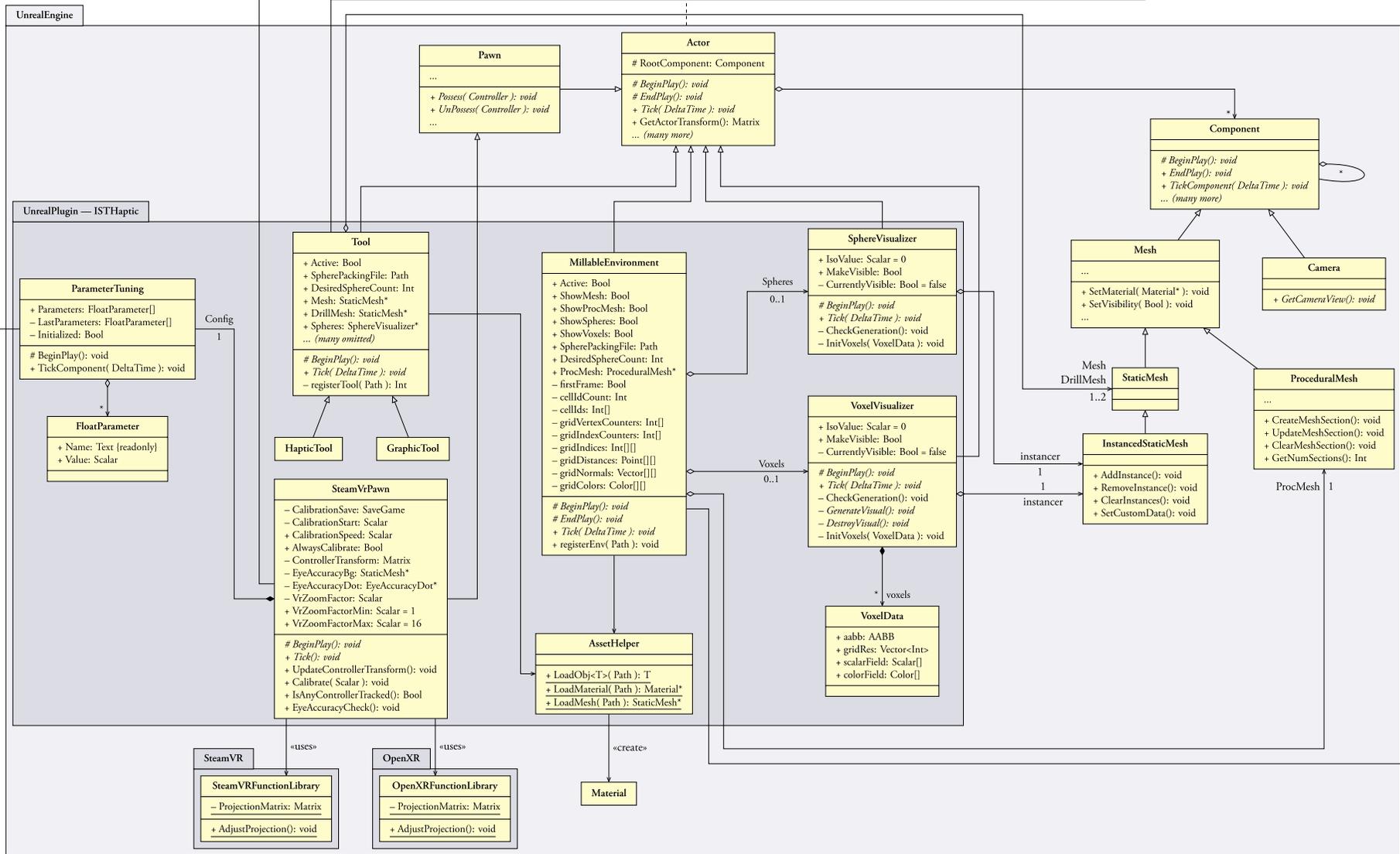


Figure 3.2: System class diagram. The system is implemented as a GE plugin that interfaces a library with two layers, where the second layer is written in CUDA to assign tasks to up to two GPUs. I have put each module on the following three pages on a larger scale to be readable on print.







## RELATED PUBLICATIONS

<b>P1.</b>	<b>A Volumetric Penetration Measure for 6-DOF Haptic Rendering of Streaming Point Clouds . . . . .</b>	<b>150</b>
<b>P3.</b>	<b>Realistic Haptic Feedback for Material Removal in Medical Simulations . .</b>	<b>168</b>
<b>P7.</b>	<b>A Shared Haptic Virtual Environment for Dental Surgical Skill Training . .</b>	<b>212</b>
<b>P8.</b>	<b>The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training . . . . .</b>	<b>220</b>

## OUTLINE

4.1	The Description of a Rigid Body . . . . .	25
4.2	Computing the Distribution of Mass using Inner Spheres . . . . .	26
4.3	The Integration of Motion . . . . .	26
4.3.1	Kinematic Motion of the Haptic Tool . . . . .	27
4.3.2	Dynamic Motion of the Graphic Tool . . . . .	27

## RIGID-BODY SIMULATION

“If you wish to make an apple pie from scratch, you must first invent the universe.”  
— Carl Sagan

POSITIONAL changes are needed to move objects through space. In this system, these positional changes are computed through the integration of external forces. Those forces change the velocity of objects, which are then integrated to yield the positional changes. Therefore, this part of the system describes a classical physically-based simulation with rigid bodies. The rigid-body simulation is a core part of the whole system, and it is tightly connected to the haptic rendering (see Chapter 6) by receiving forces (these forces act on the *Graphic Tool*) and sending forces (these forces act on the *Haptic Tool*). In this chapter, I will describe how the evolution of forces in my simulation is implemented.

## 4.1 THE DESCRIPTION OF A RIGID BODY

A rigid body  $B$  has a set of *constant* properties that are precomputed once, defined in *model* space and that characterize its nature. Another set of properties are dynamic, which characterize the body’s state in the *world*<sup>1</sup>.

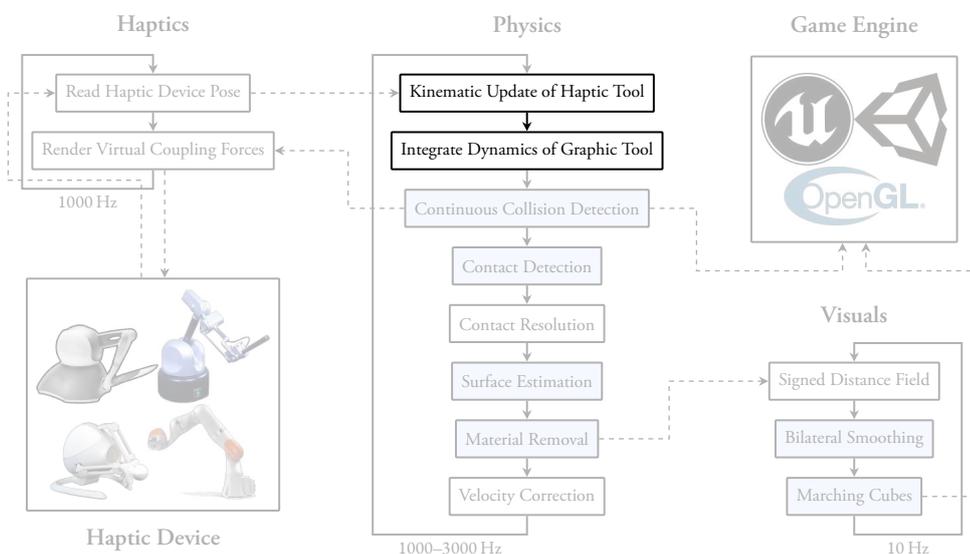
## Constant Properties

- $c(B) \in \mathbb{R}^3$ : position of the body’s center of mass [in *model*]
- $I(B) \in \mathbb{R}^{3 \times 3}$ : moment of inertia [in *model*]
- $m(B) \in \mathbb{R}$ : mass of the body [in *model*]

## State Variables

- $p(B) \in \mathbb{R}^3$ : position of the body’s center of mass [in *world*]
- $q(B) \in \mathbb{R}^4$ : orientation of the body [in *world*]
- $\vec{v}(B) \in \mathbb{R}^3$ : translational velocity of the body [in *world*]
- $\vec{\omega}(B) \in \mathbb{R}^3$ : rotational velocity of the body [in *world*]

<sup>1</sup>During implementation you most likely will have various auxiliary state variables, but to name all possibilities here would be excessive, as it depends on how you implement the simulation.



**Figure 4.1: Integration of rigid-body simulation.** Visualization of the scope of this chapter and how it integrates into the whole system. The rigid-body simulation provides the changes of position and orientation in physically-plausible way, assuming no collisions will occur. Integrating the rigid-body simulation in a common loop with the collision detection allows the use of physical units, such as velocities and forces. On the contrary, only considering positions and momentary velocities would reduce the options for realistic intervention during contact resolution (see Section 5.3).

- $\vec{f}(B) \in \mathbb{R}^3$ : external force acting on the body [in *world*]
- $\vec{\tau}(B) \in \mathbb{R}^3$ : external torque acting on the body [in *world*]
- $\vec{a}(B) \in \mathbb{R}^3$ : translational acceleration of the body [in *world*]
- $\vec{\alpha}(B) \in \mathbb{R}^3$ : angular acceleration of the body [in *world*]

#### 4.2 COMPUTING THE DISTRIBUTION OF MASS USING INNER SPHERES

In this section, I will describe how to compute the center of mass and moment of inertia. These measures are constant for rigid bodies, therefore we need to only pre-compute them once and can regard them as constants. The sphere packings simplify the computation of the mass constants. Given a 3D density mapping  $\rho : \mathbb{R}^3 \mapsto \mathbb{R}$ , we can determine the material density  $\rho(s_i)$  for all spheres  $s_i \in S$ , where  $S$  is the set of spheres that approximates the rigid body  $B$  inner volume, we can define the mass of a sphere and the body's center of mass  $c(B)$ :

$$m(s_i) = \frac{4}{3} \pi r(s_i)^3 \rho(s_i) \quad (4.1)$$

$$c(B) = \frac{\sum_{s_i \in S} m(s_i) c(s_i)}{\sum_{s_i \in S} m(s_i)} \quad (4.2)$$

The density mapping  $\rho$  can often be approximated simply by setting a uniform density, depending on the material<sup>2</sup>. The density should also be adjusted if the tool is significantly hollow, by factoring in the ratio of solid to hollow volume. In other cases, such as teeth or bone, the density distribution is non-uniform. However, even in those cases, the space can usually be subdivided such that there are distinct parts that are each uniformly dense, or a 3D function to determine the local density can be employed and stored per sphere (more details in [Section 7.3](#)).

For the application of torques, we need to compute the moment of inertia  $I(B) \in \mathbb{R}^{3 \times 3}$ , so that we can integrate angular acceleration to get angular velocity and subsequent orientation changes. Based on Steiner's theorem, we can compute the inertia tensor for the individual spheres  $I(s_i)$  and combine them to get the total moment of inertia  $I(B)$ .

$$I(s_i) = \begin{pmatrix} \frac{2}{5} m(s_i) r(s_i)^2 & 0 & 0 \\ 0 & \frac{2}{5} m(s_i) r(s_i)^2 & 0 \\ 0 & 0 & \frac{2}{5} m(s_i) r(s_i)^2 \end{pmatrix} \quad (4.3)$$

$$I(B) = \sum_{s_i \in S} I(s_i) + m(s_i) \begin{pmatrix} \vec{r}_y^2 + \vec{r}_z^2 & -\vec{r}_x \vec{r}_y & -\vec{r}_x \vec{r}_z \\ -\vec{r}_x \vec{r}_y & \vec{r}_x^2 + \vec{r}_z^2 & -\vec{r}_y \vec{r}_z \\ -\vec{r}_x \vec{r}_z & -\vec{r}_y \vec{r}_z & \vec{r}_x^2 + \vec{r}_y^2 \end{pmatrix} \quad (4.4)$$

with

$$\vec{r} = c(B) - c(s_i)$$

#### 4.3 THE INTEGRATION OF MOTION

In this section I will explain how the movement of the virtual tools – *Haptic Tool* and *Graphic Tool* – is implemented. The two instances are handled in different manners, as only the *Graphic Tool* is a regular body with dynamics. On the other hand, the motion of the *Haptic Tool* is completely controlled by the haptic device arm, as reality will ultimately

<sup>2</sup>For example, tools that are mostly made up of solid steel would receive a uniform density of 7.9 g/cm<sup>3</sup>, aluminum 2.7 g/cm<sup>3</sup> [Young et al., 2018]. However, the chosen density need not be physically accurate, and can be chosen to maximize stability or to achieve the desired feeling inertia. For example, I set the mass of the acetabular reamer to 75 g, which is less than 5 % of its real mass. Force stiffnesses and saturation thresholds need to be adjusted accordingly.

determine its position, not my simulation. Thus, I only observe its motion and adjust the state of my simulation accordingly. In theory, there could be other dynamic bodies in the simulation, besides the *Graphic Tool*. However, so far, this has not been necessary for my use-cases.

#### 4.3.1 Kinematic Motion of the Haptic Tool

Firstly, the simulation of the *Haptic Tool* (abbreviated by  $T_H$ ) is different from regular simulation objects, as it I do not control it directly. That means its movement arises from direct positional changes (based on the physical location of the haptic arm) and its velocities are derived from backtracking of the positional changes. Therefore, the motion is easily derived by the transformation matrix that is described in Equation (6.11).

Every time a kinematic update is performed, the velocity of the *Haptic Tool* is updated. This is done by keeping a history of the last  $N_{pos}$  poses  ${}^W\mathbf{H}_{T_H}(i)$  with corresponding time-stamps  $b_i$

$$\left\{ \left( {}^W\mathbf{H}_{T_H}(i), b_i \right) \mid {}^W\mathbf{H}_{T_H}(i) \in \mathbb{R}^{4 \times 4}, b_i \in \mathbb{R}, i \in \{0, 1, \dots, N_{pos} - 1\} \right\} \quad (4.5)$$

Given the current new pose  ${}^W\mathbf{H}_{T_H}(b)$  with the current time  $b$  and constant time-spans  $t_t = 10$  ms and  $t_r = 21$  ms<sup>3</sup>, we search backwards from the previous pose until we find a pose

$${}^W\mathbf{H}_{T_H}(n), b - b_n = \Delta b \leq t_t \quad (4.6)$$

with which we calculate the translational velocity based on the positional delta

$$\vec{v}(T_H) = \left( {}^W\mathbf{H}_{T_H} \mathbf{c}(T) - {}^W\mathbf{H}_{T_H}(n) \mathbf{c}(T) \right) \frac{1}{t_t} \quad (4.7)$$

We similarly search backwards until we find an pose

$${}^W\mathbf{H}_{T_H}^m = {}^W\mathbf{H}_{T_H}(m), b - b_m = \Delta b \leq t_r \quad (4.8)$$

to calculate the rotational delta to the current orientation

$${}^{T_H^m}\Delta\mathbf{H}_{T_H} = {}^W\mathbf{H}_{T_H}^m {}^W\mathbf{H}_{T_H}^{-1} \quad (4.9)$$

then we can compute the axis-angle equivalent rotation  $\vec{a}, \theta$  to  ${}^{T_H^m}\Delta\mathbf{H}_{T_H}$ . The angular velocity is then given by

$$\vec{\omega}(T_H) = \frac{\vec{a}}{|\vec{a}|} \theta \quad (4.10)$$

With this, the movement of the *Haptic Tool* is fully implemented through direct updates of the position and we always have the approximate velocity. External forces do not directly act on the *Haptic Tool*, as the user should be in control of its movement. Instead, external forces that would normally act on the tool act on the physical haptic handle, in other words, they are haptically rendered. Thereby, the user experiences an appropriate force that will steer the physical haptic handle (and thereby the *Haptic Tool*) towards the direction of the force. In that sense, one could regard this arrangement as indirect integration of the virtual forces. The only such external force that acts on the *Haptic Tool* comes from the interaction spring (details in Section 6.3).

#### 4.3.2 Dynamic Motion of the Graphic Tool

On the other hand, the *Graphic Tool* (abbreviated by  $T_G$ ) is a conventional rigid-body with full dynamics. Therefore, there is a set of sources of external force  $\mathcal{F}$  acting on the body  $T_G$ , in my implementation, these are:

<sup>3</sup>A simpler approach is to use the last two samples, but this is less reliable and robust, in my experience.

1. Interaction spring force  $\vec{f}_s, \vec{\tau}_s$  (details in Section 6.3)
2. Contact force  $\vec{f}_c, \vec{\tau}_c$  (details in Section 5.3)
3. Force of gravity  $\vec{f}_g = -\vec{n}_{up} m_T 9.8^4$

<sup>4</sup>Note that the  $\vec{n}_{up}$  vector is  $(0, 1, 0)^T = \hat{y}$  in Unity and  $(0, 0, 1)^T = \hat{z}$  in Unreal.

Most haptic rendering simulations will only need the forces from 1 and 2 to generate convincing haptic feedback from user interaction. Gravity is only included for the sake of completeness, it is disabled per default in my implementation, since it usually does not improve to the user experience, as it increases the felt air resistance. For development purposes of the simulation, it can be a useful force if it can be enabled and disabled at runtime (another developmental feature is the Camera Clutch from Section 6.1.6).

The integration loop has the following steps:

1. Transform the inertia tensor  $I(B)$  from *model* to *world* space, so that we can apply torques correctly later on

$$I(B)^{-1} \leftarrow R({}^W H_M) I(B)^{-1} R({}^W H_M)^T \quad (4.11)$$

2. Reset external forces that act on  $B$ . This is done since forces do not accumulate over time, so that the contribution of each source of external force can be newly evaluated.

$$\begin{aligned} \vec{f}(B) &\leftarrow \vec{0} \\ \vec{\tau}(B) &\leftarrow \vec{0} \end{aligned} \quad (4.12)$$

3. Linearized dynamic friction force. Given the contact force  $\vec{f}_c$ , linearized surface normal  $\vec{n}_c$  and linearized contact point  $\mathbf{p}(c)$  of the last frame, we compute friction forces  $\vec{f}_f$  and  $\vec{\tau}_f$ , based on the current tangential velocity  $\vec{v}_t$  at the contact point

$$\vec{r} = \mathbf{p}_c - \mathbf{c}(B) \quad (4.13)$$

$$\vec{v} = \vec{v}(B) + \vec{\omega}(T_G) \times \vec{r} \quad (4.14)$$

$$\vec{v}_t = \vec{v} - \vec{n}_c \quad (4.15)$$

$$\vec{f}_f = -|\vec{f}_c| \mu_d \vec{v}_t \quad (4.16)$$

$$\vec{\tau}_f = \vec{r} \times \vec{f}_f \quad (4.17)$$

$\vec{f}_f$  and  $\vec{\tau}_f$  are added to the external forces  $\vec{f}(B)$  and  $\vec{\tau}(B)$  that act on body  $B$ .

4. Iterate over all external sources of force  $F_i \in \mathcal{F}$  and evaluate their force contributions  $\vec{f}(F_i), \vec{\tau}(F_i)$  to this body (most notably the interaction spring (see Section 6.3))

$$\begin{aligned} \forall F_i \in \mathcal{F} : \vec{f}(B) &\leftarrow \vec{f}(B) + \vec{f}(F_i) \\ \forall F_i \in \mathcal{F} : \vec{\tau}(B) &\leftarrow \vec{\tau}(B) + \vec{\tau}(F_i) \end{aligned} \quad (4.18)$$

5. Calculate the accelerations  $\vec{a}(B), \vec{\alpha}(B)$  that result from the external forces  $\vec{f}(B), \vec{\tau}(B)$

$$\begin{aligned} \vec{a}(B) &= \frac{1}{m_B} \vec{f}(B) \\ \vec{\alpha}(B) &= I(B)^{-1} \vec{\tau}(B) \end{aligned} \quad (4.19)$$

6. Integrate accelerations  $\vec{a}(B), \vec{\alpha}(B)$  to yield changes in velocities  $\vec{v}(B), \vec{\omega}(B)$

$$\begin{aligned} \vec{v}(B) &\leftarrow \vec{v}(B) + \Delta t \vec{a}(B) \\ \vec{\omega}(B) &\leftarrow \vec{\omega}(B) + \Delta t \vec{\alpha}(B) \end{aligned} \quad (4.20)$$

7. Damp the velocities  $\vec{v}(B)$ ,  $\vec{\omega}(B)$ , to guarantee that the system dissipates energy.

$$\begin{aligned}\vec{v}(B) &\leftarrow \vec{v}(B) (1 - b_{d_t})^{\Delta t} \\ \vec{\omega}(B) &\leftarrow \vec{\omega}(B) (1 - b_{d_r})^{\Delta t}\end{aligned}\quad (4.21)$$

The damping parameters are  $b_{d_t}, b_{d_r} \in (0, 1)$ , which I default to  $0.99 = b_{d_t} = b_{d_r}$ .

8. Clamp the magnitude of velocities  $\vec{v}(B)$ ,  $\vec{\omega}(B)$  to increase stability

$$\begin{aligned}\vec{v}(B) &\leftarrow \begin{cases} \vec{v}(B) \frac{v_{max}}{|\vec{v}(B)|}, & v_{max} > 0 \wedge |\vec{v}(B)| > v_{max} \\ \vec{v}(B), & \text{otherwise} \end{cases} \\ \vec{\omega}(B) &\leftarrow \begin{cases} \vec{\omega}(B) \frac{\omega_{max}}{|\vec{\omega}(B)|}, & \omega_{max} > 0 \wedge |\vec{\omega}(B)| > \omega_{max} \\ \vec{\omega}(B), & \text{otherwise} \end{cases}\end{aligned}\quad (4.22)$$

The velocity thresholds  $v_{max}, \omega_{max}$  need to be chosen<sup>5</sup>, note that  $v_{max}$  depends on the unit of length of the *world* space<sup>6</sup>.

9. Integrate velocities  $\vec{v}(B)$ ,  $\vec{\omega}(B)$  to yield changes in position  $\mathbf{p}(B)$  and orientation  $\mathbf{q}(B)$ , which are for now stored in  $\mathbf{p}(B')$  and  $\mathbf{q}(B')$  as we need to perform collision detection before actually moving the body, as we intend to guarantee intersection-free positions at all times

$$\mathbf{p}(B') \leftarrow \mathbf{p}(B) + \Delta t \vec{v}(B) \quad (4.23)$$

$$\mathbf{q}(B') \leftarrow \mathbf{q}(B) + \Delta t \vec{\omega}(B) \frac{1}{2} \mathbf{q}(B) \quad (4.24)$$

then,  $\mathbf{q}(B')$  is normalized afterwards. Note that Equation (4.24) uses quaternion operations for addition and multiplication.

<sup>5</sup>The velocity thresholds should be chosen, such that they almost never get triggered during any close to normal movement through the air, so that they do not inhibit the fluidity of the interaction. The general rule should therefore be to start with low values and increase them until you no longer experience their influence during normal interaction.

<sup>6</sup>My default choices are  $\omega_{max} = 10 \frac{\text{rad}}{\text{s}}$  and  $v_{max} = 10 \frac{\text{m}}{\text{s}}$ .

## RELATED PUBLICATIONS

<b>P1.</b>	<b>A Volumetric Penetration Measure for 6-DOF Haptic Rendering of Streaming Point Clouds . . . . .</b>	<b>150</b>
<b>P3.</b>	<b>Realistic Haptic Feedback for Material Removal in Medical Simulations . .</b>	<b>168</b>
<b>P7.</b>	<b>A Shared Haptic Virtual Environment for Dental Surgical Skill Training . .</b>	<b>212</b>
<b>P8.</b>	<b>The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training . . . . .</b>	<b>220</b>

## OUTLINE

5.1	Enforcing Intersection-Free Operation . . . . .	32
5.1.1	Sample-Based Discrete Collision Detection . . . . .	33
5.1.2	Translational Continuous Collision Detection . . . . .	35
5.1.3	Maintaining Physical Laws Through Velocity Correction . . . . .	36
5.2	Detecting Points of Contact . . . . .	36
5.2.1	Estimating Surface Properties by Linearization . . . . .	39
5.2.2	Clustering Nearby Contacts . . . . .	40
5.3	Resolving Contacts . . . . .	42
5.3.1	Robust Penalty-Based Contact Resolution . . . . .	43
5.3.2	Impulse-Based Solving of Contact Constraints . . . . .	48
5.3.3	A Hybrid Solution for Contact Resolution . . . . .	51

## COLLISION DETECTION

*“Double, double toil and trouble; Fire burn, and cauldron bubble.”*

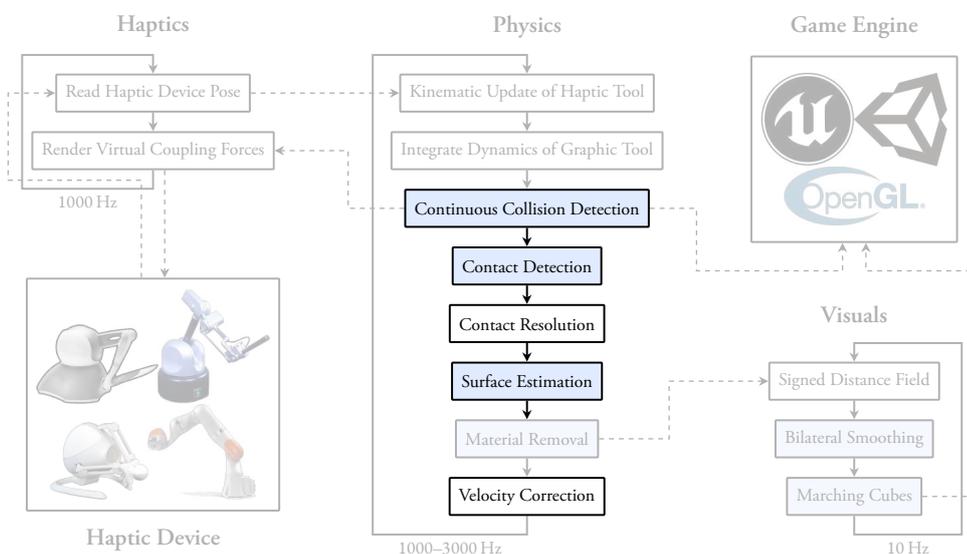
— William Shakespeare

THE simulation described so far generated all the movement of bodies by forces, with the assumption that they are known. However, how to compute these forces is yet to be explained<sup>1</sup>. Some forces, such as gravity or a single spring are easy to compute programmatically.

The goal of haptic rendering is, generally speaking, the computation of virtual forces that arise during interaction of virtual tools inside a virtual environment, which are then rendered as real forces to the user, coupling the users real hand with virtual tools. Therefore, the most important forces are those that arise from collisions and contact between individual bodies<sup>2</sup>. Unfortunately, the computation of such forces is rather computationally expensive, relative to other forces. In fact, collision detection is often the performance “bottleneck”, meaning the portion of code that takes the most time to complete. Therefore, the research on collision detection has brought forward many different body representations, algorithms and acceleration strategies to minimize these performance issues. This research area is vast and the different solutions have varying advantages and disadvantages, which make them more or less suitable for my use-case. The algorithms that I developed and present here were better suited to my use-case than any approaches that were previously introduced.

In this chapter, I will present the methods that I have developed to compute collisions and subsequent contact forces. Many of these methods already incorporate different performance optimization strategies. Performance of collision detection played an important role in the development of my simulation system, as it is also the most complex task in my system. To improve the performance I am utilizing the graphics card for computation during most algorithms. However, I believe most algorithms could also be implemented efficiently on the CPU, with more development. The parts of the system that will be covered in this chapter are shown in [Figure 5.1](#), where blue algorithms utilize the graphics card.

The collision detection procedure can be described in a meta algorithm that calls several sub-procedures (see [Algorithm 5.1](#)). The steps in the meta algorithm depend on the completion of the previous step, similar to a pipeline. In the following, I will present the individual steps in all the necessary detail to be reimplementable.



<sup>1</sup>The computation of the forces of a user interaction spring is explained on [Section 6.3](#)

<sup>2</sup>A naive implementation's complexity of collision detection growth exponentially with the number of surface features, as all surface features of a body needs to be checked against all surface features of another body

**Figure 5.1: The role of Collision Detection..** Visualization of the scope of this chapter and how it integrates into the whole system. Collision detection corrects implausible overlaps between bodies and generates contact forces. It is the essential component to enforce the physical laws that occur during contact of bodies. Without it, bodies could overlap and move through each other. It is typically computationally expensive, depending on object and scene complexity.

## 5.1 ENFORCING INTERSECTION-FREE OPERATION

The first important goal of collision detection is the avoidance of intersections between bodies. In my simulation, no bodies should intersect, for the following reasons:

1. Intersections are physically implausible and have a significant impact on the experienced degree of realism of the haptics [Lorenz et al., 2023].
2. Many known issues of penalty-based haptic rendering (instability, discontinuities, pop-through) can be prevented by ensuring bodies are intersection-free.
3. Material removal can be implemented based on proximity with high precision, if bodies do not intersect (see Chapter 7).
4. The volumetric penetration measure based on inner spheres is only valid if the bodies do not intersect

As such, it could be considered essential for my simulation to not have intersections between bodies. However, we will also compute contact forces (see Section 5.2 and Section 5.3), which are also designed to resolve intersections. Therefore one could argue that we do not need the complete guarantee of no overlap through continuous collision detection (CCD)<sup>3</sup>. In fact, the simulation also runs with CCD, though the before mentioned problems can then occur. And they will, though with varying degrees of effect and frequency. Depending on the use-case, that might be more tolerable than for haptic rendering with high forces and inexperienced users.

The problem that CCD solves is essentially having two different poses

$${}^W\mathbf{H}_B, {}^W\mathbf{H}_{B'} \in \mathbb{R}^{4 \times 4} \quad (5.1)$$

of a body  $B$  and you want to know the furthest “evolution” between these poses

$${}^W\mathbf{H}_{B_{\text{free}}} \in \mathbb{R}^{4 \times 4} \quad (5.2)$$

which can be performed while staying intersection-free.

In case of my simulation, we are checking between the current pose (which is guaranteed to be intersection-free) and the pose after a single integration step of the dynamics (see Section 4.3.2). Therefore, we are interested in finding the furthest the body can travel until it hits another body, while we have the lowest difference in the two poses.

---

**Algorithm 5.1** Collision Detection Meta
 

---

**Input:** New potential pose after integration

**Result:** Intersection-free pose and correct velocity & force acting upon bodies

```

procedure COLLISIONDETECTIONMETA
  procedure CCD
    | determine earliest point of collision during rigid-body motion
  procedure CONTACTDETECTION
    | generate all points of contact by small dilation
  procedure SURFACEESTIMATION
    | estimate the local surface properties by small dilation
  procedure CONTACTCLUSTERING
    | fast clustering of contact points (optional step)
  procedure CONTACTRESOLUTION
    | resolve found contacts by either penalty- or impulse-method
end procedure
  
```

---

<sup>3</sup>I did not thoroughly test my simulation without CCD, though I could notice material removal was inconsistent. Irratic rotations of the tool during drilling would increase the rate of material removal.

The “evolution” between two poses can be interpreted in different ways, the most intuitive evolution for me is the linear interpolation of translation and rotation independently. We define the delta transformation

$${}^{B'}\Delta\mathbf{H}_B = {}^W\mathbf{H}_{B'} {}^W\mathbf{H}_B^{-1} \quad (5.3)$$

and check for each sphere how far along the curve it can travel while remaining intersection-free. To detect the earliest point of contact between a sphere  $s_i$  on a curve and a static sphere  $s_j$ , one could simplify the problem by shifting the radii

$$r(s_j) \leftarrow r(s_j) + r(s_i) \quad (5.4)$$

$$r(s_i) \leftarrow 0 \quad (5.5)$$

Thereby,  $s_i$  becomes a point on a curve vs an enlarged sphere  $s_j$ . One could formulate a polynomial that describes the motion of  $s_i^t$  at time-point  $t$  by using linear interpolation for the translation and the Rodrigue’s formula for the rotational part

$$\vec{r} = \mathbf{c}(s_i^{t_0}) - \mathbf{c}(B) \quad (5.6)$$

$$\begin{aligned} \mathbf{c}(s_i^t) &= \cos(\theta t)\vec{r} + \sin(\theta t)(\vec{e} \times \vec{r}) + (1 - \cos(\theta t))(e \bullet \vec{r}) \\ &+ \mathbf{T}({}^{B'}\Delta\mathbf{H}_B)t + \mathbf{c}(B) \end{aligned} \quad (5.7)$$

where  $s_i^{t_0}$  is the sphere at the start of CCD and  $\vec{e}, \theta$  are the axis-angle equivalent rotation to  $\mathbf{R}({}^{B'}\Delta\mathbf{H}_B)$ <sup>4</sup>. Unfortunately, even solving this simplified formulation of the CCD analytically is too computationally expensive, compared to its benefits. However, an approximate numerical solution can be implemented by using a series of discrete collision checks, which I will explain in the next section.

### 5.1.1 Sample-Based Discrete Collision Detection

The basic idea is to interpolate the translational and rotational parts of the transformation independently to get the poses at intermediate time points, thereby substepping the motion. The presumption in that case is, that the object’s motion in between time steps can be described by a linear transformation that consists of only a translation and rotation, with constant velocity. I have outlined the substepping process in [Algorithm 5.2](#). One can either stop substepping just before a colliding pose, or stop at the colliding substep and further refine the pose nearest to the surface by bisection – the algorithm outlines the latter variant. Bisection would use the graphic tool pose of the previous frame as the outside case, and the colliding state as the inside case, approximate the root in a fixed amount of iterations  $N_2$  (for example  $N_2 = 10$ ).

The problem with this approach is that we now have to do many boolean overlap tests (e.g. 15, 5 for finding the bounds, 10 for finding the root), which is computationally expensive. This is especially true, when objects are in close proximity, without actually overlapping, since the bounding volume hierarchys (BVHs) can intersect deeply, but we can’t early exit due to having found an actual overlap. Moreover, this approach is best suited for computation on the CPU, as traversing two BVHs simultaneously is ill-suited for massive parallelization. Whereas traversing one BVH against many primitives is easy to parallelize, thus can be computed very fast on a GPU. Using this approach, while feasible in principle, would allocate a lot of computation time towards collision detection, as there are more steps in the pipeline to be done, many of which are accelerated using GPU computation. Putting one crucial task on the CPU (the CCD), while other are on the GPU, would require me to synchronize the spherepacking data between CPU and GPU, as it might be modified due

<sup>4</sup>In my experiments, it seems that an exact CCD is not essential for a convincing haptic rendering experience.

#### Implementation Caveat

Another notable problem of linear interpolation of the transformation is, that it cannot be reversly applied to the other body while producing the same path in space, meaning the found intersection for the reverse motion is generally not the correct result for the original problem. Whereas, this is possible for pure translation, if body  $A$  is moving by translation of  $\vec{d}$  towards a static body  $B$ , we can simply reverse the dynamic and static roles of the bodies by negating  $\vec{d}$ . However, the resulting interpolation factor  $t$ , that moves  $B$  along  $-\vec{d}$  until it collides with  $A$ , needs to be negated again to apply to the movement of  $A$  along  $\vec{d}$  with static  $B$ . This leads to a performance issue in my case, as the collision detection of the tool is accelerated by a BVH of spheres, whereas the obstacle is generally not. Therefore transforming the spheres of the obstacle into the tool frame (which needs to be done each frame) is computationally less expensive than transforming the all spheres of the tool, including hierarchical spheres.

**Algorithm 5.2** Sample-Based CCD

**Input:**  ${}^E\mathbf{H}_{T_G}, {}^E\mathbf{H}_{T_H} \in \mathbb{R}^{4 \times 4}$ 
**Result:**  ${}^E\mathbf{H}_{T'_G} \in \mathbb{R}^{4 \times 4}, c \in \{\text{true}, \text{false}\}$ 
**procedure** SAMPLECCD

**if** first frame **then**
 ${}^E\mathbf{H}_{T_G} \leftarrow {}^E\mathbf{H}_{T_H}$ 
 $c \leftarrow \text{false}$ 
**return**
 $\triangleright$  facilitate fast start-up

*// Prepare transform interpolation*
 ${}^{T_G}\mathbf{H}_{T_H} \leftarrow \mathbf{T}(-c) {}^E\mathbf{H}_{T_G}^{-1} {}^E\mathbf{H}_{T_H} \mathbf{T}(c)$ 
 $\vec{d} \leftarrow \text{EXTRACTTRANSLATION}({}^{T_G}\mathbf{H}_{T_H})$ 
 $\vec{a}, \theta \leftarrow \text{EXTRACTROTATION}({}^{T_G}\mathbf{H}_{T_H})$ 
 $t_T^{max} \leftarrow \|\vec{d}({}^{T_G}\mathbf{H}_{T_H})\|$ 
 $t_R^{max} \leftarrow \theta(\mathbf{R}({}^{T_G}\mathbf{H}_{T_H}))$ 
 $t_T^{mod} \leftarrow t_T^{step} / t_T^{max}{}^5$ 
 $t_R^{mod} \leftarrow t_R^{step} / t_R^{max}{}^6$ 
 $c \leftarrow \text{false}$ 
**for all**  $i \in [1, N]$  **do**<sup>7</sup>
 $t_{free} \leftarrow i/N$ 
 ${}^{T_G}\mathbf{H}_{T'_G} \leftarrow \mathbf{T}(t_{free} t_T^{mod}) \mathbf{R}(\vec{a}, t_{free} t_R^{mod})$ 
 ${}^E\mathbf{H}_{T'_G} \leftarrow {}^E\mathbf{H}_{T_G} \mathbf{T}(c) {}^{T_G}\mathbf{H}_{T'_G} \mathbf{T}(-c)$ <sup>8</sup>
**if** BOOLOVERLAP( $T, E, {}^E\mathbf{H}_{T'_G}$ ) **then**
 $c \leftarrow \text{true}$ 
**break**
 $t_T \leftarrow t_{free} t_T^{mod}$ 
 $t_R \leftarrow t_{free} t_R^{mod}$ 
 ${}^{T_G}\mathbf{H}_{T'_G} \leftarrow \mathbf{T}(\vec{d} t_T) \mathbf{R}(\vec{a}, \theta t_R)$ 
 ${}^E\mathbf{H}_{T'_G} \leftarrow {}^E\mathbf{H}_{T_G} \mathbf{T}(c) {}^{T_G}\mathbf{H}_{T'_G} \mathbf{T}(-c)$ 
*// Further refine by bisection between  ${}^E\mathbf{H}_{T_G}$  and  ${}^E\mathbf{H}_{T'_G}$* 

 BISECT( ${}^E\mathbf{H}_{T_G}, {}^E\mathbf{H}_{T'_G}, T, E$ )

 $\triangleright$  interpolate transform

 $\triangleright$  substepped pose

**end procedure**

<sup>5</sup>  $t_T^{step}$  is the desired precision of the CCD, e.g. 0.005.

<sup>6</sup>  $t_R^{step}$  is the desired angle precision of the CCD, e.g. 0.001. The rotation can then at most result in a translation of magnitude

$$2 \|\mathbf{p}(s)\| |\sin(0.5 t_R^{step})|$$

where  $s$  is the tool sphere that is furthest from the tool COM.

<sup>7</sup>  $N > 1$ , e.g. 4

<sup>8</sup>  $c$  is the tool COM in local space.

to material removal. This synchronization will further slow down the computation times, further complicating this approach.

This approach uses a CPU implementation of the bisection on the BVH level, by repeatedly calling the BOOLOVERLAP( $T, E, {}^E\mathbf{H}_T$ ) function, while modifying  ${}^E\mathbf{H}_T$ . Although this is simple to implement, the running time is

$$O((N + N_2)(\log^2 n)) = O(\log^2 n) \quad (5.8)$$

A faster to compute the CCD might be to do the bisection on the sphere level. This might be computed in a faster manner on the GPU by creating a thread for each tool leaf sphere and evaluating Equation (5.7) while traversing the environment BVH. The advantage is that we are immediately getting final collision times, which can be used as upper bounds to discard other spheres or even bounding spheres and their lower hierarchy completely.

**Algorithm 5.3** Translational CCD (CPU)

**Input:**  ${}^W\mathbf{H}_{T_G}, {}^W\mathbf{H}_E \in \mathbb{R}^{4 \times 4}, \Delta p(T_G) \in \mathbb{R}^3$   
**Result:**  $t \in \mathbb{R}$

```

procedure CCD
  select simulation GPU
   $T_G\mathbf{H}_E \leftarrow {}^W\mathbf{H}_{T_G}^{-1} {}^W\mathbf{H}_E$ 
   $\Delta p(T_G) \leftarrow {}^W\mathbf{H}_{T_G}^{-1} \Delta p(T_G)$ 
   $\Delta p_{\max} \leftarrow |\Delta p(T_G)|$ 
   $t \leftarrow \Delta p_{\max}$ 
  if  $\Delta p_{\max} > 0$  then
     $\Delta p(T_G) \leftarrow -\frac{\Delta p(T_G)}{\Delta p_{\max}}$ 
  else
    return
   $t \leftarrow \text{CCDPARALLEL}(t, \Delta p(T_G), \Delta p_{\max}, T_G\mathbf{H}_E)$ 
  synchronize collision GPU stream
  if  $t < \Delta p_{\max}$  then
     $t \leftarrow \text{clamp}(t, -\epsilon_m, \Delta p_{\max})$ 
     $t \leftarrow {}^W\mathbf{H}_{T_G}(-\Delta p(T_G) t)$ 
end procedure

```

## 5.1.2 Translational Continuous Collision Detection

Translational CCD disregards the rotation of the objects while determining the earliest point of contact. This is a considerable simplification, which might mean that the result is less “correct”. Though, a priori, it is not clear which method is the most correct, since the in-between motion is unknown. Though, one intuitively considers a linear motion to correspond well to the probability distribution of the defined poses at bounds. However, under the correct conditions, the results that translational CCD generates are of similar utility. These necessary conditions are: restoring physical simulation congruence by velocity correction (see Section 5.1.3) and integration of angular momentum before CCD<sup>9</sup>. Moreover, the running time of translational CCD is considerably lower than sample-based CCD, due to the following reasons:

1. Execution on the GPU with  $n$  threads, one per sphere, reduces time complexity to

$$O(\log n) < O((N + N_2)(\log n)) \quad (5.9)$$

2. The algorithm needs to be executed only once, therefore the factor  $(N + N_2)$  in the time complexity is removed.
3. The new algorithm can early prune BVH branches if no earlier collision can possibly be found there.

My translational CCD is described as a GPU algorithm in Algorithm 5.4, with a previous setup on the CPU in Algorithm 5.3. The algorithm calculates the travel distance  $t$  along  $\Delta p(T_G)$ , during which no intersection between tool and environment is detected. An offset of  $t \frac{\Delta p(T_G)}{|\Delta p(T_G)|}$  is then applied to the tool position, instead of the unconstrained translation  $\Delta p(T_G)$ . Therefore, the tool translation is shortened by the magnitude  $|\Delta p(T_G)| - t$ , in order to stay in an intersection-free state.

<sup>9</sup>I use a single discrete collision detection (CD) check after integrating the angular velocity while disregarding translational velocity. If the rotation is overlap free, translational CCD uses that orientation, otherwise use original orientation.

**Algorithm 5.4** Translational CCD Parallel (GPU)

**Input:**  $E$  sphere packing,  $t_{\text{root}}$  tool BVH root,  $t$ ,  $\Delta p_{\text{max}} \in \mathbb{R}$ ,  $\Delta p(T_G) \in \mathbb{R}^3$ ,  ${}^{T_G}\mathbf{H}_E \in \mathbb{R}^{4 \times 4}$   
**Result:**  $t \in \mathbb{R}$

```

procedure CCDPARALLEL
  for  $i \leftarrow i_t + i_b d_b$ ;  $i < |E|$ ;  $i \leftarrow i + d_b d_g$  do in parallel
    load  $e_i \in E$ 
     $a \leftarrow \Delta p(T_G) \cdot \Delta p(T_G)$ 
    if  $r(e_i) \leq 0^{10}$  or  $d(e_i) \leq 0^{11}$  then
      | continue
     $c(e_i) \leftarrow {}^{T_G}\mathbf{H}_E c(e_i)$ 
     $r(e_i) \leftarrow \mathcal{S}({}^{T_G}\mathbf{H}_E)_{0,0} r(e_i)$ 
    initialize stack  $\leftarrow \emptyset$ 
    node  $n \leftarrow t_{\text{root}}$ 
    repeat
      | for all  $c_j \in \text{children}(n)$  do
        |  $\vec{d} \leftarrow c(e_i) - c(c_j)$ 
        |  $b \leftarrow \Delta p(T_G) \cdot \vec{d}$ 
        |  $e \leftarrow \vec{d}^2 - (r(e_i) + r(c_j))^2$ 
        |  $f \leftarrow b^2 - ae$ 
        | if  $f < 0$  then
        | | continue
        |  $t_{\text{local}} \leftarrow (-b - \sqrt{f}) / a$ 
        | if leaf( $c_j$ ) then
        | | if  $\epsilon_m < t_{\text{local}} < \Delta p_{\text{max}}$  then
        | | | atomicMin( $t, t_{\text{local}}$ )12
        | | else if  $e \leq 0^{13}$  or  $\epsilon_m < t_{\text{local}} < \Delta p_{\text{max}}$  then
        | | | push  $c_j$  to stack
        | |  $n \leftarrow$  pop stack
      | until  $n = \emptyset$ 
  end procedure
    
```

<sup>10</sup>  $r(e_i) \leq 0$  indicates that the environment sphere has been drilled away, so we can ignore it.

<sup>11</sup>  $d(e_i) \leq 0$  is used for environment spheres that are part of an “empty” material layer, for example dental pulp (more on that in Section 7.3.2).

<sup>12</sup> One could think using an atomic operation over potentially all threads decreases performance, but it actually improves performance because of early pruning of irrelevant parts of the tree.

<sup>13</sup>  $e \leq 0$  indicates that BVH sphere  $c_j$  and environment sphere  $e_i$  overlap, which is often the case in contact situations, since  $c_j$  is a bounding sphere. This can not be true for inner spheres.

## 5.1.3 Maintaining Physical Laws Through Velocity Correction

If the collision detection can not advance the rotation or translation of the virtual tool according to the tool velocities, there is a mismatch between velocity and pose, which must be resolved in order to keep the order of the simulation<sup>14</sup>. In general, when given the pose of a body  $B$  due to unconstrained pose  ${}^W\mathbf{H}_{B'}$  and constrained pose<sup>15</sup>  ${}^W\mathbf{H}_{B_{\text{free}}}$  at this time, we can compute the translational and rotational velocity adjustments  $\Delta \vec{v}(B)$ ,  $\Delta \vec{\omega}(B)$  that will adjust  $B$ 's velocity to be physically consistent with the pose  ${}^W\mathbf{H}_{B_{\text{free}}}$

$$\Delta \vec{v}(B) = ({}^W\mathbf{H}_{B_{\text{free}}} \mathbf{c}_B - {}^W\mathbf{H}_{B'} \mathbf{c}_B) k_c \frac{1}{\Delta t} \quad (5.10)$$

$$\Delta \vec{\omega}(B) = \tau ({}^W\mathbf{H}_{B_{\text{free}}} {}^W\mathbf{H}_{B'}^{-1}) k_c \frac{1}{\Delta t} \quad (5.11)$$

where  $k_c \in (0, 1]$  is a constant to control the magnitude of correction, though generally I have used  $k_c = 1$ .

## 5.2 DETECTING POINTS OF CONTACT

The goal of contact detection is generate a list of contacts, that are existent between the two objects (here generally a tool and an obstacle). Contact configurations of two objects can be arbitrarily complex, and discretizing it as contacts allow the complex configuration to

<sup>14</sup> Otherwise, the velocity would be skewed towards the inside of the obstacle and future positional changes would keep pointing towards the colliding position.

<sup>15</sup> The constrained pose is derived during CCD (see Section 5.1)

be more easily described to allow further processing. In the physical world, contacts have the natural constraint to be always located on the surface of either object, as they can not overlap. Unfortunately, this is not generally true for simulations, and it can (easily) happen that contact situation exhibit defects such as:

1. Contacts are found, that lie on the inside of objects – due to penetration
2. Contacts on the surface are non-existent due to high penetration
3. The amount of found contacts fluctuates (seemingly) arbitrarily – also due to penetration

These are more reasons why it is advantageous to ensure non-overlapping bodies in the simulation (see Section 5.1). Missing surface contacts could lead to pop-through, where the tool implausibly slips through the obstacle. Contacts on the inside are especially disastrous when using sphere packings: The resulting force direction points in a direction that is essentially arbitrary, including inside of the obstacle, possibly resulting in the tool being forced further inside of the obstacle. The fluctuations of the contact count can not be avoided<sup>16</sup>, but they are highly limited, as the amount of overlap can be controlled and only fluctuates in a predictable manner. Consequently, I will present a simplified version of contact detection, that assumes the earlier mentioned problems are mostly taken care of through CCD.

The contact detection is major contribution to run time, even though it is *just* discrete collision detection. Interestingly, this is another issue, where combination with CCD minimizes the problem, by ensuring no interpenetration. Of course, discrete collision detection is easiest with objects that are far apart, such that simple bounding volumes already proof that a collision is not possible, and individual features can be disregarded. If the objects are deeply penetrating, the BVH needs to be traversed deeply too, and the speed-up is reduced. CCD can guarantee that objects do not penetrate, and consequently the speed-up of the BVH does not degrade significantly.

I implemented the contact detection by using the same preloaded data for sphere packings on the GPU side, which were also used in the CCD (refer to Algorithm 5.4). The algorithm setup is similarly done via a CPU program described in Algorithm 5.5, which essentially pushes the simulation state to GPU. The CPU code ultimately commands the GPU to run the contact detection algorithm (see Algorithm 5.6). Here, we traverse the BVH of the tool, checking against all the spheres of the slightly enlarged environmental object. Intersecting pairs of spheres are inserted into a global list of contacts  $c_i \in C$  with contact properties such as:

1. **Contact point**  $p(c_i)$ : The halfway point between the surfaces of both spheres, along the line that connects the centers.
2. **Contact normal**  $\vec{n}(c_i)$ : Normalized direction from environment sphere to tool leaf sphere<sup>17</sup>.
3. **Contact dop**  $dop(c_i)$ : The depth of penetration (dop) of tool leaf inside environment sphere, projected onto the line between the centers.
4. **Contact volume**  $V(c_i)$ : Volumetric dop of tool leaf inside environment sphere, which is the overlapping volume.
5. **Contact density**  $\rho(c_i)$ : Environment spheres' material density.
6. **Drilling contact dop**  $d_D(c_i)$ : Translational dop of actively drilling tool leaf inside environment spheres.
7. **Drilling contact volume**  $V_D(c_i)$ : Volumetric dop of actively drilling tool leaf.
8. **Contact distance**  $d(c_i)$ : The distance between the tool leaf and the *unenlarged*<sup>18</sup> environment sphere.

<sup>16</sup>Any amount of fluctuations in contact count can be minimized through clustering, this will be detailed in Section 5.3

<sup>17</sup>w.l.o.g. I assume the tool is to be pushed outside of the environment, otherwise reverse the direction.

<sup>18</sup>By using the unenlarged sphere, this distance is the actual distance between the two objects, which I need during material removal, as I want to eliminate distance from the removal equation to get completely drilling consistent behaviour.

---

**Algorithm 5.5** Contact Detection (CPU)
 

---

**Input:**  ${}^W\mathbf{H}_{T_G}, {}^W\mathbf{H}_E \in \mathbb{R}^{4 \times 4}$ 
**Result:**  $C$ 

```

procedure CONTACTDET
    select simulation GPU
     ${}^{T_G}\mathbf{H}_E \leftarrow {}^W\mathbf{H}_{T_G}^{-1} {}^W\mathbf{H}_E$ 
    reset contacts  $C$ 
     $C \leftarrow \text{CONTACTDETPARALLEL}({}^{T_G}\mathbf{H}_E, C)$ 
    synchronize collision GPU stream
    for all  $c_i \in C$  do
        transform from  $T_G$  to  $W$ :  $\{V, V_D, \rho, \rho_S, d, d_D, \mathbf{p}, \vec{\mathbf{n}}\} (c_i)$ 
end procedure
    
```

---

The GPU code assigns each thread a set of distinct environment spheres according to the thread ID. After completing the processing loop for an environment sphere, the thread uses a stride by the thread block dimension to grab the next distinct environment sphere. Processing an environment sphere involves loading related data from memory, transforming them from their local space to the tool space and starting BVH traversal of the tool at the root node. I process the BVH tree by recursion<sup>19</sup>. I process a BVH node by iterating over its child nodes, determining for each if the enlarged environment sphere of this thread intersects the tool node child, and if it is not a leaf, we push it to the stack to be further traversed. If it is an intersecting leaf, we calculate the intersection volume, dop and once per thread, we determine an index  $k$  in the global list of contacts  $C$  to reserve a spot in the list to parallelly write data later on. The loop ends when the stack is empty, thereby, we have processed all BVH nodes.

<sup>19</sup>I implemented the recursion by using an explicit stack, to optimize performance over standard recursion.

---

**Algorithm 5.6** Contact Detection Parallel (GPU)
 

---

**Input:**  $E$  sphere packing,  $t_{\text{root}}$  tool BVH root,  ${}^{T_G}\mathbf{H}_E \in \mathbb{R}^{4 \times 4}$ ,  $C$ 
**Result:**  $C$ 

```

procedure CONTACTDETPARALLEL
    for  $i \leftarrow i_t + i_b d_b; i < |E|; i \leftarrow i + d_b d_g$  do in parallel
        coalesced read  $c_i \in E$ 
        if  $r(c_i) \leq 0^{20}$  or  $\rho(c_i) \leq 0^{21}$  then
            continue
             $\mathbf{c}(c_i) \leftarrow {}^{T_G}\mathbf{H}_E \mathbf{c}(c_i)$ 
             $r(c_i) \leftarrow \mathcal{S}({}^{T_G}\mathbf{H}_E)_{0,0} r(c_i)$ 
             $r_e(c_i) \leftarrow r(c_i) + P_{\text{ContactMargin}}^{22}$ 
             $k \in \mathbb{N} \leftarrow -1$ 
            initialize with  $\mathbf{0}$ :  $\mathbf{p}(c_i) \in \mathbb{R}^3$ 
            initialize with  $\vec{\mathbf{0}}$ :  $\vec{\mathbf{n}}(c_i) \in \mathbb{R}^3$ 
            initialize with  $\mathbf{0}$ :  $\{d, d_D, V, V_D, \rho, \text{dop}\} (c_i) \in \mathbb{R}, N \in \mathbb{N}$ 
            initialize stack  $\leftarrow \emptyset$ 
            node  $n \leftarrow t_{\text{root}}$ 
            repeat
                for all  $t_j \in \text{children}(n)$  do
                     $\vec{\mathbf{d}} \leftarrow \mathbf{c}(c_i) - \mathbf{c}(t_j)$ 
                     $d \leftarrow |\vec{\mathbf{d}}| - r_e(c_i) - r(t_j)$ 
                    if  $d \geq 0$  then
                        continue ▷ no overlap between  $c_i, t_j$ 
                    if not leaf  $(t_j)$  then
                        push  $t_j$  to stack ▷ inner nodes cause traversal
                    continue
            
```

<sup>20</sup> $r(c_i) \leq 0$  indicates that the environment sphere has been drilled away, so we can ignore it.

<sup>21</sup> $\rho(c_i) \leq 0$  is used for environment spheres that are part of an “empty” material layer, for example dental pulp (see Figure 7.12).

<sup>22</sup> $P_{\text{ContactMargin}}$  needs to be chosen based on the scene dimensions. I have used 2 mm.

```

if  $d < -2 \min(r(e_i), r(t_j))$  then
    |  $V \leftarrow \frac{4}{3} \pi \min(r(e_i), r(t_j))^3$ 
else
    |  $a \leftarrow r(e_i) + r(t_j) - |\vec{d}|$ 
    |  $V \leftarrow \frac{\pi a^2 (|\vec{d}|^2 + 2|\vec{d}|r(t_j) + |\vec{d}|r(e_i) + 3r(t_j)^2 + 3r(e_i)^2 + 6r(e_i)r(t_j))}{12|\vec{d}|}$ 
if  $k = -1$  then  $k \leftarrow \text{ATOMICINC}(|C|)$ 23
if  $\text{interactive}(t_j)$ 24 then
    |  $\rho(c_i) \leftarrow \rho(e_i) + \rho(c_i)V$ 
    |  $V_D(c_i) \leftarrow V_D(c_i) + V$ 
    |  $d_D(c_i) \leftarrow \min(d_D(c_i), |\vec{d}| - r(e_i) - r(t_j))$ 25
 $N \leftarrow N + 1$ 
 $V(c_i) \leftarrow V(c_i) + V$ 
 $\vec{n}(c_i) \leftarrow \vec{n}(c_i) + (c(e_i) + \hat{d}r(e_i))V$ 
 $w \leftarrow 1$ 
if  $\text{interactive}(t_j)$  then
    |  $w \leftarrow \max(0, P_{\text{dop}}(1 - (\max(0, -\vec{d}\vec{w}_d(t_j)))^{P_{\text{drill}}}))$ 26
    |  $\text{dop}(c_i) \leftarrow \max(-(|\vec{d}| - r_e(e_i) - r(t_j))w, \text{dop}(c_i))$ 
    |  $d(c_i) \leftarrow \min(|\vec{d}| - r(e_i) - r(t_j), d(c_i))$ 
 $n \leftarrow \text{pop stack}$ 
until  $n = \emptyset$ 
if  $N > 0$  and  $V(c_i) > 0$  then
    |  $\hat{p}(c_i) \leftarrow \hat{p}(c_i) / V(c_i)$ 
    |  $\vec{n}(c_i) \leftarrow \vec{n}(c_i) / V(c_i)$ 
if  $V_D(c_i) > 0$  then
    |  $\rho(c_i) \leftarrow \rho(c_i) / V_D(c_i)$ 
write  $c_i$  to  $C + k$ 
    ▷ global memory for later host transfer27
end procedure
    
```

<sup>23</sup>The generated contact set index is unique, among all threads (globally). We thereby reserve a memory address to store data in.

<sup>24</sup>Interactive spheres contribute to another linearized contact representation, which includes material density, which influences how fast material is removed upon drilling contact.

<sup>25</sup>Note: I use the unextended radius to get an accurate distance, unaffected by radial sphere extensions.

<sup>26</sup> $P_{\text{dop}}, P_{\text{drill}}$  are global parameters, that typically do not change between frames, therefore their values are only transferred to GPU on key-frames.

<sup>27</sup>Here, we are writing all properties of the local contact point  $c_i$ :  $\{\hat{p}, \vec{n}, \text{dop}, d, d_D, V, V_D, \rho\}(c_i)$ . This is done in parallel, as we reserved the memory location for this thread earlier.

When the GPU code concludes, we transfer the contact points from GPU to CPU and transform all contact points' properties from tool space to world space. Given the found contact points, we can proceed to (i) estimate the local surface properties (see Algorithm 5.7), and (ii) cluster the contacts (see Algorithm 5.8), and finally (iii) solve the contacts (see Algorithm 5.9).

### 5.2.1 Estimating Surface Properties by Linearization

The purpose of surface estimation is to probe the surface of the environment around the current tool pose, to get an estimate of how the surface is currently constituted at the contact. The surface properties, that I am interested in gathering are:

1. Surface normal
2. Material density (weighted average)
3. Material density (weighted sum)
4. Contact volume
5. Contact center
6. Distance to the surface (using the unextended radius)
7. Contact volume of drilling parts
8. Distance of drilling parts to the surface

**Algorithm 5.7** Estimation of Local Surface**Input:**  $C$ **Result:**  $\{V, V_D, \rho, \rho_S, d, d_D\} (C) \in \mathbb{R}$ ,  $\{\mathbf{p}, \mathbf{\tilde{n}}\} (C) \in \mathbb{R}^3$ ,  $C_{contact}, C_{drillContact} \in \{0, 1\}$ **procedure** SURFACEESTIMATION  **initialize** with  $\mathbf{0}$ :  $\mathbf{p} (C), \mathbf{\tilde{n}} (C) \in \mathbb{R}^3$   **initialize** with 0:  $\{V, V_D, \rho, \rho_S\} (C) \in \mathbb{R}$   **initialize** with  $\infty$ :  $d (C), d_D (C) \in \mathbb{R}$   **for all**  $c_i \in C$  **do**     $V (C) \leftarrow V (c_i)$  ▷ contact volume     $V_D (C) \leftarrow V_D (c_i)^{28}$  ▷ contact volume of drilling parts     $\mathbf{\tilde{n}} (C) \leftarrow \mathbf{\tilde{n}} (c_i) V (c_i)$  ▷ surface normal     $\mathbf{p} (C) \leftarrow \mathbf{p} (c_i) V (c_i)$  ▷ contact point     $\rho (C) \leftarrow \rho (c_i) V_D (c_i)$  ▷ surface density     $d (C) \leftarrow \min (d (C), d (c_i))$  ▷ minimum separating distance (MSD)     $d_D (C) \leftarrow \min (d_D (C), d_D (c_i))$  ▷ MSD between drilling parts  **if**  $V (C) > 0$  **then**     $\mathbf{\tilde{n}} (C) \leftarrow \mathbf{\tilde{n}} (C) / V (C)^{28}$      $\mathbf{p} (C) \leftarrow \mathbf{p} (C) / V (C)$    $\rho_S (C) \leftarrow \rho (C)$   **if**  $V_D (C) > 0$  **then**     $\rho (C) \leftarrow \rho (C) / V_D (C)^{28}$    $C_{contact} \leftarrow |C| > 0$  **or**  $V > 0$  ▷ is there any intersection?  $\Rightarrow$  contact   $C_{drillContact} \leftarrow V_D > 0$  ▷ are drilling parts of the tool intersecting?**end procedure**

<sup>28</sup>As this estimation is not perfect, in practice I use the exponential moving average (EMA) to update the surface properties for density  $\rho$ , normal  $\mathbf{\tilde{n}}$  and drilling contact volume  $V_D$ , taking 0.6% of the new value. I chose this rather low value since the sampling frequency is very high with 1–4 kHz.

More surface properties could easily be added, though they would first need to be included in the material model. A possible addition is friction coefficients, meaning frictional behavior can vary on different.

Since all the necessary information can be found in the contact set, we just need to iterate over each contact and compute a gather function. Most properties are computed using a weighted summation and some use a minimum. Since the contact count is usually considerably smaller than 1000, the computation of these gather functions is very fast, making up a negligible amount of the frame-time.

### 5.2.2 Clustering Nearby Contacts

The basic idea behind contact clustering is that contacts are clustered due to their proximity, and a presumed redundancy because of that. Consequently, a cluster is presumed to well represent the average of all proximate contacts (see Figure 5.2). Removing the redundant contact points makes the contact set more manageable, which could benefit the computation of the contact resolution (see Section 5.3). In fact, clustering even provides a number of benefits to the distribution of the contacts. In detail, the following issues are reasons that one might utilize contact clustering:

- **Low Performance of Contact Resolution**

The performance of some methods of contact resolution (see Section 5.3) are highly dependent on the contact count. A notable example for this are constraint-based simulations, where each contact point imposes a constraint on the motion of the tool. To solve this set of constraints gets much more complicated with each new constraint. Therefore, limiting the amount of constraints by clustering can be a way to optimize the performance.



**Figure 5.2: Contact clustering.** I inserted the acetabular reamer secondary handle into the pelvis cavity and enabled gravity, the tool then settles in this configuration. The amount of contacts (red) is significantly reduced when enabling clustering (*Bottom*). However, the pose is held much more stably when using all contacts without clustering (*Top*), even though many contacts are *seemingly* redundant.

- **Irregular Amounts of Contact Points**

There are volumetric representations that have an undesired link between contact count and force stiffness. An example is the VoxMap-PointShell algorithm [McNeely et al., 1999a], in which the amount of points of the tool that intersect the VoxMap are linked to the dop of the tool. Therefore, besides the distance of the voxel to the surface, the amount of intersecting points also scales up the rendered force stiffness. This is not a desired effect, as, ideally, no matter the distribution of points in a shell of a tool, all sides of the tool intersecting at the same dop should result in the same force magnitude rendered to the user. Imagine you have a cone, where the pointy end has 1 point and the flat end has  $N$  points, the force experienced when touching with the flat end will be  $N$  times higher.

- **Imbalanced Contact Normals**

This is another issue that came up with the VoxMap-PointShell algorithm [Xu and Barbic, 2016]. Given an uneven distribution of points in the shell of an object, the normals of each point are unevenly represented in the set of contact points. For example, if you push the previously mentioned cone laterally against a wall, the flat end will have some amount  $m$  contact points, whereas the pointy end always has a maximum of 1 contact point. The result of this uneven distribution is, that the cone would have torques from the flat end's side acting on it, that are  $m$  times larger than those of the pointy end. Xu and Barbic [2016] have shown that this can lead to pop-through events.

I implemented clustering to explore, whether it could improve the haptical sensation of my simulation. In hindsight, I in fact realize that my simulation does not suffer from any of the above issues, and it is therefore not necessary for my system. However, the general usefulness of contact clustering has been proven by other developers of haptic rendering applications, and it is possible the use-cases that I employed the simulation in, do not force these issues to emerge. Additionally, I still regard clustering as an important detail that I want to cover to give a complete description of a haptic rendering and simulation system.

I also want to mention that clustering does not inherently improve the quality of contact detection. In fact, I found that for simulation, resolving all contacts gives more convincing results, and some forms of clustering are a noticeable decrease in stability and haptic quality. This does make sense, since clustering, besides the previously mentioned benefits, also essentially introduces a form of noise, and the exact set of contact points might no longer be present to resolve them.

In order to drive the contact resolution algorithm (see Section 5.3) with contact clusters, I simply assume contact are always clustered. However, if I disabled clustering, each contact point becomes its own separate cluster. This approach, of course, results in some performance penalty, though it was imperceivable, and it simplifies the development and on-demand configuration process. Additionally, the system needs to run at a sufficient frequency while using clustering too, which adds to the performance demand. Therefore, the unclustered configuration is not the performance bottleneck that crucially needs to be optimized.

The presented algorithm works reasonably well considering its simplicity. Even though the worst case run-time complexity is exponential<sup>30</sup>, in case every point is generating its own cluster. In practice, the algorithm usually terminates in a few microseconds and its performance impact is negligible. Still, there are two limitations of this approach:

1. It tries to cluster together contacts which are located closer than the cluster threshold. However, depending on the order of processing the contacts, this constraint is usually not globally enforced, as clusters grow during the linear search.
2. The cluster threshold needs to be tuned manually<sup>31</sup>.

<sup>30</sup>  $O(n) = \frac{n^2}{2}$

in the worst case, where  $n$  is the contact count. One could easily improve the runtime complexity to  $O(n) = n$ , in case we limit the number of clusters to some constant  $k$  and assign remaining contacts to the closest cluster (see Algorithm A.1).

<sup>31</sup>The threshold could be automatically derived if one can define some common contact situations and a desired cluster count, and then optimize the cluster threshold accordingly.

**Algorithm 5.8** Contact Clustering**Input:**  $C$ **Result:**  $L$ 

```

procedure CONTACTCLUSTERING
  if  $P_{ClusterThreshold} \leq 0$  then
     $L \leftarrow C$  ▷ no clustering  $\Rightarrow$  a cluster for each contact with  $N = 1$ 
    return
  for all  $c_i \in C$  do
     $p \leftarrow p(c_i)$ 
     $\vec{n} \leftarrow \vec{n}(c_i)$ 
     $dop \leftarrow dop(c_i)$ 
     $V \leftarrow V(c_i)$ 
     $V_D \leftarrow V_D(c_i)$ 
    found_cluster  $\leftarrow$  false
    for all  $l_j \in L$  do
      if  $\left\| p - \frac{p(l_j)}{N(l_j)} \right\| < P_{ClusterThreshold}$  then
        found_cluster  $\leftarrow$  true
         $p(l_j) \leftarrow p^{29}$ 
         $\vec{n}(l_j) \leftarrow \vec{n}$ 
         $dop(l_j) \leftarrow \max(V(l_j), dop)$ 
         $V(l_j) \leftarrow V$ 
         $V_D(l_j) \leftarrow V_D$ 
         $N(l_j) \leftarrow 1$ 
      if not found_cluster then ▷ no cluster near  $\Rightarrow$  create new cluster
         $p(l) \leftarrow p$ 
         $\vec{n}(l) \leftarrow \vec{n}$ 
         $dop(l) \leftarrow dop$ 
         $V(l) \leftarrow V$ 
         $V_D(l) \leftarrow V_D$ 
         $N(l) \leftarrow 1$ 
         $L \leftarrow L \cup \{l\}$ 
    for all  $l_j \in L$  do ▷ average all properties except penetration depth
       $p(l_j) \leftarrow p(l_j) / N(l_j)$ 
       $\vec{n}(l_j) \leftarrow \vec{n}(l_j) / N(l_j)$ 
       $V(l_j) \leftarrow V(l_j) / N(l_j)$ 
       $V_D(l_j) \leftarrow V_D(l_j) / N(l_j)$ 
end procedure

```

<sup>29</sup> I did not experiment with weighting the contact points with the contact volume when clustering them, that could possibly improve accuracy further.

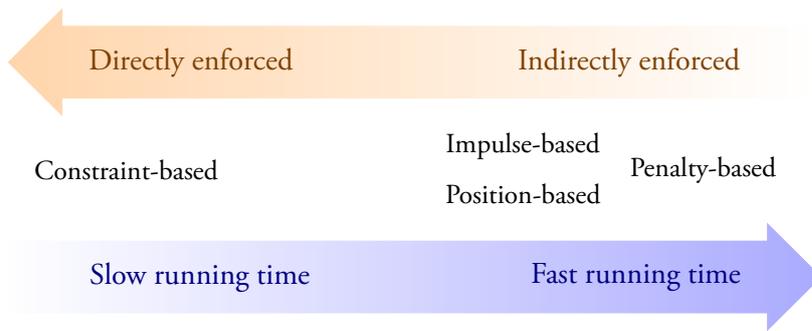
I have also developed a grid-based clustering algorithm, which does not suffer from the both aforementioned issues. The idea is: contact points are clustered if they occupy a common grid cell. The grid resolution can be hard-coded and the AABB of the contact set determine the grid bounds, therefore grid cell dimensions are defined. However, I have found that this approach causes serious quality degradation of the contacts, which makes it unusable in many cases.

*“Be not lost So poorly in your thoughts.”*

— William Shakespeare

## 5.3 RESOLVING CONTACTS

The previously found contacts define the contact configuration between the dynamic tool and the static environment. Contact between objects is the essential way that they physically interaction. As such, each contact potentially imposes a change in the physical state of the involved objects. In my case, the environment is static, therefore they are unaffected



**Figure 5.4: Contact resolution methods.** Methods to resolve contacts and how directly they enforce constraints, compared to their running time.

Penalty aggregates forces based on dop.

Impulse- and position-based approaches solve constraints one-by-one, repeatedly.

Constraint-based approaches minimize kinetic distance over the feasible set of poses, which makes it worse for large contact sets.

---

### Algorithm 5.9 Contact Resolution Meta

---

**Input:** contacts  $C$

**Result:** correct state variables of  $T_G$

```

procedure CONTACTRESOLUTION
  if use clustering then
    procedure CONTACTCLUSTERING
      sum up contact set in clusters of proximate contacts
  if use penalty-method then
    PENALTYCONTACTRESOLUTION( $C$ )
  else if use impulse-method then
    IMPULSECONTACTRESOLUTION( $C$ )
  else
    HYBRIDCONTACTRESOLUTION( $C$ )
end procedure

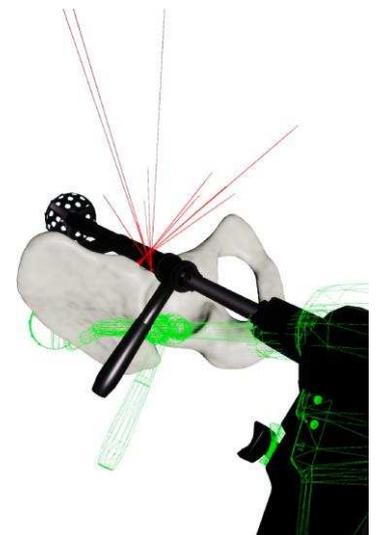
```

---

by contact, and thus, I only need to consider the physical state of the virtual tool. Each contact point is a (small) overlap of tool and obstacle, which is already physically implausible, though small overlap is allowed, since we detect the overlap with a 2 mm margin. Still, each contact point is *nearly* violating physicals, and the purpose of contact resolution is to avoid further physical violations. Therefore, each contact point can be regarded as a constraint that is imposed upon the tool (example in Figure 5.3), such as the point velocity at each contact point should no longer point towards the surface of the environment. The exact formulation of these constraints and how these constraints are enforced depends on the used resolution method. In Figure 5.4, I have laid out the basic types of contact resolution approaches and how directly they enforce these constraints. The running time increases as the directness of the enforcement, and consequently the supposed rate of success, increases. To achieve a high degree of realism the contact detection and resolution needs to behave in physically plausible way. Both parts, detection and resolution, have to work well to produce physically plausible results. However, different from contact detection, contact resolution need not necessarily take up a large amount of computation time to generate convincing results. I have implemented penalty-based and impulse-based contact resolution, and a hybrid between the two (see Algorithm 5.9).

#### 5.3.1 Robust Penalty-Based Contact Resolution

The simplest approach to contact resolution is the penalty-based approach. This approach is far removed from the idea of enforcing constraints, instead one allows for overlap and resolves it by penalizing overlap by application of force, proportional to the dop. Here, we don't focus on the end-goal of enforcing constraints, but rather the means of enforcement a physical meaning. When using a linear dop, one can think of each intersection point to tension a spring, proportional to the dop. In case one uses a volumetric measure for the



**Figure 5.3: Reamer in contact.** The acetabular reamer (Graphic Tool in black) has multiple contacts (red lines) with the hip (white), as it is trying to follow the Haptic Tool (green). Multiple contacts and arbitrary large deviation of graphic tool and haptic tool can be rendered stably, without pop-through or other artifacts. Hip (200k spheres) and tool (5k spheres) contact can be simulated in this configuration at roughly 2 kHz.

dop, such as cubic, one can rather think of the other object being a very dense liquid and applying a buoyancy force at each point of intersection. I will present penalty-based contact resolution, using a linear and a cubic dop for sphere packings.

Algorithm 5.10 shows the algorithm, which essentially accumulates the translational force  $\vec{f}_c$  and rotational force  $\vec{\tau}_c$  by the weighted sums

$$\vec{f}_c = \sum_{c_i \in C} \vec{n}(c_i) \text{dop}(c_i) k \quad (5.12)$$

$$\vec{\tau}_c = \sum_{c_i \in C} (\mathbf{p}(c_i) - \mathbf{p}(T_G(c_i))) \times (\vec{n}(c_i) \text{dop}(c_i) k) \quad (5.13)$$

Although, further improvement is possible by saturation of the force a contact point contributes, as well the sum of forces. Since the forces are used to calculate the torques, torques do not need to be limited separately. However, the sum of torques needs to be separately saturated.

---

**Algorithm 5.10** Penalty Contact Resolution
 

---

**Input:** contact set  $C$

**Result:**  $\vec{f}_c, \vec{\tau}_c \in \mathbb{R}^3$

**procedure** PENALTYCONTACTRESOLUTION

$\vec{f}_c, \vec{\tau}_c \leftarrow \vec{0}$

**for all**  $c_i \in C$  **do**

$\vec{r}(T_G) \leftarrow [\mathbf{p}(c_i) - \mathbf{p}(T_G(c_i))] k_l^{32}$  ▷ torque lever

$\vec{f} \leftarrow \frac{\vec{n}(c_i)}{\|\vec{n}(c_i)\|} \text{dop}(c_i) k_c^{33}$

**if**  $f_{max} > 0$  **and**  $f_{max} > \|\vec{f}\|$  **then**

$\vec{f} \leftarrow \vec{f} \frac{f_{max}^{34}}{\|\vec{f}\|}$

$\vec{f}_c \leftarrow \vec{f}_c + \vec{f}$

$\vec{\tau}_c \leftarrow \vec{\tau}_c + \vec{r}(T_G) \times \vec{f}$

**if**  $F_{max} > 0$  **and**  $F_{max} > \|\vec{f}_c\|$  **then**

$\vec{f}_c \leftarrow \vec{f}_c \frac{F_{max}^{35}}{\|\vec{f}_c\|}$

**end procedure**

---

<sup>32</sup>  $k_l = P_{\text{LeverMod}}$ , which I generally leave at

<sup>33</sup>  $k_c = P_{\text{ContactStiffness}}$ , for which I used

$1 \frac{\text{kN}}{\text{m}}$ .

<sup>34</sup>  $f_{max} = P_{\text{ContactForceLimitSingle}}$ , which I did not use in production.

<sup>35</sup>  $F_{max} = P_{\text{ContactForceLimit}}$ , for which I use 5 N (this seems small, but I set the tools to a low mass of < 100 g, therefore the effective force limit is equal to 110 N).

### 5.3.1.1 Defining the Depth of Penetration

Here, I will present the two measures that I have used to serve as dop during penalty-based contact resolution.

**Translational dop** Since we are looking to create a repulsive force, a spring is one of the most common types of forces that one can simulate, which is exactly what translational dop does. Imagine that the surface of the environment is locally completely elastic and any intersection with the tool locally deforms the environment surface (see Figure 5.5). The surface is attracted to the original surface position via a spring to go back to its original shape.

Implementing this concept well for spherepackings is non-trivial, since for a discrete point in time the local outside of one object can not be inferred from two colliding spheres in general. See the example in Figure 5.6, tool spheres (in blue) might quite literally be surrounded by environment spheres (in red), therefore an arbitrary force direction could emerge from colliding sphere pairs. The resulting force would have a large amount of noise in the summed force, and might point towards the environment, leading to an attracting

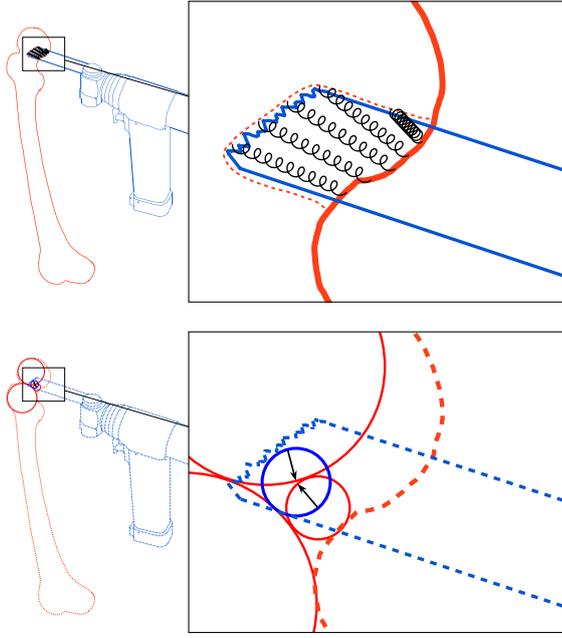


Figure 5.5: **Translational dop motivation.** Springs along the surface are tensioned when another object penetrates. The summated force of all springs acts upon the penetrating object.

Figure 5.6: **Interpenetrating spherepackings.** Spheres with centers inside other objects might produce forces that point in arbitrary directions. The red environment sphere on the right results in an attracting force, pulling the tool inside, whereas a repellent force is generally desired. In the best case, this sort of error can be seen as noise of the force direction. In the worst case, it leads to instability.

force, rather than a repelling one. In a stiff simulation, this kind of serious error would likely destabilize the simulation.

Weller et al. [Weller and Zachmann, 2009b] have proposed normal cones to mitigate the issue. However, firstly, this approach only works if the surface features do not change at runtime, which they do, of course, when working with material removal. Secondly, allowing for deep penetration with normal cones, still would allow for pop-through events, since normal cones statically assign in which direction the outside is located. Therefore, this solution is not suitable for my use-case.

The emergence of this problem is facilitated by object interpenetrating, since that allows sphere centers of one object to enter the other object. If we can guarantee that no sphere center of one object can enter another object, the differences between colliding spheres' centers are always meaningful and are pointing in congruent<sup>36</sup> directions. To circumvent this issue, I made sure that the whole simulation does not allow for interpenetration, which is ultimately enforced by the continuous collision detection (see Section 5.1). Instead, the tool spheres are momentarily enlarged by  $P_{\text{ContactMargin}}$  during contact detection to allow for contact, without sphere centers being able to enter other objects.

With these boundary conditions met, one can simply formulate the translational dop between an environment sphere  $e_i$  and a tool sphere  $t_j$  as

$$r_e(e_i) = r(e_i) + P_{\text{ContactMargin}} \quad (5.14)$$

$$\text{dop}(e_i, t_j) = \|\mathbf{c}(e_i) - \mathbf{c}(t_j)\| - r_e(e_i) - r(t_j) \quad (5.15)$$

The calculation is part of contact detection (see Algorithm 5.6).

**Volumetric dop** One selling point of volumetric representations, such as sphere packings, is their ability for simple computation of the intersection volume between objects. Given two spheres  $e_i$  and  $t_j$ , we can compute the intersection volume  $V(e_i, t_j)$  as

$$\vec{d}(e_i, t_j) = \mathbf{c}(e_i) - \mathbf{c}(t_j) \quad (5.16)$$

$$d(e_i, t_j) = \|\vec{d}(e_i, t_j)\| - r_e(e_i) - r(t_j) \quad (5.17)$$

$$a = r(e_i) + r(t_j) - \|\vec{d}\| \quad (5.18)$$

$$V(e_i, t_j) = \frac{\pi a^2 (\|\vec{d}\|^2 + 2\|\vec{d}\|r(t_j) + \|\vec{d}\|r(e_i) + 3r(t_j)^2 + 3r(e_i)^2 + 6r(e_i)r(t_j))}{12\|\vec{d}\|} \quad (5.19)$$

<sup>36</sup>By congruent, I mean that any contact point is pointing towards the same object's surface.

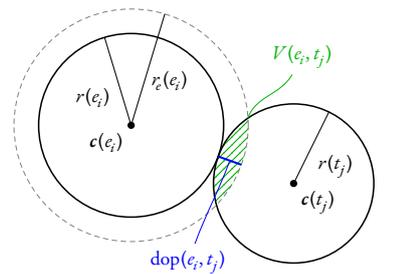


Figure 5.7: **Depth-of-penetration measures.** Sketch of an intersecting sphere pair and both dop measures visualized. Both are well-defined and meaningful.

However, in case  $d(e_i, t_j) > 0$  there is no intersection, therefore  $V(e_i, t_j) = 0$ . If  $d(e_i, t_j) < -2 \min(r(e_i), r(t_j))$ , one sphere is totally enclosed by the other, in which case the above equation no longer holds, instead

$$V(e_i, t_j) = \frac{4}{3} \pi \min(r(e_i), r(t_j))^3 \quad (5.20)$$

The calculation is part of contact detection (see [Algorithm 5.6](#)).

Calculating the displacement force based on the intersection volume is reminiscent of buoyancy force that a displaced volume exerts on objects that are emerged in it. Therefore, employing it in simulation means one regards one interpenetrating object as an extremely dense fluid. Buoyancy force  $\vec{f}_b$  is defined as a force pointing in the inverse direction of gravity,  $-g$ , with a magnitude of the product of fluid density  $\rho_f$  and displaced fluid  $V_{disp}$ , which is located at the centre of buoyancy  $c_{sub}$  [[Bajo et al., 2020](#)]. Therefore, a body  $B$  would be experiencing the buoyancy forces

$$\vec{f}_b = -\vec{g} \rho V_{disp} \quad (5.21)$$

$$\vec{\tau}_b = (c_{sub} - \mathbf{p}(B)) \times \vec{f}_b \quad (5.22)$$

Given a set of spheres  $e_i$  and  $t_j$ , one can now define the centre of buoyancy as the weighted sum of all intersecting sphere pairs:

$$c_{sub} = \frac{\sum_{V(e_i, t_j) > 0} V(e_i, t_j) \mathbf{c}(e_i) - \left( \frac{\vec{d}}{\|\vec{d}\|} - 0.5d \right)}{\sum_{V(e_i, t_j) > 0} V(e_i, t_j)} \quad (5.23)$$

The magnitude can be calculated as

$$V_{disp} \rho_f = \frac{\sum_{V(e_i, t_j) > 0} V(e_i, t_j) \rho_f}{\sum_{V(e_i, t_j) > 0} V(e_i, t_j)} \quad (5.24)$$

The direction  $\vec{g}$  is the direction that gravity is acting towards. We want forces to possible act in other directions, as this formulation is the special case of liquid that has settled on a body with large mass such as earth. To generalize this, one can use the normal direction between sphere pairs, where I always keep the same order for the subtraction of centers as such:

$$\vec{n}(e_i, t_j) = V(e_i, t_j) \frac{\vec{d}(e_i, t_j)}{\|\vec{d}(e_i, t_j)\|} \quad (5.25)$$

$$\vec{n}_c = \frac{\sum_{V(e_i, t_j) > 0} \vec{n}(e_i, t_j)}{\sum_{V(e_i, t_j) > 0} V(e_i, t_j)} \quad (5.26)$$

Further, I defined the levers, at which a contact force is applied relative to the tool body COM  $\mathbf{p}(T)$  to yield a torque, as

$$\vec{r}(e_i, t_j) = \left[ \mathbf{c}(e_i) - \left( \frac{\vec{d}}{\|\vec{d}\|} - 0.5d \right) \right] - \mathbf{p}(T) \quad (5.27)$$

Given those, one can easily calculate the earlier defined  $\vec{f}_b$  and  $\vec{z}_b$ , in the more general form, called  $\vec{f}_c$  and  $\vec{z}_c$ , which then simplifies to the following terms:

$$\vec{f}_c = \sum_{V(e_i, t_j) > 0} V(e_i, t_j) \frac{\vec{d}(e_i, t_j)}{\|\vec{d}(e_i, t_j)\|} \quad (5.28)$$

$$\vec{z}_c = \sum_{V(e_i, t_j) > 0} \vec{r}(e_i, t_j) \times \left[ V(e_i, t_j) \frac{\vec{d}(e_i, t_j)}{\|\vec{d}(e_i, t_j)\|} \right] \quad (5.29)$$

Though, it is not clear at all why this model is any more fitting to use in simulation than the spring-on-surface model, which translational dop is based on. Both are a continuous dop measure that is always well defined and provides useful information. In practice, I have found no perceivable difference between using translational dop or volumetric dop for penalty-based simulation.

### 5.3.1.2 Equally Weighted Normals Through Adaptive Stiffness

One reason to employ contact clustering is to make the distribution of contact points more uniform in terms to their *position*. Xu et al. [Xu and Barbic, 2016] have developed a method for penalty-based contact resolution that aims to make the distribution of contact points uniform in terms of their *normal direction*. The idea is that contact points are scaled inversely to their normal direction frequency of occurrence. Thus, contact points with more unique normal directions are scaled up, while contact points with common normal directions are scaled down, due to their redundant direction. The algorithm (details in Algorithm 5.11) generates a weight  $\phi_{di}$  for each contact  $c_i \in C$ . The penalty-force can then use these weights in the force summation, for example calculated as

$$\vec{f} = \sum_{i=0}^{\#C} \vec{n}(c_i) V(c_i) k \frac{\phi_{di}}{\phi_{max}} \quad (5.30)$$

though the details depend on the details of the used penalty-based force rendering.

The algorithm's running time complexity is exponential:  $O(n^2)$  with  $n$  being the contact count. This complexity is due to iterating over each contact, and each contact's weight is calculated based on the difference to all other contacts' normals. In this case, the theoretical running time complexity well shows the practical running time of the algorithm.

---

#### Algorithm 5.11 Adaptive Stiffness

---

**Input:** contact set  $C$

**Result:** contact set  $C$

```

procedure ADAPTIVESTIFFNESS37
  for all  $c_i \in C$  do
     $\vec{z}_i \leftarrow \frac{(p(c_i) - p(G)) \times \vec{n}(c_i)}{\|(p(c_i) - p(G)) \times \vec{n}(c_i)\|}$ 
    for all  $c_j \in C$  do
       $\vec{z}_j \leftarrow \frac{(p(c_j) - p(G)) \times \vec{n}(c_j)}{\|(p(c_j) - p(G)) \times \vec{n}(c_j)\|}$ 
       $\phi_{dij} \leftarrow \max(1 - \frac{\|\vec{n}_j - \vec{n}_i\|^2}{\epsilon_{adapt}^2}, 0) + \max(1 - \frac{\|\vec{z}_j - \vec{z}_i\|^2}{\epsilon_{adapt}^2}, 0)$ 
       $\phi_{di} \leftarrow \phi_{dij}$ 
       $\phi_{max} \leftarrow \max(\phi_{max}, \phi_{di})$ 
  end procedure
    
```

---

<sup>37</sup> I have used a variant that incorporates the torque direction for the contact weighting, in addition to the normal. Here,  $p(G)$  denotes the COM of the body  $G$ .

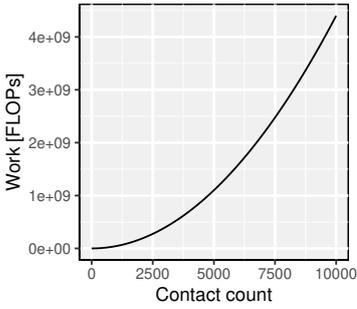


Figure 5.8: Computation of adaptive stiffness. That amount of FLOPs needed to compute the adaptive stiffness weights, depending on the contact count. The running time is truly exponential.

<sup>38</sup>For reference, a modern CPU core (AMD Ryzen 9 3950X) can compute around 226 billion FLOPs/s, or 226 million FLOPs/ms.

In many instances, I have found that this algorithm becomes a serious bottleneck in my simulation, in case the contact count is high during one frame. A presumed downside of the sphere packing algorithm is that the contact count can be very high occasionally, in which case these computations are not calculable in real-time. For example, this algorithm would incur the following amount of work for 3000 contacts:

- $3000 \times (3 + 3000 \times (3 + 3 + 3 + 1 + 1)) = 99009000$  additions
- $3000 \times (3 + 3 + 3000 \times (3 + 3 + 4 + 4)) = 126018000$  subtractions
- $3000 \times (6 + 3 + 3000 \times (6 + 3 + 3 + 3)) + 1 = 135027001$  multiplications
- $3000 \times (1 + 3000 \times (1 + 1 + 1)) = 27003000$  divisions
- $3000 \times (1 + 3000 \times (1)) = 9003000$  squareroots

which in sum are 396 million floating point operations (FLOPs)<sup>38</sup>, which need to be computed each simulation cycle. The overall needed FLOPs dependent on the contact count  $n$  amount to  $44n^2 + 20n + 1$  (see Figure 5.8). Consequently, solely computing the adaptive stiffness weights is difficult to do within haptic rates, if the contact count exceeds a few thousand contacts. Due to the volatile running time, I regard this method as an optional step that should be employed with great care, and thus, is not the default behavior of my system. However, including this feature with a prior threshold for the contact count might be feasible as well.

### 5.3.2 Impulse-Based Solving of Contact Constraints

Impulse-based simulations solely rely on impulses, which are instantaneous changes in velocity over an infinitely small time-frame, to resolve contacts. In fact, when formulating constraints on the velocity-level, this method can be used to iteratively solve these constraints, using a method called Gauss-Seidel solving. All contact constraints are solved independent of each other, leading to a local solution. When this procedure is repeated, the probability of delivering a global solution is increased.

I employ the impulse-based approach introduced by [Mirtich and Canny, 1995]. Given a contact point  $c_i$ , its location  $\mathbf{p}(c_i)$ , surface normal  $\vec{\mathbf{n}}(c_i)$  and the velocity of the tool at  $\mathbf{p}(c_i)$ ,

$$\vec{\mathbf{v}}(\mathbf{p}(c_i)) = \vec{\mathbf{v}}(T_G) + \vec{\boldsymbol{\omega}}(T_G) \times [\mathbf{p}(c_i) - \mathbf{c}(T_G)] \quad (5.31)$$

needs to point away from the other object's surface. This can be formulated in the following manner as a velocity-based separating constraint:

$$\vec{\mathbf{v}}(\mathbf{p}(c_i)) \cdot \vec{\mathbf{n}}(c_i) \geq 0 \quad (5.32)$$

A single separating constraint of this sort can be solved by using the elastic collision formula, that is part of classic physics, formulated by Isaac Newton. Though, there is a considerable chance that this adjustment will cause other contact points' separating constraint to be violated<sup>40</sup>, leading to the importance of repeating the procedure multiple times, until all separating constraints are met.

The handling of resting contacts (those are contacts where the relative velocity is 0 or nearly 0) is not well-defined, as there are various possible ways to deal with them. There are two inherit problems that arises when applying impulses for contacts with nearly 0 relative normal velocity:

1. There is an emergent behavior that bodies tend to not come to a complete rest, and "creep" along the surface of one another, even when without change to external forces.

<sup>39</sup>The surface normal needs to be reversed if they point away from the tool surface.

<sup>40</sup>Fortunately, any adjustment to velocity will not create new contact points, which is an advantage of velocity-based solving over position-based solving.

2. Bodies interpenetrate. Due to the loss of energy during elastic collisions, objects are always minimally moving towards the applied force, as the return force is not perfectly elastic (in general the coefficient of restitution (COR) is  $e \in [0, 1]$ ).

Especially the second issue is not acceptable for my use-case. Mirtich et al. [Mirtich and Canny, 1995] originally introduced the impulse-based simulation method. He, for example, classifies resting contacts based on a small point velocity magnitude threshold. Those resting contacts do not have a velocity constraint enforced, but rather so called micro-collisions are employed for those contacts. Micro-collisions simply resolve the reverse the relative normal velocity at one contact, therefore they do not actually do any work on the object. In a more recent publication, Bender [Bender, 2007] has used positional correction in those cases, to eliminate overlap. This approach is reminiscent of position-based dynamics [Müller et al., 2007], where the principal form and method of enforcing constraints is on the position-level. Here, velocities are subsequently updated based on the change in position and the time delta. Positional changes however entail a change in overlap, potentially creating new overlap, which can only be processed and resolved in the next frame. This is unacceptable for my use-case.<sup>41</sup> During my development I have found no convincing way to handle resting contacts with the impulse-based resolution method. In the algorithm presented here (see Algorithm 5.12), I have implemented Mirtich’s approach, by simply setting the COR to 0 for that iteration.

<sup>41</sup>This problem could be remedied by setting an upper limit of the positional correction that can be done per time-step, and selecting an appropriately conservative  $P_{\text{ContactMargin}}$  to find all potential contacts that could possibly be found due to a positional correction upper limit.

---

#### Algorithm 5.12 Impulse Contact Resolution

---

**Input:** contacts  $C$

**Result:** correct state variables of  $T_G$

**procedure** IMPULSECONTACTRESOLUTION

$\vec{f}_c, \vec{\tau}_c \leftarrow \vec{0}$

**for all**  $j \in [0, N^{42}] \in \mathbb{N}$  **do**

**for all**  $c_i \in C$  **do**

$\vec{r}_{T_G} \leftarrow (\mathbf{p}(c_i) - \mathbf{p}(T_G(c_i))) P_{\text{LeverMod}}$  ▷ torque lever

$\vec{r}_O \leftarrow (\mathbf{p}(c_i) - \mathbf{p}(O(c_i))) P_{\text{LeverMod}}$

$\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t \leftarrow \text{RELATIVEVELOCITIES}(T_G(c_i), O(c_i), \mathbf{p}(c_i), \vec{n}(c_i))$

**if**  $P_{\text{CollidingThreshold}} < \vec{v}_{rel} \cdot \vec{n}(c_i)^{43}$  **then** ▷ receding contact

**continue**

**if**  $-P_{\text{CollidingThreshold}} < \vec{v}_{rel} \cdot \vec{n}(c_i)$  **then** ▷ resting contact

$e_l \leftarrow 0$  ▷ micro-collision

**else** ▷  $\vec{v}_{rel} \cdot \vec{n}(c_i) \leq -P_{\text{CollidingThreshold}}$  ▷ colliding contact

$e_l \leftarrow e$

$\text{CALCULATEIMPULSE}(T_G, O, e_l, \vec{v}_{rel}, \vec{n}, \vec{r}_{T_G}, \vec{r}_O)$

$s \leftarrow \text{true}$

**for all**  $c_i \in C$  **do**

**if**  $\vec{v}_{rel} \cdot \vec{n}(c_i) \leq P_{\text{CollidingThreshold}}$  **then**

$s \leftarrow \text{false}$

**if**  $s$  **then break** ▷ constraints satisfied  $\Rightarrow$  early exit

**end procedure**

---

<sup>42</sup> $N = P_{\text{SimIterations}}$

<sup>43</sup>w.l.o.g. I assume the contact normal  $\vec{n}$  is pointing away from the obstacle, otherwise signs need to be flipped.

The actual calculation of the impulse (see Algorithm 5.14) is based on Baraff et al. [Baraff, 1994]. One advantage of the impulse method is the ease of integrating kinetic friction into the simulation. We simply apply an impulse in the opposite direction of the relative lateral velocity.

It could be interesting to extend this method to also support static friction, as presented by [Guendelman et al., 2003]. However, this would only become more relevant when working with multiple dynamic objects, and possibly stacking objects.

**Algorithm 5.13** Impulse Calculation

**Input:** body  $A^{44}$ ,  $B$ ,  $e \in \mathbb{R}$ ,  $\vec{n}, \vec{r}_A, \vec{r}_B, \vec{p} \in \mathbb{R}^3$   
**Result:** modified state variables of  $A$  and  $B$

```

procedure CALCULATEIMPULSE
   $a_1 \leftarrow \vec{n} \cdot \left( \mathbf{I}^{-1}(A) \left[ (\vec{r}_A \times \vec{n}) \times \vec{r}_A \right] \right)$ 
   $a_2 \leftarrow \vec{n} \cdot \left( \mathbf{I}^{-1}(B) \left[ (\vec{r}_B \times \vec{n}) \times \vec{r}_B \right] \right)$ 
   $a \leftarrow m^{-1}(A) + m^{-1}(B) + a_1 + a_2$ 
  if  $|a| > 0$  then
    // Normal impulse
     $\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t \leftarrow \text{RELATIVEVELOCITIES}(A, B, \vec{n})$ 
     $\vec{j}_r \leftarrow \frac{-(1+e) \vec{v}_{rel}}{a}$ 
    APPLYIMPULSE( $A, B, \vec{j}_r \frac{1}{N}, \vec{r}_A \times \left( \vec{j}_r \frac{1}{N} \right)$ )

    // Frictional impulse
     $\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t \leftarrow \text{RELATIVEVELOCITIES}(A, B, \vec{n})$ 
     $j_f \leftarrow \frac{\vec{v}_{rel}^t \cdot \vec{v}_{rel}}{a}$ 
    if  $j_f > 0$  then
      if  $|j_f| < \vec{j}_r \mu_s$  then
         $\vec{j}_f \leftarrow \vec{v}_{rel}^t j_f$ 
      else
         $\vec{j}_f \leftarrow \vec{v}_{rel}^t - j_r \mu_d$ 
        APPLYIMPULSE( $A, B, \vec{j}_f \frac{1}{N}, \vec{r}_A \times \left( \vec{j}_f \frac{1}{N} \right)$ )
    end if
  end if
end procedure

```

**Algorithm 5.14** Impulse Application

**Input:** body  $A, B$ ,  $\vec{m}_{lin}, \vec{m}_{ang} \in \mathbb{R}^3$   
**Result:** modified state variables of  $A$  and  $B$

```

procedure APPLYIMPULSE
   $\vec{v}(A) \leftarrow m^{-1}(A) \vec{m}_{lin}$ 
   $\vec{v}(B) \leftarrow m^{-1}(B) \vec{m}_{lin}$ 
   $\vec{\omega}(A) \leftarrow \mathbf{I}^{-1}(A) \vec{m}_{ang}$ 
   $\vec{\omega}(B) \leftarrow \mathbf{I}^{-1}(B) \vec{m}_{ang}$ 
end procedure

```

**Algorithm 5.15** Relative Point Velocities Calculation

**Input:** body  $A, B$ ,  $\vec{p}, \vec{n} \in \mathbb{R}^3$   
**Result:**  $\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t$

```

procedure RELATIVEVELOCITIES
  // Point velocities
   $\dot{\vec{p}}(A) \leftarrow \vec{v}(A) + \vec{\omega}(A) \times (\vec{p} - \vec{p}(A))$ 
   $\dot{\vec{p}}(B) \leftarrow \vec{v}(B) + \vec{\omega}(B) \times (\vec{p} - \vec{p}(B))$ 

  // Separate normal and tangential parts
   $\vec{v}_{rel} \leftarrow \dot{\vec{p}}(A) - \dot{\vec{p}}(B)$ 
   $\vec{v}_{rel}^n \leftarrow (\vec{v}_{rel} \cdot \vec{n}) \vec{n}$ 
   $\vec{v}_{rel}^t \leftarrow \vec{v}_{rel} - \vec{v}_{rel}^n$ 
end procedure

```

▷ relative velocity

<sup>44</sup>w.l.o.g. I assume  $A$  is the virtual tool  $T_G$ .

## 5.3.3 A Hybrid Solution for Contact Resolution

Due to the reasons mentioned in the previous section, resting contacts are not handled with satisfactory results. This has lead me to develop a hybrid approach between impulse- and penalty-based methods, since the penalty method handles resting contacts better. However, I still regard the handling of colliding contacts as superior in impulse resolution, as it is much simpler to incorporate convincing friction effects into the impulse method, compared to the penalty method.

Therefore, I decided to handle colliding contacts using the impulse method, and resting contacts with the penalty method. The idea is to solve velocity constraints of only colliding contacts, using impulses (similar to Algorithm 5.12). Afterwards, resting contacts are iterated and penalty forces are accumulated (similar to Algorithm 5.10). The algorithm is detailed in Algorithm 5.16.

**Algorithm 5.16** Hybrid Contact Resolution

**Input:** contacts  $C$

**Result:**  $\vec{f}_c, \vec{\tau}_c \in \mathbb{R}^3$ , correct state of  $T_G$

procedure HYBRIDCONTACTRESOLUTION

$\vec{f}_c, \vec{\tau}_c \leftarrow \vec{0}$

for all  $j \in [0, N^{45}] \in \mathbb{N}$  do

for all  $c_i \in C$  do

$\vec{v}_{T_G} \leftarrow (\mathbf{p}(c_i) - \mathbf{p}(T_G(c_i))) P_{\text{LeverMod}}$   $\triangleright$  torque lever

$\vec{v}_O \leftarrow (\mathbf{p}(c_i) - \mathbf{p}(O(c_i))) P_{\text{LeverMod}}$

$\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t \leftarrow \text{RELATIVEVELOCITIES}(T_G(c_i), O(c_i), \mathbf{p}(c_i), \vec{n}(c_i))$

if  $P_{\text{CollidingThreshold}} < \vec{v}_{rel} \cdot \vec{n}(c_i)^{46}$  then  $\triangleright$  receding contact

continue

if  $-P_{\text{CollidingThreshold}} < \vec{v}_{rel} \cdot \vec{n}(c_i)$  then  $\triangleright$  resting contact

continue

else  $\triangleright \vec{v}_{rel} \cdot \vec{n}(c_i) \leq -P_{\text{CollidingThreshold}}$   $\triangleright$  colliding contact

CALCULATEIMPULSE( $T_G, O, e, \vec{v}_{rel}, \vec{n}, \vec{v}_{T_G}, \vec{v}_O$ )

$s \leftarrow \text{true}$

for all  $c_i \in C$  do

if  $\vec{v}_{rel} \cdot \vec{n}(c_i) \leq P_{\text{CollidingThreshold}}$  then

$s \leftarrow \text{false}$

if  $s$  then break  $\triangleright$  constraints satisfied  $\Rightarrow$  early exit

for all  $c_i \in C$  do

$\vec{v}_{rel}, \vec{v}_{rel}^n, \vec{v}_{rel}^t \leftarrow \text{RELATIVEVELOCITIES}(T_G(c_i), O(c_i), \mathbf{p}(c_i), \vec{n}(c_i))$

if  $-P_{\text{CollidingThreshold}} < \vec{v}_{rel} \cdot \vec{n}(c_i)$  then  $\triangleright$  resting contact

$\vec{v}_{T_G} \leftarrow (\mathbf{p}(c_i) - \mathbf{p}(T_G(c_i))) k_l^{47}$   $\triangleright$  torque lever

$\vec{f} \leftarrow \frac{\vec{n}(c_i)}{\|\vec{n}(c_i)\|} \text{dop}(c_i) k_c^{48}$

if  $f_{max} > 0$  and  $f_{max} > \|\vec{f}_{max}\|$  then

$\vec{f} \leftarrow \vec{f} \frac{f_{max}}{\|\vec{f}\|}^{49}$

$\vec{f}_c \leftarrow \vec{f}$

$\vec{\tau}_c \leftarrow \vec{v}_{T_G} \times \vec{f}$

if  $F_{max} > 0$  and  $F_{max} > \|\vec{f}_c\|$  then

$\vec{f}_c \leftarrow \vec{f}_c \frac{F_{max}}{\|\vec{f}_c\|}^{50}$

end procedure

<sup>45</sup>  $N = P_{\text{SimIterations}}$

<sup>46</sup> w.l.o.g. I assume the contact normal  $\vec{n}$  is pointing away from the obstacle, otherwise signs need to be flipped.

<sup>47</sup>  $k_l = P_{\text{LeverMod}}$

<sup>48</sup>  $k_c = P_{\text{ContactStiffness}}$

<sup>49</sup>  $f_{max} = P_{\text{ContactForceLimitSingle}}$

<sup>50</sup>  $F_{max} = P_{\text{ContactForceLimit}}$

## RELATED PUBLICATIONS

<b>P3.</b>	<b>Realistic Haptic Feedback for Material Removal in Medical Simulations . . .</b>	<b>168</b>
<b>P9.</b>	<b>Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study . . . . .</b>	<b>228</b>
<b>P11.</b>	<b>Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks . . . . .</b>	<b>272</b>

## OUTLINE

6.1	Defining Coordinate Spaces and Transformations . . . . .	53
6.1.1	Transforming from Tool- to World-Space . . . . .	54
6.1.2	Transforming from Device- to World-Space . . . . .	54
6.1.3	Transforming from a Model- to Handle-Space . . . . .	55
6.1.4	Transforming from Handle- to Device-Space . . . . .	56
6.1.5	Putting It All Together . . . . .	56
6.1.6	Substitute A Physical Device by Camera Motion . . . . .	57
6.2	Automated Registration of Haptic- and VR-Space . . . . .	58
6.3	Mediating the Link Between User and Virtual Tool . . . . .	60
6.3.1	Improving Stability by Saturation of Virtual Forces . . . . .	62
6.4	The Force Rendering Algorithm . . . . .	63
6.4.1	Impedance Control – Render Forces . . . . .	63
6.4.2	Admittance Control – Render A Pose . . . . .	64

## MODULAR HAPTIC RENDERING

“To touch is to give life.” — Michelangelo

IN this chapter, I will explain what parts of the system are necessary to render convincing forces to the user, and how each of those parts are implemented. The *Haptics* thread’s work involve many transformations, therefore I will introduce the spaces between which we transform.

## 6.1 DEFINING COORDINATE SPACES AND TRANSFORMATIONS

The simulation system combines a multitude of spaces and regards haptic devices and game engines as interchangeable, therefore a robust and clear definition of the existing spaces and their transformations is essential for a robust implementation. The transformations that are necessary for the simulation mostly involve three main coordinate systems (and two contextually dependent transformations):

1. **World:** The game engine world space, in which the rendered meshes move in.
  - a) **Tool:** The tool pose before runtime *World* space.<sup>1</sup>
2. **Device:** The haptic device’s space, in which the haptic device API reports position and orientation of the haptic handle.
  - a) **Handle:** The haptic handle pose within *Device* space.
3. **Model:** The model space, in which the body’s geometry is modeled.

For a better understanding of the intuitive meaning of these spaces, see [Figure 6.2](#).

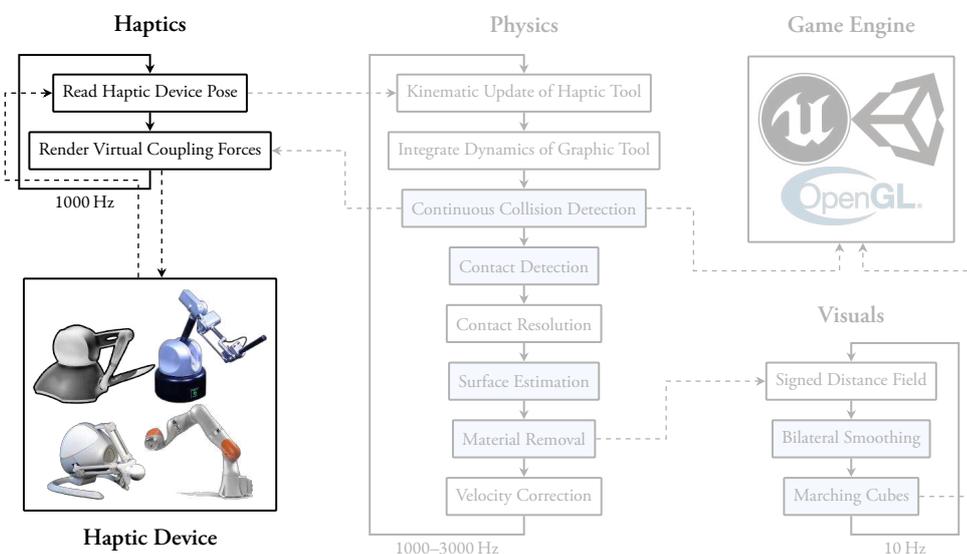
For the implementation of the haptics thread, the transformation between these spaces need to constructed as homogeneous transformation matrices. Defining just the following transformations will suffice:

1. *device to world*: depending on device and GE, rotations and scalings are necessary (see [Section 6.1.2](#)).

“These deeds must not be thought  
After these ways. So, it will make us  
mad.”

— William Shakespeare

<sup>1</sup>The purpose of this pose to enable the user to control the *Model-to-Device* and *Device-to-World* transformation from within the game engine editor, without modifying code.



**Figure 6.1: Integration of Haptic Rendering.** Visualization of the scope of this chapter and how it integrates into the whole system. The haptic rendering module reads the pose of the haptic device to move the *Haptic Tool*, thereby providing the interaction forces that act on *Graphic Tool*. Additionally, it calculates the forces that act upon the *Haptic Tool* and sends them to the haptic device API.

- a) *tool to world*: A translation depends on the scene, wherever the tool is placed (see Section 6.1.1).
2. *device-handle to device*: A haptics API reports the current pose relative to *device* as such (see Section 6.1.4)
3. *model to handle*: The developer can decide how to attach a model to the device handle (see Section 6.1.3)

These transformations can then be combined in a modular way to yield the pose of a *Haptic Tool* that is attached to the haptic device (details in Section 6.1.5).

### 6.1.1 Transforming from Tool- to World-Space

${}^W\mathbf{H}_T$  defines the transformation that is set inside the game-engine editor to be easily modified and drive the following transformations. The developer or even a user can easily manipulate the scene node that is used to define  ${}^W\mathbf{H}_T$  before the application is run, thereby making adjustments of how the tool is attached to the haptic handle and where the haptic device origin is placed in the game world. The scene nodes transformation when starting the game is then read and stored as  ${}^W\mathbf{H}_T$  and it will be used to compute the transformations that follow. However, during runtime, the transformation of the scene node (I will refer to it as  ${}^{W'}\mathbf{H}_T$  to denote that it is the temporary modified version at runtime) is overwritten by the application. Note that game-engines almost universally do not propagate transformation changes at runtime to the initial scene setup, therefore when the application is run the next time, it will be unchanged by the transformation changes that occurred at runtime due to movement of the tool.

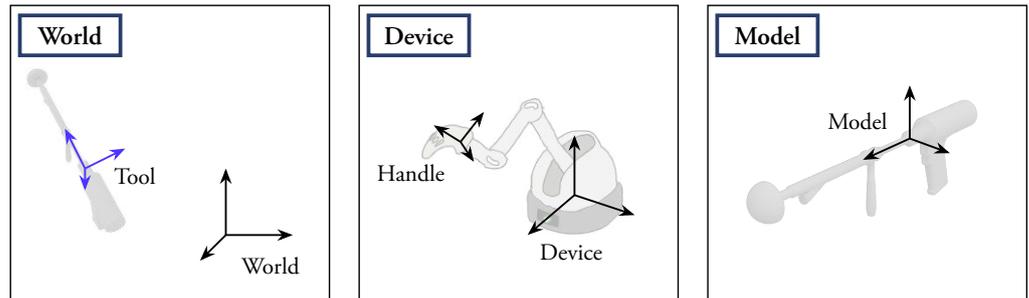
$${}^W\mathbf{H}_T \in \mathbb{R}^{4 \times 4} : \text{Defined inside the game-engine editor} \quad (6.1)$$

The game-engine then calls the library initialization with this transformation matrix, as it usually does not need to be updated at runtime, and it is being overwritten due to movement of the haptic handle anyway. The purpose of this transform will become clearer in the following explanations.

### 6.1.2 Transforming from Device- to World-Space

${}^W\mathbf{H}_D$  transforms from *Device* space to *World* space. This transformation on the one hand handles the conversion of the unit of lengths, as well as rotation and inversion of axes. Additionally,  ${}^W\mathbf{H}_D$  needs a mapping of the *Device*-origin in *World* space, which is something that should easily be able to be adjusted, instead of hard-coded. Therefore I decided to mention the *Tool* space earlier, as the transformation of *Tool-to-World* can usually be very intuitively controlled from within an editor of a game-engine and suffices to determine the *Device*-origin. The translational part of  ${}^W\mathbf{H}_T$  is used to position the *Device*-origin in *World*

Figure 6.2: Coordinate spaces to integrate a haptic device in a game-engine. The main coordinate spaces that need to be considered to attach a body at a haptic device handle and be placed inside of a game-engine world. This is the core task of the *Haptics* thread to process the input from a haptic device, and to process the interaction spring forces to output them to the haptic device.



space. In case the user makes a change to the scene in some way, for example moving objects around, the haptic device transformation is then automatically corrected in this way. Here are some transformations for combinations of game-engines and haptic devices that I have successfully integrated:

- Unreal from Chai3D

$${}^W\mathbf{H}_D = T({}^W\mathbf{H}_T) S({}^Wl_D) S(1, -1, 1) R_z\left(\frac{\pi}{2}\right) \quad (6.2)$$

- Unreal from KUKA

$${}^W\mathbf{H}_D = T({}^W\mathbf{H}_T) S({}^Wl_D) S(1, -1, 1) \quad (6.3)$$

- Unity from Chai3D

$${}^W\mathbf{H}_D = T({}^W\mathbf{H}_T) S({}^Wl_D) S(-1, -1, -1) R_x\left(\frac{\pi}{2}\right) R_z\left(\frac{\pi}{2}\right) \quad (6.4)$$

- Unity from Haption

$${}^W\mathbf{H}_D = T({}^W\mathbf{H}_T) S({}^Wl_D) S(-1, -1, -1) R_x\left(\frac{\pi}{2}\right) \quad (6.5)$$

Here,  ${}^Wl_D$  needs to be chosen based on the ratio of the units of length. Assuming a scene scale of 1 unit  $\hat{=}$  1 m, the game-engine can be disregarded, and  ${}^Wl_D$  solely depends on the used haptic device API in the manner detailed in Table 6.1.

Haptics APIs	${}^Wl_D$
Chai 3D (m)	1
KUKA (mm)	0.001
Haption (m)	1

### 6.1.3 Transforming from a Model- to Handle-Space

${}^H\mathbf{H}_M$  transforms from *Model* space to *Handle* space, thereby attaching the model to the haptic handle, with parameterization in the following way:

$${}^H\mathbf{H}_M = S\left(\frac{1}{{}^Wl_D}\right) R_z\left(\frac{\pi}{2}\right) R_x\left(\frac{\pi}{2}\right) S(-1, 1, 1) R({}^W\mathbf{H}_T) T(-c(T)) \quad (6.6)$$

As the translation  $T(-c(T))$  indicates, we attach the virtual tool with its center of mass at the origin of the haptic handle. I chose this as the general solution, as this is coherent with the user-interaction spring attachment point (details in Section 6.3), therefore the transformation of the rendered force can easily be derived from the spring force by inverting the direction and scaling the units, without an extra torque to translate an off-center force to the center of mass. In this way, the stiffness that can be used for the spring and the haptic device rendering is maximized.

If the tool should not be attached at its center of mass, but instead to some arbitrary attachment point  $\mathbf{p}_a$ , there are different ways of handling this situation.

- (i) Repositioning of the center of mass
- (ii) Computing an additional torque to translate the force
- (iii) Attaching the virtual spring at  $\mathbf{p}_a$

The easiest solution is (i): by moving the center of mass after the calculation of the physical properties (see Section 4.2) of the virtual tool to  $\mathbf{p}_a$ , we are still attaching the spring and applying forces at the same point. Option (ii) computes an additional torque  $\vec{\tau}_a$ :

$$\vec{\tau}_a = ({}^W\mathbf{H}_{H_i} (c(T) - \mathbf{p}_a)) \times \vec{f}_s \quad (6.7)$$

Table 6.1: Length scaling of haptics APIs. The length scaling of positional data from different haptic device APIs to a game-engine that uses meter as units for the scene.

where  $\vec{f}_s$  is defined in Equation (6.21) and  $\vec{\tau}_a$  should be added to  $\vec{\tau}_s$  in Equation (6.21). Lastly, (iii) proposes to attach the user-interaction spring that connects the *Haptic Tool* and *Graphic Tool* (see Section 6.3) at  $\mathbf{p}_a$  at each of the tools. The attachment point for the *Haptic Tool* is given by

$$\mathbf{p}_a = {}^W\mathbf{H}_H \mathbf{p}_a \quad (6.8)$$

and for the *Graphic Tool* by

$$\mathbf{p}_a = {}^W\mathbf{H}_G \mathbf{p}_a \quad (6.9)$$

The downside of this approach is significantly reduced stiffness that can be stably simulated, as the off-center spring introduces significant angular vibration.

#### 6.1.4 Transforming from Handle- to Device-Space

All transformations mentioned so far remain static during runtime of the simulation<sup>2</sup>. The only dynamic transformation in the *Haptics* chain of transformations is  ${}^D\mathbf{H}_H$ .

${}^D\mathbf{H}_H$ : Transform of haptic handle to haptic device base. This transformation is what a device API such as Chai3D provides, although usually in the form of a vector for translation and a quaternion for rotation and constructed in such a way:

$${}^D\mathbf{H}_H = \begin{pmatrix} \mathbf{M}_{00} & \mathbf{M}_{01} & \mathbf{M}_{02} & x(\mathbf{p}) \\ \mathbf{M}_{10} & \mathbf{M}_{11} & \mathbf{M}_{12} & y(\mathbf{p}) \\ \mathbf{M}_{20} & \mathbf{M}_{21} & \mathbf{M}_{22} & z(\mathbf{p}) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.10)$$

where the vector  $\mathbf{p}$  is the position and  $\mathbf{M}$  the rotation matrix constructed from the quaternion  $\mathbf{Q}$ , where  $\mathbf{p}$  and  $\mathbf{Q}$  are usually the chosen formats in which translation and rotation are reported by the haptic device API.

#### 6.1.5 Putting It All Together

Given these matrices, we can attach the virtual tool at the haptic handle by a chain of transformations. If the haptic device changes, some of the matrices need to be adjusted. Similarly, a change in the used game engine results in a need to adjust some of these transformations. However, all of these details can easily be handled by a configuration file that parameterizes the game-engine and haptic device used by enumerations. The final transformation  ${}^W\mathbf{H}_H$  that transforms the *Haptic Tool* from its *Model* space to *World* space at runtime is given by

$${}^W\mathbf{H}_H = {}^W\mathbf{H}_D {}^D\mathbf{H}_H {}^H\mathbf{H}_M \quad (6.11)$$

${}^W\mathbf{H}_H$  can then be used to locate the *Haptic Tool* inside the world. This is necessary for the physics in the simulation, and could also be used to visualize the tool inside the game-engine, in case that is the desired behavior. However, generally speaking displaying the *Haptic Tool* should not be the default behavior, as it is distracting for most laymen users of such systems. Firstly, displaying multiple tools will cause ambiguity as to which of them the user is supposed to focus on. Secondly, the *Haptic Tool* can (and will in almost all cases) penetrate the *Obstacle*, which could break immersion as it violates basic physical laws. Instead, graphically rendering the *Graphic Tool* and haptically rendering the forces the *Haptic Tool* (in other words the forces that would accelerate the *Haptic Tool* towards the *Graphic Tool*) will provide the user with the complementary information from both objects, that together create a coherent user experience. However, leaving the option to toggle on and off the display of the *Haptic Tool* might still be a useful option, in case people are curious as to how the rendered forces are computed.

<sup>2</sup>Although that does not mean that they should necessarily be hard coded, as I would argue that parametrization of them through a configuration text file or other means improves the generality of the application. It might even be necessary if simulation is implemented as a library with support for various game engines and haptic devices.

6.1.6 *Substitute A Physical Device by Camera Motion*

When the developer does not have access to a haptic device, he might want to fallback to other 6 DOF input devices to drive the user interaction in a way to steer the simulation. For example, such a device is the Space Mouse, which allows relative position in 3 translational directions and 3 rotational axes, thereby offering the same input precision as a 6 DOF haptic device, though the position is relative not absolute. Of course, the simulation output has to be analyzed by some method other than subjective force feedback evaluation, as that is only available through a haptic device (and only one with 6 DOF output, as 3 DOF output devices create completely incorrect feedback).

If the developer does not even have such a substitution device, there are no obvious choices. Therefore, I developed a simple software solution to steer the *Haptic Tool* with only conventional mouse and keyboard<sup>3</sup>, which I will call *Camera Clutch*. The idea is simple: Inside the game-engine, we control a camera via keyboard and mouse, like in many commercial games. Then there is an extra button that switches the clutch on or off. While the clutch is on, the *Haptic Tool* is moved as if it is attached to the camera, therefore any movement of the camera also controls the *Haptic Tool*. It is detached when the clutch is turned off, allowing for normal movement of the camera view, with a static *Haptic Tool*. All of this is integrated with the coordinate system transformation mentioned in Section 6.1, therefore one could even use the haptic device and this clutch mechanism in conjunction. Though I find it to be too imprecise (for example a rotational offset could diminish the realism of the haptic sensation) and not repeatable to make use of that too often.

**Capturing the Camera Pose** Many game-engines already implement a flying camera, or a similar camera paradigm that meets the following description, which can then be used instead: The camera viewing direction is rotated via mouse movement, as in any first-person game. The camera position is moved via 6 directional buttons (two for all three direction, each has one in the positive and one in the negative direction). This defines a pose in the game world, where the camera is currently located, called  ${}^W\mathbf{H}_C$ . Inside the game-engine loop, this update is reported to the *Haptics* thread each frame. The clutch switch from off to on and vice versa is reported as a single event from the game-engine input system to the *Haptics* thread.

**Applying the Clutch Motion Through Transformation** We store the total clutch transformation in  ${}^T\mathbf{H}_T$ , which essentially transforms from the haptic tool world base  ${}^W\mathbf{H}_T$  (details in Section 6.1.1) to an adjusted tool world base  ${}^W\mathbf{H}_{T'}$  that has all of the clutch transformations applied to it. Since the clutch transformation is initially zero, we initialize  ${}^T\mathbf{H}_T$  with the identity matrix. When the clutch is switched from off to on, we calculate the current transformation from *Haptic Tool* to Camera

$${}^C\mathbf{H}_{T'} \leftarrow {}^W\mathbf{H}_C^{-1} {}^T\Delta\mathbf{H}_T {}^W\mathbf{H}_{T'} \quad (6.12)$$

On this frame and consecutive frames, as long as the clutch is not switched off again, we update  ${}^T\Delta\mathbf{H}_T$

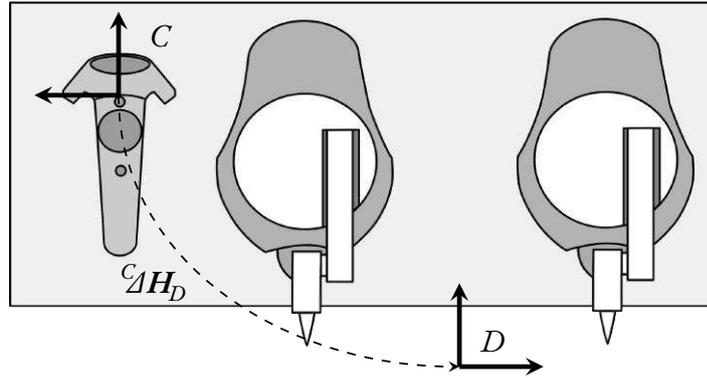
$${}^T\Delta\mathbf{H}_T \leftarrow ({}^W\mathbf{H}_C {}^C\mathbf{H}_{T'}) {}^W\mathbf{H}_{T'}^{-1} \quad (6.13)$$

Intuitively speaking,  $({}^W\mathbf{H}_C {}^C\mathbf{H}_{T'})$  is the haptic tool after clutch transformations, therefore in Equation (6.13), we calculate the delta transformation. The delta transformation can then be used this frame and consecutive frames (even when the clutch is switch off, and was off for an arbitrary amount of frames) to adjust  ${}^W\mathbf{H}_{T'}$

$${}^W\mathbf{H}_{T'} \leftarrow {}^T\Delta\mathbf{H}_T {}^W\mathbf{H}_{T'} \quad (6.14)$$

<sup>3</sup>Interestingly, most GEs (if not all) allow for only 5 DOF camera motion using mouse and keyboard, i.e. usually the roll-angle can never be modified. However, my camera clutch system still allows for full 6 DOF motion by repeatedly engaging and disengaging of the clutch. I do not provide a proof, however, one easily realizes that by rotating an object, with the clutch disengaged (i.e. by moving the camera around it), one will change the rotation axis for consecutive engaging of the clutch. Thus, one converts e.g. the pitch-rotation of the camera into a roll-rotation of the clutched object. In other words, the concatenation of  $x$  and  $y$ -rotations can create a  $z$ -rotation.

**Figure 6.3: Haptic device setup.** The registration of the haptic devices is simplified by mounting them, together with a VR controller, on predefined positions on a common surface. By mounting the haptic devices at a fixed offset, their local origin can be tracked in VR-space, enabling exact registration.



The adjustment in Equation (6.14) is applied after Equation (6.11), then the result is the described clutching behavior. The following C++ code implements the transformation algorithm:

```
void Haptics::cameraClutchTransformation( pd::Matrix4<float> &M_WfT )
{
    if (m_targetCameraClutch) {
        if (!m_targetCameraInitial) {
            m_M_CfT_ = !m_M_WfC * m_M_T_fT * M_WfT;
            m_targetCameraInitial = true;
        }
        const auto M_WfT_ = m_M_WfC * m_M_CfT_;
        m_M_T_fT = M_WfT * !M_WfT * M_WfT_ * !M_WfT;
    }
    M_WfT = m_M_T_fT * M_WfT;
}
```

Code 6.1: Camera tool clutch

## 6.2 AUTOMATED REGISTRATION OF HAPTIC- AND VR-SPACE

(based on [11])

In order to maximize spatial presence and learning efficacy, there should be a perfect spatial match between the virtual tools and the real handles of the devices, i.e., perfect visuo-haptic synchronicity [Freyberger et al., 2007]. Without registration of the haptic devices into the virtual world, only the relative motion of the virtual tools would be correct, but their position in the virtual world would be more or less offset from where the user would expect them due to the user’s kinaesthetic sense.

My registration procedure aims to minimize the positional and rotational error between the real haptic device handles and the virtual haptic tools, which are “attached” to the haptic devices so as to follow their movements. Importantly, I want to automate this registration, as manual registration is a time-consuming and tedious task, which might need to be repeated multiple times during a user study<sup>4</sup>. To facilitate this, we have mounted the haptic devices on a rigid wooden board, together with a VR controller (see Figure 6.3), thereby fixing the relative poses of both haptic devices and VR controller. Thus, the two haptic coordinate frames can be considered as one with the origin in the middle between them. We can move this origin with a fixed transformation into the virtual patient’s mouth, such that the virtual tools are to the left and right of the virtual patient’s mouth when the real haptic handles are held in a neutral position. The VR controller on the board serves as kind of an “extension” of the SteamVR tracking system to the haptic devices. What remains to be done is the transformation of the virtual camera such that offset between camera and virtual tools is exactly the same as the offset between the user’s HMD and the haptic handles.

<sup>4</sup>In fact, I have conducted a user study with manual registration of the haptic device with VR. Even just registering a single haptic device consumed about 10–20 min every morning, and whenever someone accidentally bumped into any of the assumed to be static VR equipment. On the other hand, two user studies with my automated registration proved its value, as this task was basically done in less than 10 s. In fact, people that were almost completely unfamiliar with my simulator were able to successfully conduct a user study with it, without me needing to be on sight.



In UE this can be accomplished by giving all VR objects (HMD camera and controllers) a common parent node with a non-zero transformation  ${}^H\mathcal{H}_H$ . We can now show  ${}^H\mathbf{H}_T = {}^H\mathbf{H}_D$ :

$$\begin{aligned} {}^H\mathbf{H}_T &= {}^W\mathbf{H}_H^{-1} {}^W\mathbf{H}_T = \left( {}^H\mathcal{H}_H {}^W\mathbf{H}_H \right)^{-1} {}^W\mathbf{H}_T \\ &= \left( {}^T\mathcal{H}_D {}^W\mathbf{H}_H \right)^{-1} {}^W\mathbf{H}_T = {}^W\mathbf{H}_H^{-1} {}^T\mathcal{H}_D^{-1} {}^W\mathbf{H}_T \\ &= {}^W\mathbf{H}_H^{-1} {}^W\mathbf{H}_D = {}^H\mathbf{H}_D \end{aligned} \quad (6.20)$$

Therefore, the real device is (relative to the user), at the same location as the virtual tool is (relative to the VR camera). Thereby, they are physically located and visually rendered at the same location.

In practice, the setup procedure by SteamVR (or others) provide a world coordinate frame well aligned with the (real) floor; we also assume that the table top is parallel to the floor. So, we only need to consider translation, and rotation along z in  ${}^H\mathcal{H}_H$ , which simplifies the manual registration part.

During registration, we encountered the issue that the haptic device handles and arms take up more space than the virtual tools, therefore they would interfere with each other during manipulation with both tools at the same time. Bi-manual manipulation is a necessary feature the simulator should support, as dentists will use the dental mirror during preparation to verify the bur depth, rotation, and position. To circumvent this issue, we have mounted the haptic devices at a distance of 30 cm, but placed the virtual tool origins at a distance of 20 cm and calibrated the midway point. Therefore, each virtual tool is 5 cm offset towards the middle from the actual haptic device handles, thereby avoiding the physical collisions and allowing true bi-manual control. I found this small translational alignment violation does not significantly hinder visual-motor coherence, as the difference is barely noticeable, even when being aware of it. During a user-study, which we conducted with this setup [Kaluschke et al., 2024], none of the participants noticed this or asked about a misalignment, therefore we presume that it did not have an adverse effect on the usability. This was also verified beforehand by three experts in dentistry who tried the simulator with the alignment violation.

### 6.3 MEDIATING THE LINK BETWEEN USER AND VIRTUAL TOOL

Given the poses of both instances of the virtual tool  ${}^W\mathbf{H}_{T_H}$  and  ${}^W\mathbf{H}_{T_G}$ , we can construct a spring. The purpose of the spring is to mediate the interaction of the user and the virtual environment in a controlled manner, the basic idea is inspired by the so called VC algorithm [Colgate et al., 1995; McNeely et al., 1999b]. There are several reasons to implement VC, for example it has been shown that VC can increase the stiffness that can stably be rendered by the haptic device [Colgate et al., 1995; Hou and Sourina, 2012]. Moreover, more elaborate tool dynamic schemes exist that could further improve stability, such as quasi-static approximation (QSA) [Wan and McNeely, 2003b] and “stable dynamic”-algorithm [Hou and Sourina, 2012]. It should be noted that these more elaborate dynamic schemes are not physically-based, and thus, they might introduce unrealistic forces, such as during movement in free space, in favor of stability. The stability improvement stems from the fact that the movement is controlled by parameters that the developers choose, such as force saturation, maximal velocity and damping. Contrary to that, during direct force rendering, the user and the physical parameters of the robot and the users arm determine these parameters (force saturation, maximal velocity and damping). Importantly, VC is the most intuitive way<sup>5</sup> of implementing a physically-based interaction inside of a physics simulation, and the idea would naturally come to one’s mind when developing algorithms to combine haptic rendering and a physics simulation. Here, I introduce additional damping parameters to further improve stability and allow for finer parameter tuning, which differentiates this method from the classic VC approach.

<sup>5</sup>VC might as well be the only feasible way to compute forces from a physics simulation, as far as I know.

Additionally, having two instances of the virtual tool provides an elegant way to implicitly store the "history" of the user's movement in the pose of the *Graphic Tool*. By resolving collisions and contacts with appropriate adjustments of the *Graphic Tool* pose (details in Chapter 5) we can guarantee that the *Graphic Tool* remains intersection-free, thereby storing the way out of the intersection, but influenced by the "history" of the user's movement. This helps avoid problems such as pop-through, which can arise when the direction out of the current intersection is computed on-the-fly, without regards for the "history" of the tool. Additionally, this opens up an intuitive way to implement linearized friction, as the state on the surface is stored in the *Graphic Tool*.

The user-interaction spring is attached to two bodies, the *Haptic Tool* and the *Graphic Tool*, depending on from which body's point of view the spring force  $\vec{f}_s$  is evaluated, the sign is flipped<sup>6</sup>, therefore we assume here the point of view of the *Haptic Tool*  $T_H$ :

$$\begin{aligned}\vec{f}_s &= [\mathbf{p}(T_G) - \mathbf{p}(T_H)] k_t^7 & - \vec{v}(T_H) b_t^{\text{abs}} & - [\vec{v}(T_G) - \vec{v}(T_H)] b_t \\ \vec{\tau}_s &= \vec{\tau}_t k_r^7 & - \vec{\omega}(T_H) b_r^{\text{abs}} & - [\vec{\omega}(T_G) - \vec{\omega}(T_H)] b_r\end{aligned}\quad (6.21)$$

with

$$\mathbf{p}(T_H) = {}^W\mathbf{H}_{T_H} \mathbf{c}_t \quad \mathbf{p}(T_G) = {}^W\mathbf{H}_{T_G} \mathbf{c}_t$$

$\vec{\tau}_t$  is computed from the rotational difference of  $T_H$  and  $T_G$  in their center of mass  $\mathbf{c}_t$ . To calculate it, we compute the rotational difference as a matrix  ${}^{T_G}\Delta\mathbf{H}_{T_H}$  by composing the *Haptic Tool* and *Graphic Tool*

$${}^{T_G}\Delta\mathbf{H}_{T_H} = {}^W\mathbf{H}_{T_G} {}^W\mathbf{H}_{T_H}^{-1} \quad (6.22)$$

with transformations of a body  $B$  being defined in the following manner:

$${}^W\mathbf{H}_B = \mathbf{T}(\mathbf{p}(B)) \quad \mathbf{R}({}^W\mathbf{Q}_B) \mathbf{S}({}^W\mathbf{H}_B) \quad (6.23)$$

$${}^W\mathbf{H}_B = \mathbf{T}(\mathbf{p}(B) - [\mathbf{R}({}^W\mathbf{Q}_B) \mathbf{S}({}^W\mathbf{H}_B)] \mathbf{c}(B)) \quad \mathbf{R}({}^W\mathbf{Q}_B) \mathbf{S}({}^W\mathbf{H}_B) \quad (6.24)$$

${}^W\mathbf{H}_{T_H}$  transforms  $\mathbf{0}$  to the tool's center of mass in world space, thereby it is the optimal point on the body to use to compute the rotational difference, thereby attaching the interaction spring between *Haptic Tool* and *Graphic Tool* at their respective center of masses. To clarify,  ${}^W\mathbf{H}_{T_H}$  on the other hand transforms  $\mathbf{0}$  to the tool's origin in world space. As the origin in model space is arbitrary, generally speaking, it does not hold a physical meaning and does therefore not suit for attaching the interaction spring to it.

The absolute damping terms  $b_t^{\text{abs}}$  and  $b_r^{\text{abs}}$  are additions on the known VC formula. I found that these terms can help stabilize the system, though they should be used sparingly, instead developers should prefer the relative damping terms, in case they show stable behavior, as they better eliminate unwanted damping, for example in intersection-free movement of the tool. In some cases, relative damping, especially relative angular damping  $b_r$ , did not suffice to stabilize the haptic rendering loop. In those cases using  $b_r^{\text{abs}}$  or a combination of  $b$  and  $b_r^{\text{abs}}$  could solve the instability.

I also found that in case that there is a common contact point with a large lever  $\vec{r}$  relative to the center of mass  $\mathbf{c}$ , for example the reaming head during acetabular reaming interaction, it can help to compute additional damping terms that are designed to stabilize the center of those common contact points  $\mathbf{p}_c$ , here the center of the reamer head. Given a lever

$$\vec{r} = \mathbf{c} - \mathbf{p}_c \quad (6.25)$$

and the linear velocity  $\vec{v}$  at the haptic handle center, we can compute a rotational damping  $\vec{b}_r^{\text{ext}}$  in the following manner:

$$\vec{b}_r^{\text{ext}} = (\vec{r} \times \vec{v}) b_r^{\text{ext}} \quad (6.26)$$

which is then additionally subtracted from  $\vec{\tau}_s$ .

<sup>6</sup>In the case of rotational difference, the order of transformations needs to be flipped.

<sup>7</sup>I have successfully used  $k_t = 12 \frac{\text{kN}}{\text{m}}$  and  $k_r = 24 \text{ Nm/rad} \approx 0.42 \text{ Nm/deg}$  in production (during the user study described in Section 9.2.2). Although the extend to which the stiffnesses can be increased primarily depends on the used haptics device.

## 6.3.1 Improving Stability by Saturation of Virtual Forces

The stiffness parameters  $k_t$  and  $k_r$  need to be select very high in practice, so that the interaction spring force  $\vec{f}_s$  does not exhibit noticeable inertia.  $k_t$  should be chosen as high as can stably be rendered (though this also depends on how high the contact forces can stably be scaled, as both forces should not be high enough to overpower the other, as an equilibrium state is needed for naturally feeling contact situations). Overly high inertia would create a sluggish and heavy feeling in the force rendering, which is usually not desirable. Additionally, haptic devices have a limited force range, which could not be allocated to rendering the contact interaction, as a noticeable amount of force would already be allocated for intersection-free movement. This “muddies” the force rendering, as the difference between intersection-free force and contact interaction force is significantly decreased. Thereby reducing the user experienced change in force due to their interaction. The equivalent audio sensation would be being spoken (in a regular speaking voice) to by another person during a loud concert. However, high stiffnesses also introduce some challenges for the simulation, namely

1. Destabilization of the simulation due to inaccuracy of integration
2. Complicate collision detection by large positional changes per frame

The first issue of instability can be addressed on the one hand by a low  $\Delta t$  for the integration of forces and velocities, so that integration remains stable and accurate. A low  $\Delta t$  in turn requires a high frequency, therefore the performance of the simulation is crucial for the user experience. Secondly, a saturation of  $\vec{f}_s$  also helps stabilize the simulation, as well as help limit the positional change per frame. As detailed in [Chapter 5](#), the collision detection is continuous, therefore pop through can not happen even with large steps. However, the rotational approach by the *Graphic Tool* towards the surface is mostly handled integration of  $\vec{\tau}_s$  at the surface position, therefore an upper bound to the integration steps is helpful for a smooth experience.

But most importantly, in the case of a resting contact, the *Haptic Tool* is penetrating the surface more as the user pushes further. Meanwhile, the *Graphic Tool* is constrained to the surface, and after finding a local minimum at the surface, it can no longer move to alleviate the spring force and torque. Consequently, without saturation, the force would grow larger and larger (as I mentioned earlier the spring stiffness should be chosen as high as can be simulated stably) and the system would become unstable and the forces too high and the *Graphic Tool* would move erratically. Additionally, the penetration depth would change a lot per frame, as the large forces would often create integration targets that overlap, which would be caught by the continuous collision detection, which will result in large velocity corrections (details in [Section 5.1.3](#)) due to violations of the integration target. The implementation is simple, the magnitude of  $\vec{f}_s$  and  $\vec{\tau}_s$  are clamped to reasonable positive values  $f_s^{\max}$  and  $\tau_s^{\max}$ :

$$\vec{f}_s \leftarrow \begin{cases} \vec{f}_s \frac{f_s^{\max}}{|\vec{f}_s|}, & f_s^{\max} > 0 \wedge |\vec{f}_s| > f_s^{\max} \\ \vec{f}_s, & \text{otherwise} \end{cases} \quad (6.27)$$

$$\vec{\tau}_s \leftarrow \begin{cases} \vec{\tau}_s \frac{\tau_s^{\max}}{|\vec{\tau}_s|}, & \tau_s^{\max} > 0 \wedge |\vec{\tau}_s| > \tau_s^{\max} \\ \vec{\tau}_s, & \text{otherwise} \end{cases} \quad (6.28)$$

It is important to tune the balance  $f_s^{\max}$  and  $\tau_s^{\max}$  with  $f_c^{\max}$  and  $\tau_c^{\max}$  (see [Section 5.3](#)), to achieve a responsive, yet smooth experience.

I have found, that many people intuitively think saturation is necessary to limit the force that is sent to the haptic device, as to not render a force that is too high for the device,

thinking it could be damaged. However, that is not at all the case. Firstly, haptic device APIs, at least all that I have ever used, handle forces that have a magnitude too high without problems. They might (i) ignore forces that are too high and simply render a force in the same direction with the maximum force that it can render or (ii) render the high force as long as the hardware can support it (for example as long as the current is not too high). I found a much better way to support the health of the (1) haptic device hardware and the health of the (2) user, is to limit the speed of the haptic handle (see Section 8.2). Thereby limiting the (1) amount of vibrations and (2) the irratic nature of the rendered force.

#### 6.4 THE FORCE RENDERING ALGORITHM

Haptic devices can be driven with two different methods of control, called called (i) impedance- and (ii) admittance-based control. These control methods are logical models that haptic device APIs implement in the form of interfaces to offer a logical way to control them. Therefore, some devices might support both control methods, or even more. I will explain how devices that support either of these methods can easily be driven by the framework that I layed out.

##### 6.4.1 Impedance Control – Render Forces

Impedance-based control seems to be the most common control type, therefore I am making use of the logical model through out this thesis.

The haptic device reports its haptic handle pose in the form of a position  $\vec{p}$  that resembles the translation and a transformation matrix  $\mathbf{M}$  that represents the rotation from the haptic handle base (see Section 6.1.4). This is the output of the haptic device. The input is expected as a force vector  $\vec{f}$  and a torque vector  $\vec{\tau}$  that represent the forces and torques that should be rendered at the haptic handle (the exact point of logical application depends on the haptic device). As detailed in Section 6.3, the interaction spring can be used to easily generate the rendering force

$$\begin{aligned}\vec{f} &= {}^D\mathbf{H}_W (-\vec{f}_i) \\ \vec{\tau} &= {}^D\mathbf{H}_W (-\vec{\tau}_i)\end{aligned}\quad (6.29)$$

The spring force and torque are calculated in Equation (6.21).

To optimize the stiffness and stability, it could be useful to have a separate set of stiffness and damping constants than those used to calculate  $\vec{f}_i$ . In that case, the rendered force is formulated in the following form (which is a modified version of Equation (6.21)):

$$\begin{aligned}\vec{f} &= {}^D\mathbf{H}_W \left[ (\mathbf{p}(T_G) - \mathbf{p}(T_H)) k_{D_i} \quad - \vec{v}(T_H) b_{D_i}^{\text{abs}} \quad - (\vec{v}(T_G) - \vec{v}(T_H)) b_{D_i} \right] \\ \vec{\tau} &= {}^D\mathbf{H}_W \left[ \vec{\tau}_i k_{D_r} \quad - \vec{\omega}(T_H) b_{D_r}^{\text{abs}} \quad - (\vec{\omega}(T_G) - \vec{\omega}(T_H)) b_{D_r} \right]\end{aligned}\quad (6.30)$$

The parameters  $k_{D_i}, b_{D_i}, b_{D_i}^{\text{abs}}, k_{D_r}, b_{D_r}, b_{D_r}^{\text{abs}}$  need to be then tuned during haptic rendering to achieve the desired haptic sensation. Using this separate set of parameters allows for finer tuning, as they do not directly affect the simulation behavior. Of course, indirectly, they still affect the simulation, but, generally speaking, they will not cause the simulation to become noticeably less stable.

This method was successfully implemented and tested on Chai3D devices, such as the Phantom Omni (6 DOF input, 3 DOF output, 3.3 N) (see Section 8.1 for an application) and on the Haption Virtuose 6D Desktop (6 DOF, 7 N). It was also tested on the new Haption Virtuose 6D with a force limit of 70 N (6 DOF) (see Section 8.2 for an application) and on the KUKA iiwa 14kg (6 DOF, 137 N).

*“I pull in resolution, and begin To doubt the equivocation of the fiend That lies like truth.”*

— William Shakespeare

## 6.4.2 Admittance Control – Render A Pose

Admittance control is more commonly found in control of robots than of haptic devices. In this control paradigm, the device receives a pose, which is desired by the application. In our case, this would be the pose of the *Graphic Tool*. The stiffness can usually still be controlled by the user, by a separate interface. For the damping, the user usually provides the velocity of the *Graphic Tool*. The final force, that is exerted by the robot or device, is computed by the device API. Given the logical framework, that I layed out in this chapter, the calculation of the desired pose becomes trivial. The simulation already calculates the pose of the *Graphic Tool*  ${}^W\mathbf{H}_{T_G}$  at all times and at an appropriate frequency. This pose now just needs to be transformed into *Device* space:

$${}^D\mathbf{H}_{T_G} = {}^W\mathbf{H}_D^{-1} {}^W\mathbf{H}_{T_G} {}^H\mathbf{H}_M^{-1} \quad (6.31)$$

This use-case was one of the reasons for me to rework my haptic rendering concept, such that it is completely matrix multiplication based. Each single step in the chain of transformations is trivially invertible, therefore to infer a haptic handle pose from an arbitrary world pose becomes easy to implement. When using other programmatically ways to process data from e.g. the device to the GE world<sup>8</sup>, it will render the inversion of such processing rather complex. In fact, when one has implemented this for multiple devices and multiple GEs, suddenly such a simple task becomes incredibly difficult to implement. At this point, I consider it a significant technological debt, when data is transformed without a proper theoretical foundation.

On the input side, either the actual pose is provided, as is the case in the impedance paradigm. Alternatively, the currently applied forces are sometime also provided, instead of a pose. This method was successfully implemented for the new Haption Virtuose 6D with 70 N.

<sup>8</sup>Methods that I have observed include swapping, inverting or other scaling of vector or quaternion components, creating matrices by inserting values or multiplying matrices without a theoretical foundation, or working with euler angles.



## Demo: Acetabular Reamer Interaction with Pelvic Bone

Video: <https://www.youtube.com/watch?v=0VYyJXuwcKY> (online; accessed 2024-04-11)

The video shows a demo of the simulation of a hip reamer with the pelvis, while forces on the reamer are haptically rendered using a Haption Virtuose in 6 DOF. The interaction renders the complete tool, remains always stable, does not allow pop-through, renders torques that feel physically-plausible and natural at frequencies exceeding the desired 1 kHz. It also supports material removal at the same haptic ready rate (more on that later in [Chapter 7](#)).

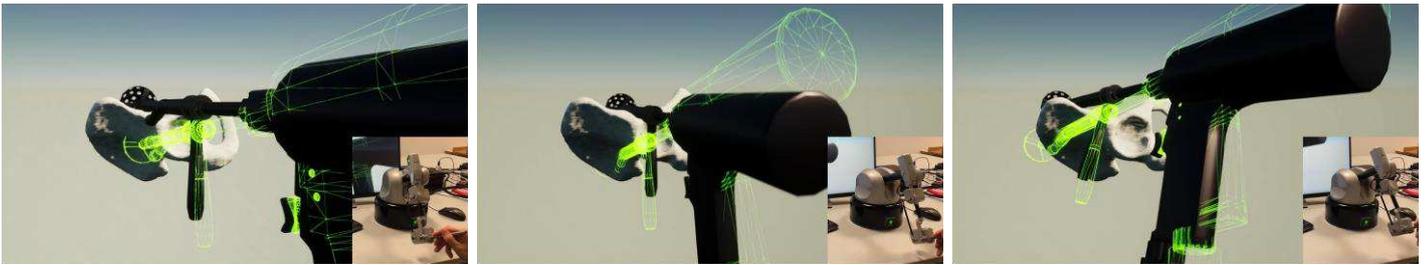


Figure 6.5: Small, complex tool features are also rendered. Sliding the hip reamer across the top surface of the pelvis. The angled reamer head intake (a small bump near the middle along the reamer shaft) collides with the hip and prevents further sliding. Green shows the haptic tool (see [Section 6.3](#)).



Figure 6.6: Smooth sliding. The other side of the shaft has no such features, and allows for smooth low-friction sliding along the top half of the pelvic bone. Green shows the haptic tool (see [Section 6.3](#)).



Figure 6.7: Arbitrary contacts. Any other configuration of the tool on the surface of the pelvis is possible. Even very unusual contact configurations (*Center*) can be stably rendered, without noticeable artifacts. Green shows the haptic tool (see [Section 6.3](#)).

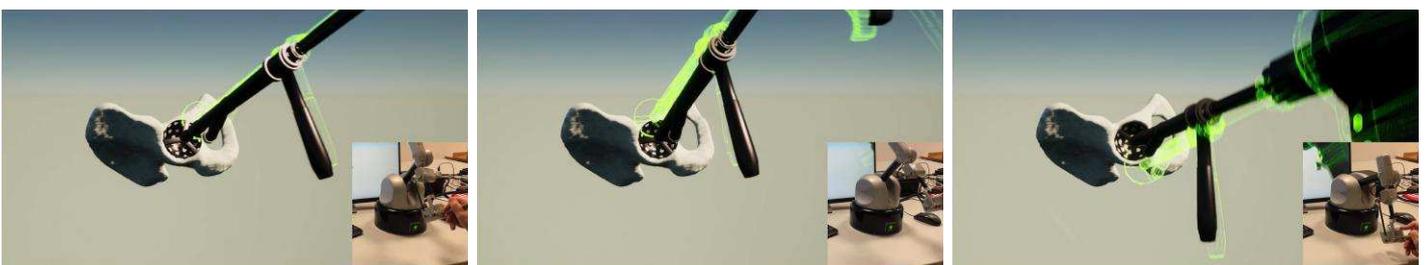


Figure 6.8: Reaming contact. The reamer head naturally slides into the hemisphere of the acetabulum and is constrained to it when rotating the tool. Here, material is also being taken off, though it is not clearly visible in screenshots. Green shows the haptic tool (see [Section 6.3](#)).

## RELATED PUBLICATIONS

<b>P3.</b>	<b>Realistic Haptic Feedback for Material Removal in Medical Simulations . . .</b>	<b>168</b>
<b>P5.</b>	<b>A cadaver-based biomechanical model of acetabulum reaming for surgical virtual reality training simulators . . . . .</b>	<b>188</b>
<b>P7.</b>	<b>A Shared Haptic Virtual Environment for Dental Surgical Skill Training . . .</b>	<b>212</b>
<b>P8.</b>	<b>The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training . . . . .</b>	<b>220</b>

## OUTLINE

7.1	Virtual Power Tools . . . . .	67
7.2	Defining A Local Cutting Model . . . . .	68
7.3	Representation of Removable Material . . . . .	70
7.3.1	Human Bone Model – Cortical Shell and Spongiuous Core . . . . .	71
7.3.2	Human Tooth Model – Three Layers . . . . .	72
7.4	Fast Continuous Removal of Material . . . . .	73
7.5	Variations on the Material Removal Algorithm . . . . .	74
7.5.1	Cleaning up the BVH during Material Removal . . . . .	75
7.5.2	Leveraging Purely Translational Motion for Continuous Removal . . . . .	77
7.6	The Visualization Pipeline . . . . .	77
7.6.1	Generating A Signed Distance Field From Spheres . . . . .	82
7.6.2	Fast Triangulation of the Signed Distance Field . . . . .	88

## MODELING 3D MATERIALS

“Foul whisperings are abroad: unnatural deeds  
Do breed unnatural troubles”  
— William Shakespeare

REPRESENTING all collidable objects as a dense packing of spheres, where no spheres shall overlap, is the foundation of my system. These spheres are augmented by additional properties, in order to facilitate the material removal process (see Figure 7.1 for details how material removal fits into the system as a whole). The two categories of objects are virtual tools and removable material.

## 7.1 VIRTUAL POWER TOOLS

In this section I will explain how virtual tools are represented in the material model. Virtual tools are the main mode of interaction that a user has in my simulation. Tools (i) generate virtual forces that can be felt in reality, and they can (ii) modify the geometry of other objects by removing material, while still producing forces. I will refer to these two types of interaction as *passive* and *active*. The focus of this chapter is *active* interaction, as the previous chapters have already fully detailed *passive* interaction.

The virtual tools require a distinction between volume that can remove material and volume that can not remove material. I will similarly refer to these two kinds of volume as *active* and *passive* volume. This distinction is important to allow some parts of tools to contribute to the physical interaction between objects, while not removing material. One might argue that allowing all volume to remove material would make for a simple and sensible solution. However, akin to reality, tools usually have a large amount of parts that are not at all designed for cutting performance, such as a handle and other auxiliary housing. The relevance of this even more clear when some parts of tools are used by surgeons to constrain the cutting result in some way. A good example for this can be observed during root-canal access opening: the hand-piece head can't enter a tooth cavity due to its size, thereby preventing drilling deeper than the bur length (see Figure 7.2). The bur length is purposefully designed with this in mind to minimize possible drilling mistakes. Less importantly, during caries removal, the conical bur shaft might constrain the bur head from

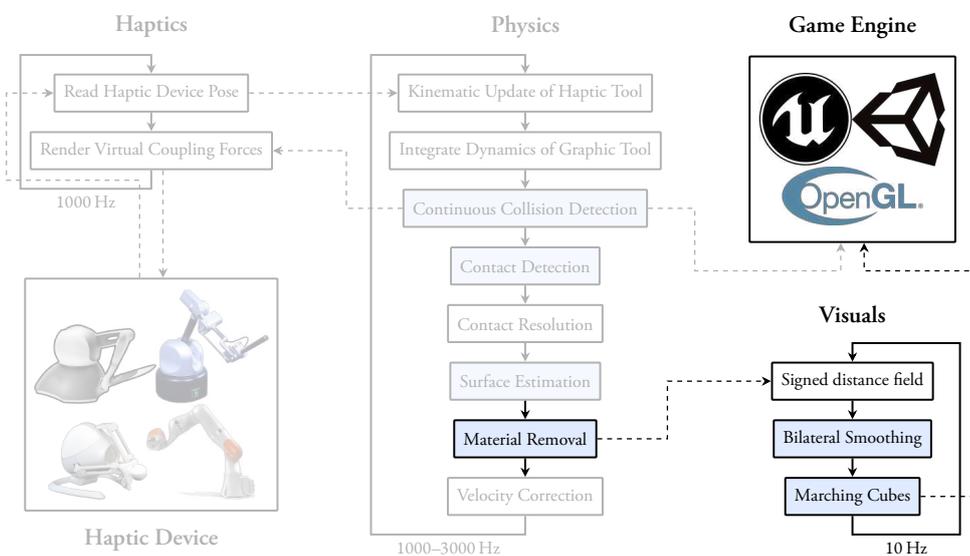
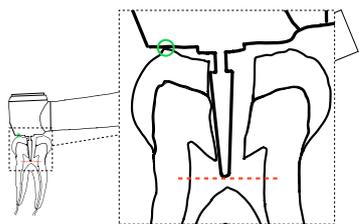
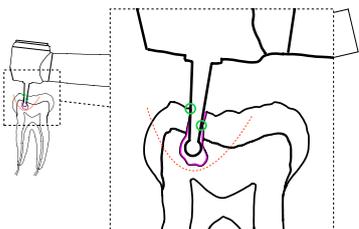


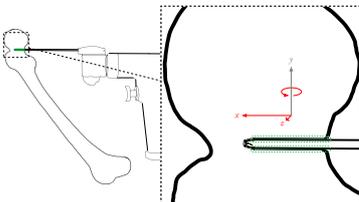
Figure 7.1: Integration of material removal & visuals. Visualization of the scope of this chapter and how it integrates into the whole system. The material removal is part of the physics loop and continuously decimates removable material of such objects. In a much slower loop, the visualization generates a dynamic mesh from the dynamic material model to represent the current state of the material. The dynamic mesh is finally drawn using the GE renderer, thus, this chapter has some parts that need to be implemented per GE, in case one supports multiple GEs.



**Figure 7.2: Passive contacts in root canal.** The handpiece head makes passive contact with the tooth surface (green circles), which constrains the bur depth (red dashed line), thereby reducing risk of perforation.



**Figure 7.3: Passive contacts in caries removal.** The conic shaped bur shaft makes passive contact (green circles), which constrains the bur head's movement and depth (red dashed line), thereby constraining the material removal. Pink surface is infected by caries.



**Figure 7.4: Passive contacts in femur head sawing.** Passive contacts along the blade body (marked by dashed green border) constrain the saw movement down to 3 DOF movement in the cutting plane (normal to  $y$ ): (1) translation along  $x$  and (2)  $z$ , and (3) rotation around  $y$ . Thus, translation along  $y$  and rotation around  $x$  and  $z$  are blocked.



**Figure 7.5: Passive contacts during acetabular reaming.** Elastic tissue and retractors around the operation sight constrain the acetabular reamer shaft, and thereby offers haptic guidance and leverage. Image reproduced from [Das, 2020].

reaching further down or to the sides, if the tooth opening is too small (see Figure 7.3). Another example is acetabular reaming, in which the contact of the reamer shaft against the skin and retractors might help leverage off or to constrain the angle of reaming (see Figure 7.5). Though the most significant example can be seen when sawing off the femur head during hip surgery. In this case, numerous passive contacts along the lengthy body of the saw blade establish a cutting plane which constraints the tool movement to in-plane movements (see Figure 7.4). These are just the examples in simulated surgery tasks, that I have worked on. I suppose many, if not all, simulated surgery task would benefit from passive contacts. Passive contacts provide context and haptic guidance for the surgeon, and therefore, I regard them as crucial to be supported in a good simulator.

The two virtual tools in which I have implemented the concept of active and passive contacts are the femur bone saw and acetabular reamer, both of which are used during total hip arthroplasty. I distinguish the volumes which are passive and active by supplying a manually designed bounding mesh for the active parts, as those tend to be the minority (see Figure 7.6 Center and Figure 7.7 Center).

It should be noted, that haptically rendering tools with arbitrary geometry, such as complex tools like the acetabular reamer or the bone saw, it is only feasible to render to a 6 DOF haptic device, which supports torque rendering. Without torque rendering, but allowing the user interaction spring (see Section 6.3) to be wound up rotationally, the user can not feel the spring being wound up and thus the haptic connection between tool and user is significantly flawed. This feels noticeably confusing and unintuitive when interacting with such a 6 DOF simulation using a 3 DOF haptic device. A feasible way to mitigate such an issue could be to make the user interaction spring as stiff in the rotational interaction, as possible. This is physically plausible, since in a normal 6 DOF interaction, when the tool does not follow the haptic device, the degree to which it does not follow is proportionally converted to a force, that is felt by the user. This force provides the user with haptic information that the tool is resisting in a specific DOF, indicating, that e.g. the environment is hindering such movement<sup>1</sup>. When such information can not be transmitted, as the device is unable to do so, it would only seem natural to not allow the environment to enforce such limitations on the tool. By increasing the stiffness significantly above the stiffness of contacts, the user interaction spring force or torque would likely overpower any contact force. Another possibility is to simply tune down contact torques, as this would have a similar result. For example, an off-center impulse could only incur an angular impulse that is 5% of the physically correct angular impulse. Thus, colliding contacts would mostly be resolved by translational impulses. However, I have not tested the latter approach, whereas the former is used in the dental VR simulator (see Section 8.1), as it uses two 3 DOF devices.

## 7.2 DEFINING A LOCAL CUTTING MODEL

Both aforementioned tools are designed to cut away material, though, the manner in which they cut material is distinctly different. Therefore, I decided to not only tag spheres as *active* and *passive*. I also augment each active sphere with a *drill direction*, which will determine in which direction they can remove material, similar to *normal cones*<sup>2</sup> (mentioned in Section 5.3.1). I have generated these drill directions using a simple function: I define a *drill center*  $c_d \in \mathbb{R}^3$  and each sphere  $s_i$  is given the normalized direction from the drill center to the sphere center  $c(s_i)$  as its drill direction  $\vec{d}(s_i)$ :

$$\vec{d}(s_i) = \frac{c(s_i) - c_d}{\|c(s_i) - c_d\|} \quad (7.1)$$

The drill centers I chose for the two THA tools are (in mm):

- **Acetabular reamer:**  $c_d = (-300.96, 0, 0)^T$  is located at the center of the reamer head.

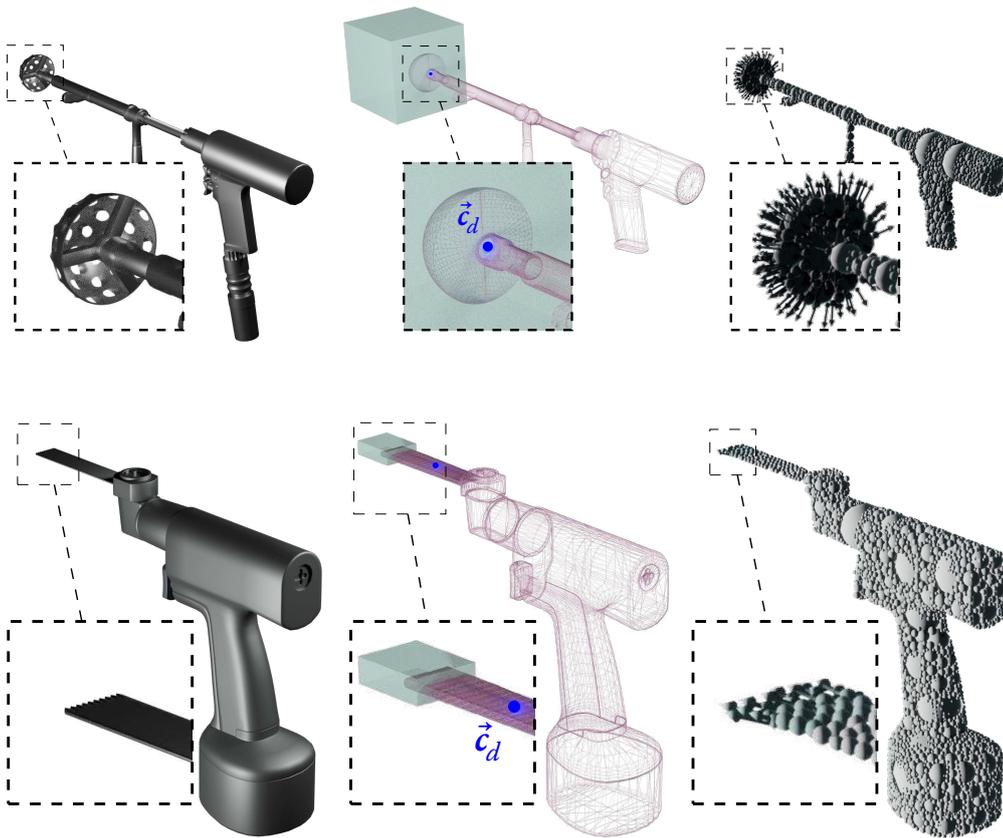


Figure 7.6: Acetabular reamer. The virtual tool of the acetabular reamer for total hip arthroplasty (THA).

Left Ground truth mesh, provided by project partner.

Center The collision mesh; note the simplified rotational extrusion of the reamer head. This mesh is used for the sphere packing generation, and turquoise cube tags active spheres.

Right The resulting sphere packing (2k spheres), with arrows on active spheres for their drill direction.

Figure 7.7: Femur bone saw. The virtual tool of the femur bone saw for THA.

Left Ground truth mesh, provided by project partner.

Center The collision mesh; note the slightly thickened and rotational extrusion of the saw blade. This mesh is used for the sphere packing generation, and turquoise cube tags active spheres.

Right The resulting sphere packing (5k spheres), with arrows on active spheres for their drill direction.

- Femur bone saw:  $c_d = (-177.87, 0, 214.9)^T$ <sup>3</sup> is located at the center of the blade body.

The needed geometry and the resulting augmented sphere packings for the acetabular reamer and femur bone saw can be seen in Figure 7.6 and Figure 7.7. The sphere packing augmentation is done in a small program (see Code 7.1) that calls function of my simulation library, to import, augment and export a sphere packing.

```

auto reamer = new SphereTree<float>( "Reamer.obj.101029.spheres",
    Matrix4<float>::Identity(), 1, 2000, 1 );
reamer->setActiveWithDirectionFromMesh( "Reamer_DrillMesh.obj", 1, "
    Reamer_DrillOrigin.obj" );
reamer->exportHierarchy( "Reamer.obj.2000.tree" );

auto saw = new SphereTree<float>( "Saw.obj.104716.spheres", Matrix4<
    float>::Identity(), 1, 5000, 1 );
saw->setActiveWithDirectionFromMesh( "Saw_DrillMesh.obj", 1, "
    Saw_DrillOrigin.obj" );
saw->exportHierarchy( "Saw.obj.5000.tree" );

```

Code 7.1: Tool sphere packing augmentation

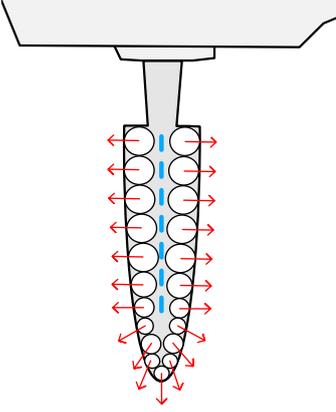
The same procedure could be used to describe spherical burs for dental surgery simulation, with  $c_d$  set as the sphere center. Though, in case it is in fact a perfect sphere, it would be much simpler to use a single sphere and disable direction material removal, as it can drill omnidirectionally. A conical bur could be described by defining a straight line segment along the main axis of the cone and computing the closest point on that line for each sphere and using this direction as the drill direction (see Figure 7.8).

To make the point velocity calculations (see Equation (5.31)) more accurate, we can use the active tag of spheres to determine whether they are locally rotated. We can approximate the cutting speed and add it to the angular velocity, if the virtual tool is cur-

<sup>1</sup>A simple example is when the bone saw during cutting of the femur head has many DOF disabled, when being inserted into a previously cut slit (see Figure 7.4).

<sup>2</sup>One might wonder whether *drill direction* suffers from the same problems as I mentioned exist for *normal cones*. No it does not, as (1) cutting always happens at the outside of tools, therefore, it only needs to be defined on the outside, and (2) my simulation enforces an intersection-free state, meaning contacts on the inside should not occur.

<sup>3</sup>One might also choose a constant direction for the bone saw, with only a small degrade in drilling quality.



**Figure 7.8: Conical drill directions.** How a conical dental bur could define the drill directions of the active spheres. The graphic shows select active spheres, with the drill direction as red arrows. The drill direction is derived from the closest point on a line segment on the principal axis of the cone (dashed blue line).

rently cutting, which can be controlled for example by a button press on the haptic device. The cutting speed of the reamer can be calculated by taking the rotation speed of around 270 RPM [Sherman et al., 2021], which converts to  $28.27 \frac{\text{rad}}{\text{s}} = s_d$ . The axis of rotation is  $\vec{r}_d = (1, 0, 0)^T$ , the local x-axis. Similarly, one could add an angular velocity to the saw blade when it is oscillating. A sagittal bone saw blade is moved at around 10 000–20 000 oscillations/min [James et al., 2013]. For a typical oscillation range of 3–5°, this equates to roughly  $166.67 \text{ RPM} = 17.45 \frac{\text{rad}}{\text{s}} = s_d$ . The axis of rotation is the local z-axis, therefore  $\vec{r}_d = (0, 0, 1)^T$ .

These angular velocities are then added to the point velocity in the following manner, if cutting is enabled, for each contact point  $c_i$  that involves an active sphere  $s_j$ :

$$\dot{\mathbf{p}}_{\text{cut}}(c_i) = \left[ \left( \mathbf{c}_d + \frac{(\mathbf{p}(c_i) - \mathbf{c}_d) \cdot \vec{r}_d}{\vec{r}_d \cdot \vec{r}_d} \vec{r}_d s_d \right) - \mathbf{p}(c_i) \right] \times \vec{r}_d \quad (7.2)$$

$$\dot{\mathbf{p}}_{c_i} = \dot{\mathbf{p}}_{c_i} + \dot{\mathbf{p}}_{\text{cut}}(c_i) \quad (7.3)$$

I have also used a weighting factor  $w_d \in [0, 1]$  for  $\dot{\mathbf{p}}_{\text{cut}}(c_i)$  dependent on the inverse of the overall overlap volume of active contacts  $V_D$  (see Section 5.2.1). This will modulate with how much force the user is pushing the tool into contact, leading to larger cutting friction proportional to the force with which one pushes. This can be expressed in following function

$$w_d = \text{clamp}(1E^{10} V_D, 0, 1) \quad (7.4)$$

where the factor  $1E^{10}$  is manually tuned and scales the contact volume to be close to 1 at the typically deepest contact situation.

Although adding velocity to contacts might make solving the velocity constraints more difficult<sup>4</sup>, it enables the emergence of turbulent forces that realistically occur from high friction during cutting. Take note though, that this only works if the used contact resolution approach handles dynamic friction well. I have found this to be true only for the case for the impulse-based and my hybrid approach, which is the main reason to not use a penalty approach, since all my attempts at combining cutting friction with it have not worked out. Although Zhao et al. [Zhao et al., 2018] have added computation of even static friction to a penalty-based approach, this friction computation adds 5-10x the computation time of collision detection. The resulting approach stays within haptic rates only for 100 contacts or less, which my application can easily exceed. For example, when the reamer head is tightly enclosed by the hip socket, the two perfect hemispheres enter hemispherical coupling, making it a worst case scenario for the contact count. It is also possible that force-based solving, which is core to penalty approach, simply do not generate appropriate simulation state changes enough to keep up with such quickly changing frictional contacts. The friction impulses are instantaneous changes in velocity, which can create a haptical ”kick” to the haptic device, which feels very natural, without instability.

### 7.3 REPRESENTATION OF REMOVABLE MATERIAL

In the earlier sections, I have only discussed the virtual tools. However, the other component to the interaction is the the object which can the tool is working on, which I have called environment ( $E$ ) earlier. The significant difference is that the environment represents removable material. In the medical field this is typically organic material, which is part of the human body. These materials can be complex, and therefore, I will represent this complexity in the spheres by augmenting the sphere packings. In this section I will present how the organic materials human bone and human teeth can be implemented using sphere packings. An environment sphere  $s_i$  has the following additional fields<sup>5</sup>:

1. Layer tag  $\gamma(s_i)$

<sup>4</sup>It might be feasible to simply ignore these added velocities during the application of the normal impulse, and to only use them during the frictional impulse.

<sup>5</sup>These fields could be implemented using unions with the fields added to tool spheres, although I have not done that yet.

2. Density modifier  $\rho_{mod}(s_i) \in \mathbb{R}$
3. Density  $\rho(s_i) \in \mathbb{R}$
4. Color  $\vec{c}(s_i) \in \mathbb{R}^3$

where the layer tag is an enum with the value as one of the following

$$y(s_i) \in \{ \text{Undefined, Normal, Empty,} \\ \text{Caries, Enamel, Dentin, Pulp, PulpChamber,} \\ \text{CorticalBone, SpongiusBone } \}$$

Layer tags and colors are used during the visualization procedure to determine how the current surface will optically look like. A triangle will have the layer tag that it was generated from and all triangles associated to a particular layer can be rendered using a corresponding shader. The color field allows procedural generation of persistent 3D volumetric colors, according to the layer (for more details see [Section 7.6](#)). The color parameter might not be necessary in all cases, since I could imagine such colors could also be generated procedurally at runtime, though my approach is certainly simpler. Densities are mostly used to locally modulate the speed at which material is removed (for details see [Section 7.4](#)).

### 7.3.1 Human Bone Model – Cortical Shell and Spongius Core

In this section I will discuss how I augmented sphere packings for removal material used in THA. In particular, I have generated a representation for pelvic bone and femur bone, which are worked on during THA.

Although these are distinct bones, with special geometry and properties, the overall model should generalize to many, if not all, bones. The human bone typically has a thin harder outside layer called cortical bone. On the inside, the bone is much softer, called spongius bone. As such, I generated a sphere packing for the whole volume, tagged all spheres as spongius bone, and afterwards tagged all spheres near the surface as cortical bone.

The thickness of the cortical bone for the pelvis is on average  $1.77 \pm 0.69 \text{ mm} \in [0.3, 8.3]$ , though it varies across the surface [[Giudice et al., 2018](#)], for example, it is especially thin at the acetabulum (see [Figure 7.9](#)). This variation is even larger for femoral cortical bone thickness, depending on the subject's age, though the average generally is around  $4.2 \text{ mm} \in [1.0, 8.7]$  [[Du et al., 2018](#)]. The head portion of the femur has much lower cortical bone thickness ( $0.917 \pm 0.017 \text{ mm}$ ) compared to the shaft ( $8.700 \pm 0.019 \text{ mm}$ ) [[Du et al., 2018](#)]. I have represented these thicknesses and local variations in my model (see [Figure 7.9](#)) The sphere packing augmentation is done in a small program (see [Code 7.2](#)).

```
auto pelvis = SphereTree<float>( "Pelvis.obj.125112.spheres", Matrix4<
float>::Identity(), 1, 100000, 1 );
pelvis.setTag( MaterialType::SpongiusBone );
pelvis.setTagFromMeshOverlap( "Pelvis_Shell.obj", MaterialType::
CorticalBone );
pelvis.exportHierarchy( "Pelvis.obj.100000.tree" );

auto femur = SphereTree<float>( "Femur.obj.105517.spheres", Matrix4<
float>::Identity(), 1, 100000, 1 );
femur.setTag( MaterialType::SpongiusBone );
femur.setTagFromMeshOverlap( "Femur_Shell.obj", MaterialType::
CorticalBone );
femur.exportHierarchy( "Femur.obj.100000.tree" );
```

Code 7.2: Human bone sphere packing augmentation



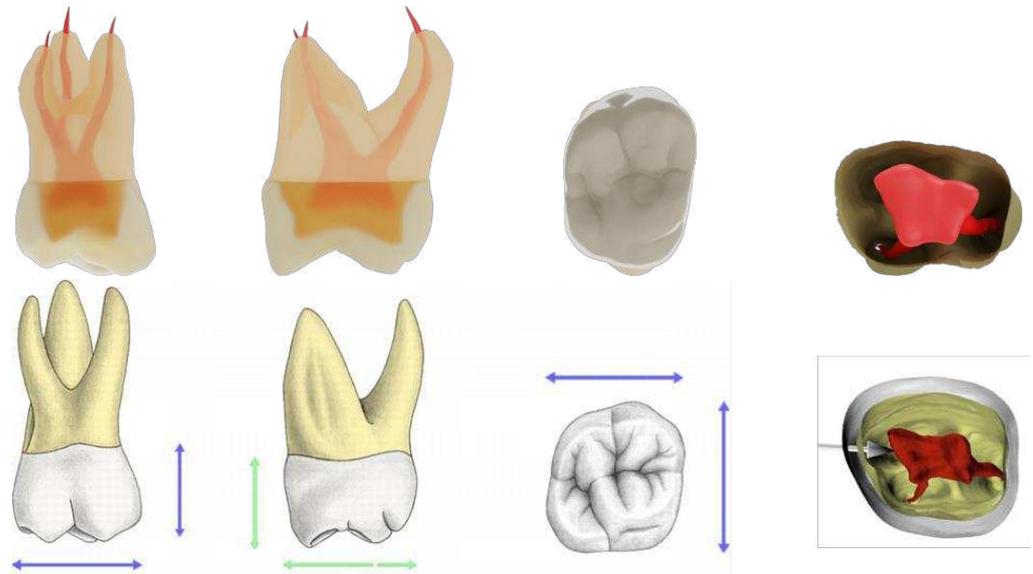
Figure 7.9: Human bone model. An outer layer (rendered in black) represents harder cortical bone, which surrounds a softer inner layer (shown in red) of spongius bone, which fills the whole rest of the geometry. The thickness of the outer layer varies locally, which is more pronounced in the mid section of the shaft of the femur.

Top Pelvis bone.

Bottom Femur bone.

**Figure 7.11: Tooth anatomy.** The real tooth anatomy that I used as a reference when designing my tooth model. *Top row* shows my tooth model in similar setting as the references in the *bottom row*, as shown by Dhaimy et al. [Dhaimy et al., 2019]. The references are digitized CT-scans of real data.

Left Buccal view.  
 Left-Center Mesial view.  
 Center-Right Crown view.  
 Right Pulp chamber.

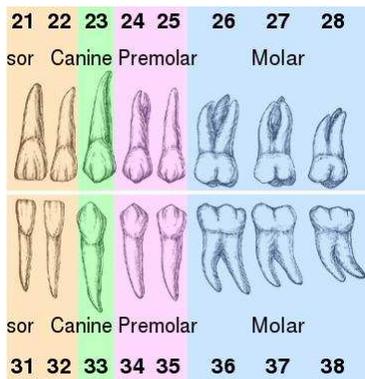


**Figure 7.12: Anatomical tooth model.** My model of tooth #16, manually designed based on CT data.

Left The three layers, enamel (top), dentin (yellow, bottom) and pulp (red, middle).  
 Left-Center Auxiliary meshes used in Code 7.3 to classify pulp chamber and roots.  
 Center-Right The resulting sphere packing with colors and tags (beige enamel, yellow dentine).  
 Right Cross-section of voxelized form; the pulp is well represented in the packing (bright red pulp chamber, dark red roots).



### 7.3.2 Human Tooth Model – Three Layers



**Figure 7.10: Unique teeth in dentition.** The subset of teeth that are unique, the rest can be generated through mirroring.

Compared to human bone, the tooth model is more intricate, because the internal anatomy is much more complicated. Human teeth have root canals that connect the pulp chamber with internal blood vessels and nerve endings to the body. I have designed teeth #26 and #36, since we focused on those in two user studies (see Section 9.1). However, due to human teeth anatomy, each model can construct the left and right side teeth (see Figure 7.10). I have designed the teeth in accordance to root canal anatomy of first maxillary molars as described by Dhaimy et al. [Dhaimy et al., 2019]. This includes crown geometry, pulp chamber morphology, and root count, shape and placement (see Figure 7.11). I created two distinct spherepackings with around 100 k spheres each<sup>6</sup>, followed by augmenting the sphere packings. Augmentation involves tagging spheres according to its layer, and additionally tagging dentin that surrounds the pulp, and dentin that surrounds the pulp chamber (see auxiliary geometries in Figure 7.12 *Left-Center*). The code that augments the sphere packings is described in Code 7.3.

```
enamel = new SphereTree<float>( "Tooth16_Enamel.obj.104526.spheres",
    Matrix4<float>::Identity(), 1, 100000 );
enamel->setTag( MaterialType::Enamel );
enamel->exportHierarchy( "Tooth16_Enamel.obj.100000.tree" );

dentin = new SphereTree<float>( "Tooth16_Dentin.obj.106395.spheres",
    Matrix4<float>::Identity(), 1, 100000 );
dentin->setTag( MaterialType::Dentin );
```

<sup>6</sup>I usually also filter sphere packings for outside spheres by odd ray hits from each sphere center against the surrounding mesh.

```

dentin->setTagFromMeshWithTolerance( "Tooth16_Pulp.obj", MaterialType
::Pulp, 2.0 );
dentin->setTagFromTriMeshWithTolerance( "Tooth16_Pulp.obj", "
Tooth16_Pulp_Cube.obj", "Tooth16_Pulp_Chambersphere.obj",
MaterialType::Pulp, MaterialType::PulpChamber, 2.0 );
dentin->exportHierarchy( "Tooth16_Dentin.obj.100000.tree" );

```

Code 7.3: Human tooth augmentation

The density was initially chosen based on CT-scan data, in which tooth dentin has an average density of  $2\text{ g/cm}^3$ , and enamel of  $2.7\text{ g/cm}^3$  [Gradl et al., 2016]. Later on, I have tuned these values with an expert in dentistry iteratively to optimize the realism of the haptic feedback during material removal, and the speed of material removal.

#### 7.4 FAST CONTINUOUS REMOVAL OF MATERIAL

In this section I will describe the algorithms behind material removal. To implement material removal, one has to modify the sphere packing at runtime, while keeping the simulation consistent and stable (see Figure 7.13). In the following, I have gathered desirable properties of material removal, which essentially define its quality by objective measures.

1. If the force that the user exerts on the cutting contact doesn't change, the cutting speed should be constant (assuming an unchanging contact configuration).
2. However, cutting speed can not be arbitrarily high, no matter how hard the user pushes, as it is limited by the cutting force of the tool.
3. Cutting speed should take into consideration the local material properties, i.e. cutting should be faster for less dense material, and slower for denser material.
4. Cutting should be continuous, in order to not allow creating cavities inside, without cutting an appropriate opening first.

These properties are complemented by the following list of subjective markers of quality, related to the emergent haptic feeling as experienced by the user.

5. The user should feel whether he is cutting through denser or less dense material.
6. The user should be able to haptically tell, when cutting through a preexisting cavity<sup>7</sup>.
7. The user should still have the feeling of being haptically guided by the surface, to a certain extend.

These markers can only be verified in experiments with human subjects, as they describe high-level effects on the human perception during haptical cutting, which are difficult quantify in other ways.

To facilitate the objective quality markers, I have taken each of them into account while designing the material removal algorithm, and the overall system. When I started out developing the system, I had been using a ad-hoc penalty-based approach, which essentially violated each of the above objective markers. Integrating continuous collision detection solves issue Item 4 and enabled me to implement solutions to issues Item 1, Item 2 and Item 3. The material removal itself is a simple procedure, the tool spheres are extended by the desired drill radius  $r_d$  at this time-step, and all overlapping spheres are resolved by reducing the environment spheres. The computation of the drill radius  $r_d$  is a function of the user's applied force, to a certain extend. Additionally, I parameterize a reference and



Figure 7.13: Dynamic features. Naturally, dynamic features, which were created due to material removal can be contacted and haptically rendered stably. Here, the hip reamer removed too much material on the acetabulum, penetrating the bone. The secondary handle is then inserted from the backside of the pelvis. Green shows the haptic tool, black the graphic tool.

<sup>7</sup>In dentistry there is a haptical sensation called “drop” which is experienced when the enamel and dentin is cut and the bur drops down into the empty pulp chamber.

maximum drilling radii,  $r_d^{\text{ref}}$  and  $r_d^{\text{max}}$ , which determine drilling speed when touching, i.e. contact without any force applied, and the maximally achievable drilling speed due to applying force<sup>8</sup>. The upper limit helps to keep the cutting rate at a physically plausible level, even if a very large amount of force is applied by the user. The reference drilling radius does not directly translate to a minimal feedrate, as the contact volume of active spheres is used to further derive the actual drilling radius  $r_d$ . If the contact volume  $w_D \neq 0$ , then

$$r_d = r_d^{\text{ref}} \log_{10} \left[ \left( \frac{P_{\text{imp}}}{w_D} + 1 \right)^{P_{\text{pow}}} \right] \quad (7.5)$$

else, I default to  $r_d = r_d^{\text{ref}}$ . Further, I calculate a user force  $\vec{f}_u$  and torque  $\vec{\tau}_u$  that determines with what force the user is pushing on the object. The force is calculated based on the difference between the pose of Graphic Tool and Haptic Tool. Essentially, this is the force of the user interaction spring (see Equation (6.21)), however, evaluated from the Graphic Tool side. Additionally, I use a different set of spring constants to further allow the developer to fine-tune the desired drilling behavior<sup>9</sup> and to remove haptic rendering parameters' effect on it.

$$\begin{aligned} \vec{f}_u &= [\mathbf{p}(T_H) - \mathbf{p}(T_G)] k_t^u & - \vec{v}(T_G) c_t^u & - [\vec{v}(T_H) - \vec{v}(T_G)] b_t^u \\ \vec{\tau}_u &= \vec{\tau}_t k_r^u & - \vec{\omega}(T_G) c_r^u & - [\vec{\omega}(T_H) - \vec{\omega}(T_G)] b_r^u \end{aligned} \quad (7.6)$$

with

$$\mathbf{p}(T_H) = {}^W \mathbf{H}_{T_H} \mathbf{c}_t \quad \mathbf{p}(T_G) = {}^W \mathbf{H}_{T_G} \mathbf{c}_t$$

The algorithm (described in Algorithm 7.2) is programmatically similar to contact detection, as we again perform an overlap test. The main difference is that in this case, we are only interested in active parts of the BVH. Additional differences are, that we do not collect data, but potentially modify environment spheres, and of course a different radius extension. The radius extension is the exact amount of material, given as a distance, that is to be removed from the environment. This procedure is dependent on the surface estimation to be finished, as the minimal distance  $d_D$  between tool and environment needs to be known. This distance  $d_D$  is subtracted from the detected distance during overlap tests in material removal, to ensure, that we do, in fact, remove material equal to a linear distance of  $r_d$ . Otherwise, we would be removing material equal to  $r_d - d_D$ , even though the distance  $d_D$  is a rather arbitrary measurement, that could fluctuate considerably and thus, heavily influence the drilling behavior<sup>10</sup>.

Since during contact detection, the used radius extension, contact margin  $P_{\text{ContactMargin}}$ , is in general much larger than  $r_d$ <sup>11</sup>, we guarantee that the minimal distance is accurate for all spheres that could potentially remove material. In fact, the following requirement is necessary for the algorithm to function properly

$$P_{\text{ContactMargin}} > r_d \quad (7.7)$$

Interestingly, even if  $d_D$  were negative, i.e. the actual sphere packings overlap (without radius extension), this approach would still result in the requested removal, equal to a linear distance of  $r_d$ .

## 7.5 VARIATIONS ON THE MATERIAL REMOVAL ALGORITHM

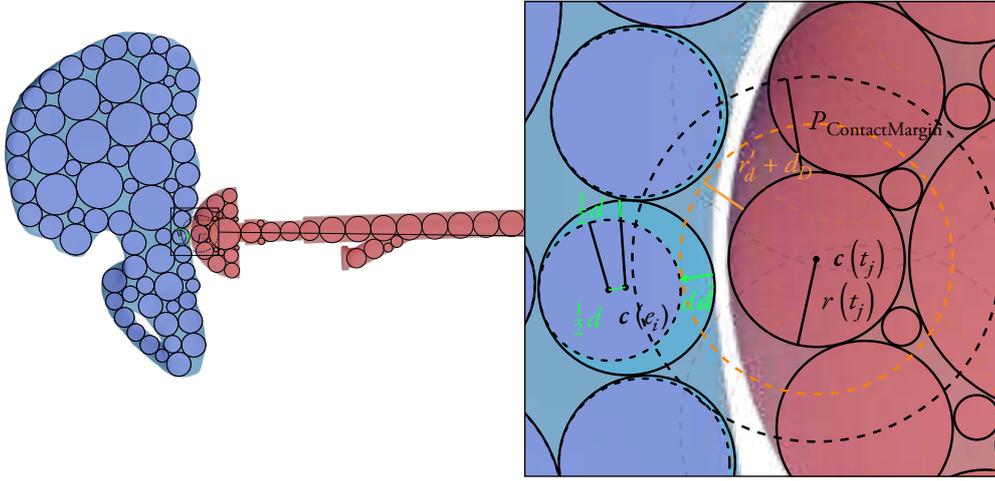
There are several different, earlier versions of the material removal, that I have developed. These versions had different challenges, disadvantages and advantages, of which I want to mention a few here.

<sup>8</sup>I have used  $r_d^{\text{ref}} \in [10 \mu\text{m}, 50 \mu\text{m}]$  and  $r_d^{\text{max}} \in [50 \mu\text{m}, 200 \mu\text{m}]$ , for simulation of acetabular reaming and femur sawing, respectively, during hip surgery. This limits the feedrate to roughly  $50 \frac{\text{mm}}{\text{s}}$  for reaming and  $200 \frac{\text{mm}}{\text{s}}$  for sawing.

<sup>9</sup>If this extra set of constants seem like too many extra parameters to tune, it is feasible to default them to the same defaults that are used for the user interaction spring. However, one might want avoid coupling drilling behaviour and haptic rendering parameters, as the latter might be adjusted for new haptic devices, while the drilling behavior ought not to change.

<sup>10</sup>In fact, before introducing the distance adjustment, I had considerable spikes in actual drilling performed, if I rotated a tool in contact during drilling, as  $d_D$  would momentarily be much lower. This resulted in the drilling behaviour depending more on the amount of rotation in contact, rather than the amount of static force applied to the tool, which was undesirable for me.

<sup>11</sup>I chose  $P_{\text{ContactMargin}}$  2-3 orders of magnitude larger than  $r_d$ , e.g. 2 mm and 10  $\mu\text{m}$ .



**Figure 7.14: Material removal sketch.** A simplified 2D sketch that visualizes the material removal algorithm, focusing on one sphere pair.  $P_{\text{ContactMargin}}$  and  $d_D$  are the only global values, all other values are local to the thread and vary depending on the current sphere pair. The end result of the current iteration is, that the environment sphere is directionally shrunk by  $\vec{d}$ , which is reflected in modification of its center and radius.

Red spheres belong to the virtual tool (here a acetabular reamer), blue spheres to the removable material, also called environment (here the pelvic bone). Spherepackings are simplified, in practice the density is much larger, and environment spheres are smaller.

---

### Algorithm 7.1 Material Removal Setup (CPU)

---

**Input:**  ${}^W\mathbf{H}_{T_G}, {}^W\mathbf{H}_E \in \mathbb{R}^{4 \times 4}$   $\vec{f}_u, \vec{z}_u \in \mathbb{R}^3$ , callback  $\lambda$   
**Result:** modified spherepacking  $E$

```

procedure MATERIALREMOVAL
  select simulation GPU
  // Transform data to  $T_G$  space
   ${}^{T_G}\mathbf{H}_E \leftarrow {}^W\mathbf{H}_{T_G}^{-1} {}^W\mathbf{H}_E$ 
   ${}^E\mathbf{H}_{T_G} \leftarrow {}^{T_G}\mathbf{H}_E^{-1}$ 
   ${}^{T_G}\mathbf{H}_W \leftarrow {}^W\mathbf{H}_{T_G}^{-1}$ 
   $\vec{f}_u \leftarrow {}^{T_G}\mathbf{H}_W \vec{f}_u$ 
   $\vec{z}_u \leftarrow {}^{T_G}\mathbf{H}_W \vec{z}_u$ 
   $d_D \leftarrow S({}^{T_G}\mathbf{H}_W)_{00} d_D$ 
   $r_d \leftarrow 1$ 
  if  $w_D \geq 0$  then
     $r_d \leftarrow r_d^{\text{ref}} \log_{10} \left[ \left( \frac{P_{\text{imp}}}{w_D} + 1 \right)^{P_{\text{pow}}} \right]$ 
    MATERIALREMOVALPARALLEL( ${}^{T_G}\mathbf{H}_E, {}^E\mathbf{H}_{T_G}, r_d, \vec{f}_u, \vec{z}_u, c(T_G)$ )
  // In the meantime, perform CPU task
  CALL( $\lambda$ )
  synchronize collision GPU stream
end procedure

```

▷ allow the caller to assign a parallel CPU task<sup>12</sup>  
 ▷ GPU- & CPU-task terminated, proceed

---

<sup>12</sup>I do contact resolution using this interface, as material removal does not depend on the contact resolution result. Depending on the use-case this might be more or less useful, e.g. if the contact resolution uses expensive constraint solving, this approach might save some time due to the overlapped computing.

#### 7.5.1 Cleaning up the BVH during Material Removal

When using the sample-based CCD (see Section 5.1.1), the algorithms were implemented as CPU programs, which necessitated the use of BVHs on both sides, the tool and the environment. Consequently, one has to keep the environment BVH up-to-date, even when leaves are being removed. Although, all algorithms still work if the BVH were to be not updated, if a radius check was implemented on the leaf-level, to not do computations with radii of zero and below. However, keeping the BVH up-to-date turns out to be essential for the performance, as due to removing leaves near the surface, we would otherwise have deep intersections of large parts of the BVH tree, which are the worst scenario for BVHs. The CCD with two BVH performs generally best if: (1) the BVH traversal can be stopped early due to no possible collision, or (2) a deep leaf intersection raises the minimal distance to such an extend, that other parts of the BVH can be disregarded, as no earlier collision is possible, leading again to stopping the BVH traversal early. For a boolean overlap test,

**Algorithm 7.2** Material Removal in Parallel (GPU)

**Input:**  $T_G \mathbf{H}_E, {}^E \mathbf{H}_{T_G} \in \mathbb{R}^{4 \times 4}$   $\vec{f}_u, \vec{z}_u, c(T_G) \in \mathbb{R}^3, r_d, d_D \in \mathbb{R}$   
**Result:** modified spherepacking  $E$

**procedure** MATERIALREMOVALPARALLEL

**for**  $i \leftarrow i_t + i_b d_b; i < |E|; i \leftarrow i + d_b d_g$  **do in parallel**

*// Read and prepare data of  $e_i$  from memory*

**coalesced read**  $e_i \in E$

**if**  $r(e_i) \leq 0$  **or**  $\rho(e_i) \leq 0$  **then**

**continue**

$c(e_i) \leftarrow T_G \mathbf{H}_E c(e_i)$

$r(e_i) \leftarrow \mathcal{S} \left( T_G \mathbf{H}_E \right)_{00} r(e_i)$

$r'_d \leftarrow \min \left( r_d + \left\| \vec{f}_u + \vec{z}_u \times [c(e_i) - c(T_G)] \right\|, r_d^{\max} \right)$

*// Calculate if and how  $e_i$  is modified due to material removal*

$b_{\text{modified}} \leftarrow 0$

**initialize** stack  $\leftarrow \emptyset$

    node  $n \leftarrow t_{\text{root}}$

**repeat**

**for all**  $t_j \in \text{children}(n)$  **do**

**if**  $t_j$  is not an active sphere **then**

**continue**

▷ only consider active subtrees

$\vec{d} \leftarrow c(e_i) - c(t_j)$

$d \leftarrow \left\| \vec{d} \right\| - r(e_i) - r(t_j) - r'_d - d_D$ <sup>13</sup>

**if**  $d \geq 0$  **then**

**continue**

▷ no possible overlap between  $e_i, t_j$

**if not leaf**( $t_j$ ) **then**

**push**  $t_j$  to stack

▷ inner nodes cause traversal

**else**

$\vec{d} \leftarrow \vec{d} / \left\| \vec{d} \right\|$

$r'_d \leftarrow r'_d \left[ \left( \vec{d} \vec{d}(t_j) + 1 \right) 0.5 \right]^{P_{\text{directionPower}}}$ <sup>14</sup>

$d \leftarrow \left\| \vec{d} \right\| - r(e_i) - r(t_j) - r'_d - d_D$

**if**  $d \geq 0$  **then**

**continue**

▷ no actual overlap between  $e_i, t_j$

$d \leftarrow -0.5 d$

▷ the radius is used twice in the following

$r(e_i) \leftarrow r(e_i) - d$

▷ use #1

$\vec{c}(e_i) \leftarrow \vec{c}(e_i) + d \vec{d}$

▷ use #2

$b_{\text{modified}} \leftarrow 1$

**until**  $n = \emptyset$

*// Transform and store data of  $e_i$  in memory*

**if**  $b_{\text{modified}}$  **then**

▷ only write back data if it was actually modified<sup>15</sup>

$c(e_i) \leftarrow {}^E \mathbf{H}_{T_G} c(e_i)$

$r(e_i) \leftarrow \mathcal{S} \left( T_G \mathbf{H}_E \right)_{00} r(e_i)$

        set dirty flag of  $e_i$  to 1

**end procedure**

<sup>13</sup>The distance between active spheres and the environment is subtracted to consider all contacting spheres as touching (i.e. their distance is 0). This provides consistent and predictable drilling behaviour, otherwise the rather arbitrary distance between spheres would influence the material removal that occurs between those contacting spheres:

- low distance: too much removal
- high distance: too little or no removal

<sup>14</sup> $P_{\text{directionPower}} \in [1, 10] \subset \mathbb{R}$  controls the strictness of the influence of the removal direction, where larger values mean less influence of active spheres with drilling directions that point away from the shortest connecting line between the contact spheres.

<sup>15</sup>This is done to increase numerical robustness, as matrix inversion is not numerically robust. For example, a program that uses 32-bit floats for a 3D point  $p$ , the following equality is generally only true within a finite accuracy

$$M^{-1} M p = p$$

the cleanliness of the BVH is even more crucial for performance, since a clear miss and any overlap will lead to immediate termination of the algorithm. To clean up a BVH, I have created a recursive method that updates the BVH when a node is to be removed (see [Algorithm 7.3](#)), typically only leaves are removed – e.g. due to material removal. However, if all leaves of a subtree are removed, the whole subtree can be removed, as only the leaves hold actual geometrical data of the body.

**Algorithm 7.3** Sphere Removal – BVH Update**Input:** SphereTree  $S$ , SphereNode  $s$ **Result:**  $S$  with  $s$  removed and BVH updated

```

procedure REMOVEUPWARD
   $p \leftarrow$  parent of  $s$ 
  if  $p = s$ 16 or  $p$  is root of  $S$  then
    return
  for all children  $c_i$  of  $p$  do
    if  $c_i = s$  then
      delete  $c_i$  at  $p$ 
    if  $p$  has no children then
      REMOVEUPWARD( $S, p$ )
end procedure

```

<sup>16</sup>  $p = s$  means  $s$  has no sensible parent node.

▷ remove this node from the parent's children

▷ propagate removal upwards the BVH

## 7.5.2 Leveraging Purely Translational Motion for Continuous Removal

When I had been working with purely translational CCD, there was potential to improve the material removal to be continuous. As the translational motion of spheres during one time-step can be represented by a set of translationally swept spheres, i.e. a set of capsules, the material removal can be adjusted to overlap capsules and spheres (see Figure 7.15). This would also remove the necessity to increase the radius during removal, and thus, would allow for more precise cutting. The algorithm (see Algorithm 7.4) is a variation of Algorithm 7.2, and assumes the current motion of a body from pose  $B$  to another pose  $B'$ , denoted as  $\Delta H_B$ , is purely translational. If it is purely translational, we can extract the translation vector  $\Delta p$ <sup>17</sup> by

<sup>17</sup> Whether the frames  $B$  and  $B'$  are given with the origin in the center of mass of the body or not is irrelevant for this equation.

$$\Delta p = T \left( {}^{B'} \Delta H_B \right)_{i3}, i \in [0, 2] \subset \mathbb{N} \quad (7.8)$$

This vector can then be used to construct a line starting at the center of  $e_i$ ,  $c(e_i)$ , and ending at  $c(e_i) - \Delta p$ . A capsule that is constructed around this line, with radius  $r(e_i)$ , can then be tested against  $c(t_j)$  by testing the line against the sphere  $t_j$ , with center  $c(t_j)$  and radius  $r(e_i) + r(t_j)$ , to check whether  $t_j$  and the swept sphere of  $e_i$  collide. The collision check first calculates the point on the line, called  $g$ , that is closest to  $t_j$ , which has three possible cases, which need to be differentiated:

1. The closest point is at the start of the line  $g = c(e_i)$
2. The closest point is at the end of the line  $g = c(e_i) - \Delta p$
3. The closest point is between the start and end point: project  $c(t_j)$

$$g = c(e_i) - \frac{\Delta p}{\|\Delta p\|} \left[ \left( c(t_j) - c(e_i) \right) \cdot \left( -\frac{\Delta p}{\|\Delta p\|} \right) \right] \quad (7.9)$$

Then, the distance between  $g$  and  $c(t_j)$  needs to be smaller than  $r(e_i) + r(t_j)$ , in which case sphere and capsule collide. The overlap is resolved by directional shrinking of  $e_i$ , as was done in Algorithm 7.2.

## 7.6 THE VISUALIZATION PIPELINE

In this section, I want to discuss the visualization of sphere packings. In my scenarios so far, there is only one virtual tool and one or more static environment objects; both sides will be

**Algorithm 7.4** Capsule Material Removal in Parallel (GPU)

**Input:**  ${}^T_G\mathbf{H}_E, {}^E\mathbf{H}_{T_G} \in \mathbb{R}^{4 \times 4}$   $\Delta p \in \mathbb{R}^3$ ,  $r_d, d_D \in \mathbb{R}$   
**Result:** modified spherepacking  $E$

```

procedure CONTINUOUSMATERIALREMOVALPARALLEL
  for  $i \leftarrow i_t + i_b d_b$ ;  $i < |E|$ ;  $i \leftarrow i + d_b d_g$  do in parallel
    // Read and prepare data of  $e_i$  from memory
    coalesced read  $e_i \in E$ 
    if  $r(e_i) \leq 0$  or  $\rho(e_i) \leq 0$  then
      | continue
       $r'_d \leftarrow \min(r_d + \|\vec{f}_u + \vec{z}_u \times [c(e_i) - c(T_G)]\|, r_d^{\max})$ 
       $c(e_i) \leftarrow {}^T_G\mathbf{H}_E c(e_i)$ 
       $r(e_i) \leftarrow \mathcal{S}({}^T_G\mathbf{H}_E)_{00} r(e_i)$ 

    // Calculate if and how  $e_i$  is modified due to material removal
     $b_{\text{modified}} \leftarrow 0$ 
    initialize stack  $\leftarrow \emptyset$ 
    node  $n \leftarrow t_{\text{root}}$ 
    repeat
      for all  $t_j \in \text{children}(n)$  do
        if  $t_j$  is not an active sphere then
          | continue ▷ only consider active subtrees
          if  $-\Delta p \cdot [c(t_j) - c(e_i)] \leq 0$  then ▷ capsule case 1
            |  $g \leftarrow c(e_i)$ 
            else if  $-\Delta p \cdot [c(t_j) - (c(e_i) - \Delta p)] \geq 0$  then ▷ capsule case 2
              |  $g \leftarrow c(e_i) - \Delta p$ 
            else
              |  $g \leftarrow c(e_i) - \frac{\Delta p}{\|\Delta p\|} \left[ (c(t_j) - c(e_i)) \cdot \left( -\frac{\Delta p}{\|\Delta p\|} \right) \right]$  ▷ capsule case 3
               $\vec{d} \leftarrow c(t_j) - g$ 
               $d \leftarrow \|\vec{d}\| - r(e_i) - r(t_j)$  ▷ no radius extension
              if  $d \geq 0$  then
                | continue ▷ no overlap between  $e_i, t_j$ 
                if not leaf  $(t_j)$  then
                  | push  $t_j$  to stack ▷ inner nodes cause traversal
                else
                  |  $\vec{d} \leftarrow \vec{d} / \|\vec{d}\|$ 
                  |  $d \leftarrow -0.5 d$  ▷ the radius is used twice in the following
                  |  $r(e_i) \leftarrow r(e_i) - d$  ▷ use #1
                  |  $\vec{c}(e_i) \leftarrow \vec{c}(e_i) - d \vec{d}$  ▷ use #2
                  |  $b_{\text{modified}} \leftarrow 1$ 
              until  $n = \emptyset$ 

    // Transform and store data of  $e_i$  in memory
    if  $b_{\text{modified}}$  then ▷ only write back data if it was actually modified18
      |  $c(e_i) \leftarrow {}^E\mathbf{H}_{T_G} c(e_i)$ 
      |  $r(e_i) \leftarrow \mathcal{S}({}^T_G\mathbf{H}_E)_{00} r(e_i)$ 
      | set dirty flag of  $e_i$  to 1
  end procedure
    
```

<sup>18</sup>This is done to increase numerical robustness, as matrix inversion is not numerically robust. For example, a program that uses 32-bit floats for a 3D point  $p$ , the following equality is generally only true within a finite accuracy

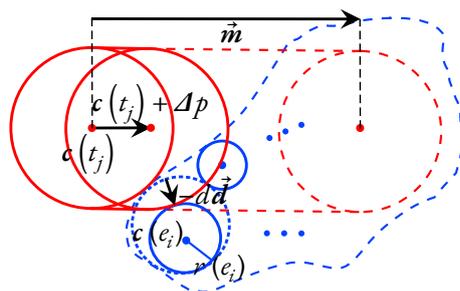
$$M^{-1} M p = p$$

discussed in this section. The sphere packing of virtual tools are simple to visualize, since the tool is rigid, the sphere packing is static, and it can only ever be linearly transformed, there is no need to consider visualizing the sphere packing. Instead, we can simply render the original mesh in place of the sphere packing. The virtual haptic tool is coupled to the haptic device (for more details on the generation of the transform [Chapter 6](#)), and is considered an auxiliary object to enable virtual coupling between virtual tool and real device. Therefore, it is simply *not* shown in production — however, rendering it during development might be helpful. The virtual graphical tool on the other hand is simulated and indirectly follows the haptic device, if possible without collision and thereby allows physically-simulated interaction with the virtual environment (more details on the simulation in [Chapter 4](#)). The virtual tool is rendered at the pose of the rigid body, as the simulation state currently dictates. In general, when activating the cutting instrument (e.g. saw blade, reamer head or dental bur tip), it is rotating, or oscillating in a very small range. This might be no coincidence, as designing tools such that the cutting tools have the same rough shape when turned off and on makes handling them more predictable. Fortunately, this also means we do not need to actually do any rotation of a subset of the spheres to simulate rotation, instead such local rotation can simply be ignored on the sphere level. However, the subset of the mesh that represents the cutting instrument is locally rotated when cutting is enabled. This further convinces the user that a realistic cutting process is ongoing when initiated<sup>19</sup>, although the sphere packing of the tool not actually modified and remains completely static, except for global linear transformations.

In contrast, environment spheres are difficult to visualize, for the one reason that their sphere packings are prone to change due to material removal. This immediately removes the option to render a static mesh, as is done for tools. The ad-hoc solution would be to simply render the spheres, which could be done in a fragment shader at considerable speed. However, the visual quality of such an object is not convincing and would considerably lower immersion and the overall quality of a surgery simulator designed to look and feel realistic. Instead, I decided to generate an implicit surface from the dynamic sphere packing, which I discretized by a SDF, and triangulate by marching cubes (MC) at run-time to produce a dynamic mesh. Naively implementing this procedure will mean all of the work would have to be repeated every single frame. This would create a very slow program, in  $O(nm)$ , where  $n$  is the grid count and  $m$  the sphere count. Additionally, both numbers are usually large, e.g.  $n = 128^3 \approx 2\,000\,000$  and  $m \approx 100\,000$ . Consequently, considerable spatio-temporal optimization was necessary to make the procedure run at interactive rates. These optimizations make the algorithm considerably complicated, and the presented solution is algorithmically simplified to be more understandable, although some optimizations might be omitted.

The basic idea how this dynamic mesh is rendered, is to keep a 3D grid of triangles, normals and vertex colors inside the game-engine. This is well represented by UE's ProceduralMeshComponent (PMC), as they can be segmented into *sections*, which each can hold a face array that indexes a local vertex, normal and colors array. An asynchronous call to `ImplicitObject` is periodically placed to generate new triangles (see [Algorithm 7.5](#)). When

<sup>19</sup>It would be interesting to further improve immersion by displaying a particle effect when cutting is ongoing at the current center of contacts.



**Figure 7.15: Capsule material removal sketch.** A visualization of the material removal variant where the tool's motion  $\vec{m}$  is assumed to be purely translational. Such motion stops at first contact, afterwards each sphere is nudged towards the intersection slightly by  $\Delta p$ . Any overlap between environment and tool is resolved by directional shrinking of the environment sphere.

**Algorithm 7.5** Dynamic Mesh Update**Input:****Result:** updated dynamic mesh

```

procedure GAMEMESH TICK
  if MESHUPDATE() then ▷ return type indicates task completion
     $D, I, V, N, C^{22} \leftarrow \text{RETRIEVEDIRTYCELLS}()$ 
    for all  $d \in D \subset \mathbb{N}$  do ▷ iterate over dirty cells
      CREATEMESHSECTION( $d, I[d], V[d], N[d], C[d]$ )
      if first update then
        for all  $d \in D \subset \mathbb{N}$  do
          initialize shader of mesh section at index  $d^{23}$ 
  end procedure

```

<sup>22</sup>  $I[d], V[d], N[d], C[d]$  are 2D arrays that hold indices, vertices, normals and colors for each coarse grid cell with index  $d$ .

<sup>23</sup> All possible sections are present in the first update, and even a reset of the object to its initial state would not violate this approach.

<sup>20</sup> This performance difference was observable when using PMC. However, I have also used a different UE rendering plugin called RuntimeMeshComponent, which had similar performance for both implementations. Unfortunately, this plugin is highly unstable and would crash regularly, possibly only when using two GPUs.

<sup>21</sup> These vertex pools are local to each section. One might think having a global vertex pool would result in superior performance, however, this (1) would considerably complicate updating vertices, and (2) sections already group proximate vertices, so the majority of duplicates will be within the same section anyway, and the only other possible duplicates could occur between neighbouring sections.

<sup>24</sup> Writing this thread handling routine is necessary, since handling the extra thread from the game-engine proved not reliable, as the game-engine exiting would not be aligned with the thread join. The consequence was occasionally the asynchronous thread would still be working after the game-engine exited the simulation, leading to access of released memory and other fatal errors.

<sup>29</sup> Actually, we only need to consider active spheres (and only when drilling is on-going), thereby, we are shrinking down the movement AABB considerably.

<sup>30</sup> During this step I also track whether SDF values *actually* changed to refine the dirty flags.

the new triangles generation completes, the periodical call will return true, which indicates the new data can be retrieved. A faster synchronous call can then be placed to `ImplicitObject`, which returns the actual triangle data in a compact form. The data is organized into a coarser grid (each coarse cell encompasses e.g. 8 regular cells), where each of the coarse cells corresponds to a PMC section. Introducing these coarse cells provides a considerable (about 10x) speed-up for the rendering speed, as we only need to update vertices within dirty cells, instead of the complete mesh<sup>20</sup>. There is also a dirty grid to indicate which sections were modified during the last call, indicating that those sections should be updated with the new data inside the game-engine. Thereby, the rendered mesh stays up-to-date, while avoiding unnecessary work. To avoid duplicate vertices, each section has a vertex pool, which is indexed by the face array<sup>21</sup>.

From the game-engine side, an actor would periodically (e.g. each tick) check via a synchronous call whether new mesh data is available, meaning the asynchronous task completed. This interface routine is described in [Algorithm 7.6](#), and handles multiple small tasks at once. Firstly, the return type indicates task completion. Additionally, the routine also handles creating new threads, if the last one completed<sup>24</sup>. Thread safety is ensured by locking the thread creation using a shared semaphore with its destruction in the `ImplicitObject` deconstructor. Therefore, we halt an intermediate simulation exit until the thread terminates its work, without possibility to create another new thread intermediately. If there is new data available, the routine returns true and resets the new data flag. This return value indicates to the GE script that new data is to be read, though it needs to be retrieved via another function. A subsequent function call is also mandatory to be processed, as the data is incremental. Only data of coarse cells which actually have data modified (e.g. due to localized material removal) since the last retrieval are transferred. The incremental data is represented as a list of indices of dirty cells, and the corresponding cell data. The dirty cell indices are each used to access cells and update or create a new section for that index on the GE side (see [Algorithm 7.5](#)).

The thread that is created in [Algorithm 7.6](#) is given the task to call the function `MESHUPDATEASYNC()` (see [Algorithm 7.7](#)). This routine is called asynchronously, because its typical run-time is too slow to halt the game-loop for its long duration. It basically takes the swept AABB of the tool<sup>29</sup> since the last call (this is a spatio-temporal optimization measure), and regenerates the triangles of those cells. To generate triangles from a sphere packing, the following steps are necessary:

1. Generate SDF from spheres (see [METASDF\(\)](#) in [Algorithm 7.8](#))
  - a) Generate raw SDF based on current sphere packing state (see [UPDATESDF\(\)](#) in [Algorithm 7.9](#))
  - b) Bilateral smoothing of raw SDF<sup>30</sup>, including distances, normals and colors (see [FIELDSMOOTHING\(\)](#) in [Algorithm 7.10](#))

**Algorithm 7.6** Mesh Update Threading Handler**Input:** tool movement AABB  $b_{\min}, b_{\max}$ **Result:** {false, true}

```

procedure MESHUPDATE

```

```

   $b \leftarrow b_{\text{data}}$ 25

```

```

  if  $b_{\text{working}}$  then

```

```

    return false

```

▷ task already in progress

```

  if not  $b \wedge \text{not } b_{\text{working}}$ 26 then

```

▷ no new data, no task in progress

```

    lock semaphore decon

```

▷ decon is locked in deconstructor<sup>27</sup>

```

    if  $b_{\text{destroy}}$  then

```

```

      return false

```

```

       $T \leftarrow \text{new thread}$  doing  $\text{MESHUPDATEASYNC}(b_{\min}, b_{\max})$ 28

```

```

      return false

```

```

  if  $b$  then

```

```

     $b_{\text{data}} \leftarrow \text{false}$ 

```

▷ the new data is now being evicted

```

    return true

```

▷ only case in which new data is available

```

  return false

```

```

end procedure

```

<sup>25</sup>  $b_{\text{data}}$  is a member variable of the implicit object instance that is initially false.

<sup>26</sup> Similarly, a member variable, initially false

<sup>27</sup> The semaphore lock is necessary to catch a simulation deconstruction before the thread completed. The semaphore is automatically unlocked once the context is switched, such as exiting the function call.

<sup>28</sup> During the deconstruction,  $T$  needs to be joined while the semaphore decon is locked to be thread safe.



**Figure 7.16: Dynamic mesh pipeline.** Visualization of the intermediate states of the dynamic material.

**Left** Sphere packing.

**Left-Center** SDF data as voxels.

**Center** Triangulation of raw SDF.

**Center-Right** Triangulation of smoothed SDF.

**Right** Triangulation of smoothed SDF with smoothed normals.

## 2. Generate triangles from SDF

- a) Triangulate smoothed SDF using Marching Cubes<sup>31</sup>, and trilinear interpolation of normals and colors (see `PARALLELMARCHINGCUBES()` in Algorithm 7.11)
- b) Reduce the vertex count by pooling duplicate vertices (see `VERTEXPOOLING()` in Algorithm 7.12)

The intermediate results are visualized in Figure 7.16. The overall routine is described in `MESHUPDATEASYNC()`. Which in turn calls several subsequent routines for each task, which I will explain in more detail in the following sections.

```

cudaMemcpyAsync( triGridDirty, d_triGridDirty, triGridRes*sizeof(bool)
, D2H, stream );
cudaStreamSynchronize( stream ); // Wait for dirty flags to download
// Start an asynchronous download to run concurrently
cudaMemcpyAsync( triGridVertexCounts, d_triGridCounts, triGridRes*
sizeof(int), D2H, stream );
dirtyCellIds.clear();
for (int i=0; i<triGridRes1d; ++i) {
  auto dirty = triGridDirty[i];
  if (!dirty) continue;
  cudaMemcpyAsync( triGrid [i], d_triGrid [i], triGridCellSize*
sizeof(Vec3<T>), D2H, stream );
  cudaMemcpyAsync( normGrid[i], d_normGrid[i], triGridCellSize*
sizeof(Vec3<T>), D2H, stream );

```

<sup>31</sup>The algorithm runs on the GPU, therefore the data needs to be transferred afterwards from GPU to CPU (see Code 7.4). Ideally, one would avoid this transfer, as the GE later has to reupload them to the GPU to render them. However, this level of rendering control is difficult to achieve when using comprehensive and complex GEs, such as Unreal Engine. One would have to debug, study and modify the complex renderer in great detail to achieve this kind of integration. Additionally, this would negatively affect modularity and compatibility of the simulation library, e.g. with other GEs.

```

    cudaMemcpyAsync( colGrid [i], d_colGrid [i], triGridCellSize*
                    sizeof(Vec3<T>), D2H, stream );
    dirtyCellIds.push_back( i );
}
cudaStreamSynchronize( stream );

```

Code 7.4: Compacted mesh transfer GPU to CPU

**Algorithm 7.7** Mesh Update Meta Routine**Input:** tool movement AABB  $\mathbf{b}_{\min}$ ,  $\mathbf{b}_{\max}$ **Result:****procedure** MESHUPDATEASYNC $\mathbf{b}_{\text{working}} \leftarrow \text{true}$ 

stall if update frequency too high

 $\mathbf{b}_{\text{reset}} \leftarrow \text{check if there is reason to reset}$ <sup>32</sup> $\mathbf{b}_{\text{resetted}} \leftarrow \mathbf{b}_{\text{reset}}$ *// SDF generation* $\mathcal{g}_{\min}, \mathcal{g}_{\max} \leftarrow \text{METASDF}(\mathbf{b}_{\min}, \mathbf{b}_{\max})$ **if**  $\mathbf{b}_{\text{resetted}}$  **then**  $\mathbf{b}_{\text{reset}} \leftarrow \text{false}$ *// Coarse field bounds* $\mathcal{g}_{\min}^t, \mathcal{g}_{\max}^t \in \mathbb{N}^3 \leftarrow \mathcal{g}_{\min}, \mathcal{g}_{\max}$ **if not** first update **then** $\mathcal{g}_{\min}^t \leftarrow \mathcal{g}_{\min} / \mathcal{g}_d^t$ <sup>33</sup> $\mathcal{g}_{\min}^t \leftarrow \mathcal{g}_{\min}^t \cdot \mathcal{g}_d^t$ <sup>34</sup> $\mathcal{g}_{\max}^t \leftarrow \mathcal{g}_{\max} / \mathcal{g}_d^t + (1, 1, 1)^T$  $\mathcal{g}_{\max}^t \leftarrow \mathcal{g}_{\max}^t \cdot \mathcal{g}_d^t$ clamp  $\mathcal{g}_{\min}^t$  and  $\mathcal{g}_{\max}^t$  elementwise to  $[(0, 0, 0)^T, \vec{\mathcal{g}}_r - (2, 2, 2)^T]$ *// Marching cubes***select** rendering GPU**set** all dirty flags  $d \in D$  to **false****kernel call** PARALLELMARCHINGCUBES()

DOWNLOADTRIANGLES()

▷ download modified data (Code 7.4)

VERTEXPOOLING()

 $\mathbf{b}_{\text{data}} \leftarrow \text{true}$ 

▷ indicate new data is available to retrieve

 $\mathbf{b}_{\text{working}} \leftarrow \text{false}$ 

▷ thread concludes, new thread may be created

**end procedure**<sup>32</sup> e.g. reset key hit or parameters like iso-value modified.<sup>33</sup>  $\mathcal{g}_d^t$  is the coarse grid cell size, where the unit is fine grid cell count, e.g.  $\mathcal{g}_d^t = 8$  means a coarse grid cell encompasses 8 fine grid cells.<sup>34</sup> This multiplication is not inverse of the previous division, as the division was in integer-math, meaning it also floored the result.

## 7.6.1 Generating A Signed Distance Field From Spheres

Firstly, `METASDF()` is called to generate the SDF, and other fields, such as normal- and color field. The SDF is defined for a voxel  $c_i$  and a sphere packing  $s_j \in S$  by the following function:

$$\max_{s_j \in S} \left[ - \left( \|\mathbf{p}(c_i) - \mathbf{p}(s_j)\| - r(s_j) - \frac{\|\vec{\mathcal{g}}_d\|}{4} \right) \right] \quad (7.10)$$

Since we only sample each voxel at its center point, we need to determine a good approximation of the distance, such that the surface if sampled at  $\alpha = 0$  has the following properties:

1. Minimizes holes in the interior, since the sphere packing has nearly countless holes of various sizes.

2. The outer contour aligns well with the spheres shape, as packing is close to the ground truth.

The first property can be remedied by enlarging the implicit volume towards the “outside”, as this fills the holes.

However, this is counter-productive towards fulfilling the second property, as enlargement will also increase the size of the outer contour (see Figure 7.17 for the end result that has both properties). Fortunately, the collision detection is based on enlarging the virtual tool (see contact margin in Algorithm 5.6), which means collisions are detected slightly earlier than they visually occur. In case of a similarly small enlargement of the environment, both inconsistencies are mutually mitigated. Therefore, the choice of the enlargement needs to be carefully chosen<sup>35</sup>. For example, if a sphere is mostly inside a cube, but just missed the voxel center pointer, I still want to consider the voxel inside of the object. I achieve this by reducing the distance (before negation) further by  $\frac{\|\vec{g}_d\|}{4}$ , which is 25 % of the distance of a cell diagonal. Therefore, a sphere is considered inside, as long as it penetrates a voxel at least half way from corner towards the center. This already results in a well sampled SDF, when triangulated with at iso-value  $\alpha = 0$ . I have experimented with other possible offsets (see also Figure 7.19), such as:

- 0: the sphere needs to touch the center of the voxel to be considered inside.
- $\frac{\|\vec{g}_d\|}{2\sqrt{3}}$ : the sphere touching a cardinal voxel wall in the center is considered inside.
- $\frac{\|\vec{g}_d\|}{2}$ : the sphere touching any voxel corner is considered inside.

My experiments showed that the resulting mesh when choosing the midway value between the two extremes 0 and  $\frac{\|\vec{g}_d\|}{2}$

$$0 + \frac{\|\vec{g}_d\|}{2} = \frac{\|\vec{g}_d\|}{4} \quad (7.11)$$

provides a good compromise between eliminating holes and staying close to the groundtruth contour (see Figure 7.19). These values might still need to be tweaked for differently sized sphere packings. In the worst case, the resulting mesh is larger than the sphere packing in all directions by one voxel. The maximally negated distance is, of course, found close to the voxel  $c_i$ , therefore we can simplify the algorithm to search the bounding box of  $c_i$ , based on how the bounding box maps to grid coordinates. This function is then calculated for each voxel inside the bounding box.

I have implemented the SDF generation in a GPU program, therefore data transfers between CPU and GPU are necessary. Additionally, during SDF generation I will also refine the dirty flags and possibly the movement AABB to reduce the amount of data that needs to be processed by further steps. Therefore, I have written a meta algorithm (see Algorithm 7.8) that describes how the raw SDF is called and smoothed, including completion of the previously described extra tasks.

In the past I have also done the complete algorithm on the GPU, which would instead necessitate a data transfer between two GPUs, which can be performed directly with great speed. However, this kind of transfer only works if both GPUs are the same model and are installed using NVLink-SLI, which was not available in my hardware configuration. Consequently, I had to download the data from one GPU to CPU, and upload them to the next GPU, which was quite slow. Additionally, the increased demand for the second GPU would noticeably decrease VR rendering performance, which was the main reason for me to refrain from exclusively computing the results on the GPU.

The raw SDF generation iterates over each sphere and computes the bounding box, which is then translated to grid coordinates. For those cells, that are within both the sphere



Figure 7.17: Visualization of sphere packing. The shape defined by the spheres is faithfully replicated, i.e. there is no visual enlargement. However, the surface is without irregularities, such as holes or high-frequency bumps.

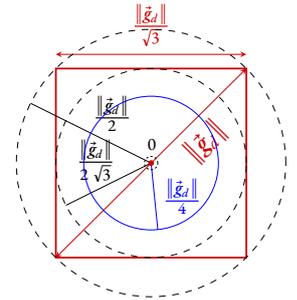


Figure 7.18: SDF offsets. Visualization of possible ways to classify a sphere inside of a voxel. Red denotes the voxel, its center and dimensions. The three black dashed spheres are the ad-hoc solutions, blue my solution.

<sup>35</sup>One could even imagine calculating the implicit volume enlargement based on the contact margin. However, this would not guarantee sufficient hole filling, and one also has to consider that smoothing (see Section 7.6.1.1) and default distance values influence the contour dimensions.

**Algorithm 7.8** SDF Generation

**Input:** Spherpacking  $\mathcal{S}$   
**Result:** 3D SDF describing implicit surface

```

procedure METASDF
  if first update then
    initialize raw layer tag field as Undefined
    initialize raw fields with default values36
  select render GPU
  download environment sphere packing37

  // Distance field bounds
   $g_{\min} \leftarrow (0, 0, 0)^T$ 
   $g_{\max} \leftarrow \vec{g}_r - (1, 1, 1)^T$ 
  if not first update and not  $b_{\text{reset}}$  then
     $g_{\min} \leftarrow \text{FLOORELEMENTWISE}((b_{\min} - g_o) \div \vec{g}_d)$ 
     $g_{\max} \leftarrow \text{CEILELEMENTWISE}((b_{\max} - g_o) \div \vec{g}_d)$ 
    clamp  $g_{\min}$  and  $g_{\max}$  elementwise to  $(0, 0, 0)^T, \vec{g}_r - (1, 1, 1)^T$ 
   $g'_{\min} \leftarrow \vec{g}_r$ 
   $g'_{\max} \leftarrow (0, 0, 0)^T$ 

  // Signed distance field
   $g'_{\min}, g'_{\max} \leftarrow \text{UPDATERAWSDF}(b_{\min}, b_{\max}, g_{\min}, g_{\max})$ 
   $g'_{\min} \leftarrow \text{MAXELEMENTWISE}(g'_{\min}, (0, 0, 0)^T)$ 
   $g'_{\max} \leftarrow \text{MINELEMENTWISE}(g'_{\max}, \vec{g}_r - (1, 1, 1)^T)$ 
  wait for GPU stream

  // SDF smoothing
  if use smoothing then
    upload fields (distances & normals) to GPU (padded38)
    if first update then ▷ transferred once, as voxel colors are static
      upload color field to GPU (padded)
       $\text{FIELDSMOOTHING}(b_{\min}, b_{\max}, g'_{\min}, g'_{\max}, N)$ 
      download color field from GPU
    wait for GPU stream
  first update  $\leftarrow$  false

  // Adjust movement AABB based on smoothing radius N
  if not  $b_{\text{reset}}$  then
     $g'_{\min} \leftarrow \text{MAXELEMENTWISE}((\frac{N}{2}, \frac{N}{2}, \frac{N}{2})^T, g'_{\min} - (\frac{N}{2}, \frac{N}{2}, \frac{N}{2})^T)$ 
     $g'_{\max} \leftarrow \text{MINELEMENTWISE}(\vec{g}_r - (\frac{N}{2}, \frac{N}{2}, \frac{N}{2})^T, g'_{\max} + (\frac{N}{2}, \frac{N}{2}, \frac{N}{2})^T)$ 
end procedure

```

<sup>36</sup>Distance is  $-1$  or user configured, normal is  $(0, 0, 0)^T$  and color is  $(-1, -1, -1)^T$ .

<sup>37</sup>I use pinned memory to speed up these transfers. Further optimization could be achieved by using dirty flags and stream compaction.

<sup>38</sup>The smoothed SDF is padded on the GPU to reduce bound checks in the kernel. Those cells keep the default values, as they are not written to during the data upload.



Figure 7.19: Effect of SDF offset.

**Left** Without enlargement, countless holes inside the volume.

**Right** Enlargement by 25% of the voxel diagonal – inner holes are mostly eliminated and the contour is smoothed.

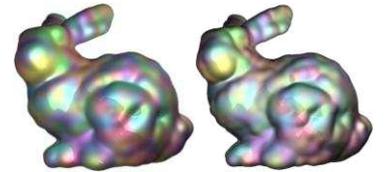
AABB and the tool movement AABB, I evaluate the SDF value, based on the aforementioned function. The SDF values of such cells are then compared against the SDF values of the previous generation call. If they differ, those cells are declared *dirty*, such that in later steps we can reference whether it needs to be further processed, i.e. regenerate triangles and update the triangles in the renderer. Layer tags and colors are only updated during the very first SDF generation, as they are static and will never change. These cell properties of the SDF, such as color, are not to be confused with the colors that are visibly rendered for a particular mesh. The colors of the mesh can, of course, change, but only because material is taken away and consequently colors that were previously on the inside are becoming visible due to them now being on the outside, which is visualized. Due to material removal, only the signed distance and normal can change in 3D space, since those are describing the object's geometry, which can be modified.

**Algorithm 7.9** Raw SDF Generation**Input:** Spherepacking  $S$ **Result:** 3D SDF describing implicit surface

```

procedure UPDATERAWSDF
  if first update then
    | initialize whole SDF with default distance value  $d_{\text{def}}^{39}$ 
  else
    | for all grid cells  $c_i$  within movement AABB do
    | | initialize  $d(c_i)$  with default distance value  $d_{\text{def}}$ 
  for sphere  $s_i \in S$  do in parallel
    | if  $r(s_i) \leq 0 \vee \rho(s_i) \leq 0$  then
    | | continue
    | // Calculate AABB of sphere  $s_i$ 
    |  $\mathbf{b}_{\min}^i \leftarrow \mathbf{p}(s_i) - (1, 1, 1)^T 2r(s_i)$ 
    |  $\mathbf{b}_{\max}^i \leftarrow \mathbf{p}(s_i) + (1, 1, 1)^T 2r(s_i)$ 
    |  $\mathbf{g}_{\min} \leftarrow (\mathbf{b}_{\min}^i - \mathbf{b}_{\min}) / \vec{\mathbf{g}}_d$   $\triangleright$  lower bound of cells in AABB
    |  $\mathbf{g}_{\max} \leftarrow (\mathbf{b}_{\max}^i - \mathbf{b}_{\min}) / \vec{\mathbf{g}}_d$   $\triangleright$  upper bound of cells in AABB
    | if not first update and not  $b_{\text{reset}}$  then
    | | clamp  $\mathbf{g}_{\min}$  and  $\mathbf{g}_{\max}$  to tool movement AABB ( $\mathbf{b}_{\min}, \mathbf{b}_{\max}$ )
    | clamp  $\mathbf{g}_{\min}$  and  $\mathbf{g}_{\max}$  to  $[(0, 0, 0)^T, \vec{\mathbf{g}}_r - 1]$ 
    | for all cells  $c_j \in [\mathbf{g}_{\min}, \mathbf{g}_{\max}]$  do
    | | compute 3D index  $(x, y, z)^T$  from 1D index  $j$ 
    | |  $\mathbf{p}(c_j) \leftarrow \mathbf{g}_o + (x, y, z)^T \mathbf{g}_d$   $\triangleright$  position of cell
    | |  $\vec{\mathbf{d}} \leftarrow \mathbf{p}(c_j) - \mathbf{p}(s_i)$   $\triangleright$  vector from sphere center to voxel center
    | |  $d \leftarrow \|\vec{\mathbf{d}}\| - r(s_i)$   $\triangleright$  distance of voxel center and sphere surface
    | |  $p \leftarrow -\left(d - \frac{\|\vec{\mathbf{g}}_d\|}{\sqrt{3}}\right)^{40}$   $\triangleright$  iso-value
    | | enter atomic section
    | | if  $p > d(c_j)$  then
    | | |  $d(c_j) \leftarrow p$   $\triangleright$  update SDF cell  $c_j$ 
    | | |  $\vec{\mathbf{n}}(c_j) \leftarrow \vec{\mathbf{d}} / \|\vec{\mathbf{d}}\|$   $\triangleright$  update normal field cell  $c_j$ 
    | | | if first update then
    | | | |  $t(c_j) \leftarrow t(s_i)$   $\triangleright$  update layer-tag field cell  $c_j$ 
    | | | |  $\vec{\mathbf{c}}(c_j) \leftarrow \vec{\mathbf{c}}(s_i)$   $\triangleright$  update color field cell  $c_j$ 
    | | | if  $d(c_j)$  differs from  $d(c_j)$  of the previous frame then
    | | | | expand a dirty AABB by  $[(x, y, z)^T + \vec{\mathbf{1}}, (x, y, z)^T - \vec{\mathbf{1}}]$ 
    | | | | update previous  $d(c_j)$ 
    | | exit atomic section
  end procedure

```

<sup>39</sup>  $d_{\text{def}} < 0$ , e.g.  $d_{\text{def}} = -1$  is a reasonable value.<sup>40</sup> There is also the possibility to formulate a smoother surface by using one of many metaball functions, though when looking at such small neighbourhoods the impact would be neglectable.

**Figure 7.20: Metaball visualization.** Experiments with raytracing metaballs of a small spherepacking ( $\sim 2$  k spheres), with random sphere colors.

**Left** [Blinn, 1982].

**Right** [McCormack and Sherstyuk, 1998].

There are many possible other ways to formulate the distance function, such as metaballs (sometimes also called blobbies) [Blinn, 1982], which would nicely fit the spherepacking representation (see Figure 7.20). However, from my experience, metaball surfaces excel, when there are few large spheres (or even ellipses) with noticeable overlap – all of these requirements are not met. In fact, quite the opposite – the spherepacking is completely polydisperse, and spheres are purposefully selected to be of small size, e.g. less than ten times the size of a voxel<sup>41</sup>. However, this is parameter that I have tweaked for each sphere packing individually, so far. Another factor is performance, some metaball functions without any bounds are in  $O(nm)$ , where  $n$  is the grid resolution and  $m$  is the sphere count, as each sphere needs to be considered in the computation of one distance value. This is considerably worse than my algorithm, which is in  $O(10m) = O(m)$ , as I visit each sphere once and

<sup>41</sup> The reason for this low maximum is that it works much better with my material removal algorithm (see Section 7.4).

only accumulate its contribution towards cells in its bounds (which, as I mentioned above, are about 10, or some other low constant amount of voxels maximally.) Metaballs have also been used to render organic cone-shaped structures, like arteries [Oeltze and Preim, 2004]. This is a possible future improvement to increase detail in the tooth's root anatomy or in soft tissues, if such an extension is ever developed.

“Wash your hands, put on your  
nightgown; look not so pale.”  
— William Shakespeare

### 7.6.1.1 Improving Visual Quality by Bilateral Smoothing

There several issues with the so far rather simple solution. Most importantly, since the SDF is generated from a spherepacking with a finite amount of spheres, the volume is not perfectly approximated, especially near the surface. The SDF in those areas is rather bumpy, which I want to remedy, as tooth are typically extremely smooth objects, especially the tooth crown. I have specifically received this feedback from a dentist when presenting a visualization without smoothing. Consequently, I developed a smoothing method, which is applied after the raw SDF has been generated. Figure 7.21 shows the triangulation of the raw and smoothed SDF – the surface is visible less bumpy. The basic idea is based on bilateral smoothing, which is a method to smooth RGB images, while retaining hard edges [Tomasi and Manduchi, 1998]. Bilateral smoothing weights the contribution of other pixels based on two relations: range and domain. In my data, the range relation is defined by the distance between the two voxel centers (influence controlled by  $\sigma_r$ ), and the domain is the difference of the signed distance to the implicit surface of both voxels (influence controlled by  $\sigma_d$ ). This concept nicely translates to 3D data, however, the speed is relatively slow, as each cell needs to compute the contribution of each other cell. In practice, I sped up the algorithm by having an additional parameter for the neighborhood dimensions (see  $N$  in Algorithm 7.10), which determines how many neighboring cells are considered during smoothing. This parameter needs to be carefully chosen, as small values might result in visible artifacts in the resulting SDF. The reason for this is, that large smoothing parameters ( $\sigma_r, \sigma_d$ ) can result in voxels having large contributions to far away voxels, which would be skipped if  $N$  was chosen too small to consider them. Therefore, choosing a too small  $N$  essentially results in calculating an incomplete weighted sum, resulting in a non-continuous smoothing (see Figure 7.22). A possible future improvement could be to automatically calculate a reasonable  $N$ , based on the range factor  $\sigma_r$ . I also sped up the algorithm by porting it to a GPU program that calculates voxels in parallel (see Algorithm 7.10).



Figure 7.21: Effect of distance smoothing. Comparing the triangulation of different SDFs.

Left Raw SDF.

Right SDF with moderate smoothing ( $\sigma_d = 1, \sigma_r = 3$ ) of the signed distances only (normals are not smoothed to focus on the influence of distances).



Figure 7.22: Effect of overly optimized smoothing.

Left  $N = 5$  – visible streaking artifacts along the main axes.

Right No artifacts, as  $N = 13$  is chosen large enough.

<sup>42</sup>These are indices for the SDF grid that mark the lower and upper bounds of cells overlapping the movement AABB.

---

#### Algorithm 7.10 Parallel Bilateral 3D SDF Smoothing

---

**Input:** raw 3D SDF, movement bounds  $b_{\min}, b_{\max} \in \mathbb{N}^{3 \times 42}$ , grid origin  $g_o \in \mathbb{R}^3$ , grid res.  $g_r \in \mathbb{N}^3$ , grid cell dim.  $g_d \in \mathbb{R}^3$

**Result:** smoothed 3D SDF

**procedure** FIELDSMOOTHING

$i_{lo} \leftarrow x(b_{\min}) + y(b_{\min}) x(r) + z(b_{\min}) x(r) z(r)$

$i_{hi} \leftarrow x(b_{\max}) + y(b_{\max}) x(r) + z(b_{\max}) x(r) z(r)$

**for** cells  $c_i$  in  $[i_{lo}, i_{hi}]$  **do in parallel**

▷ assign set of cells to each thread

$x \leftarrow i \% x(\vec{g}_r)$

$y \leftarrow i / x(\vec{g}_r) \% x(\vec{g}_r)$

$z \leftarrow i / (x(\vec{g}_r) y(\vec{g}_r))$

**if** any component of  $(x, y, z)$  outside  $b_{\min}, b_{\max}$  **then**

**continue**

▷ go to next cell assigned to this thread

$p \leftarrow d(c_i)$

▷ raw SDF cell  $c_i$

$\vec{c} \leftarrow \vec{c}(c_i)$

▷ raw color field cell  $c_i$

$\vec{n} \leftarrow \vec{n}(c_i)$

▷ raw normal field cell  $c_i$

$p' \leftarrow 0$

$\vec{c}' \leftarrow \vec{0}$

```

 $\vec{n}' \leftarrow \vec{0}$ 
 $w' \leftarrow 0$ 
write  $p$  to SDF cell  $c_i$ 
write  $\vec{n}$  to normal field cell  $c_i$ 
if raw color field cell  $c_i$  is not initialized yet then
  write  $(1, 1, 1)^T$  to color field cell  $c_i$ 
else
  write  $\vec{c}$  to color field cell  $c_i$ 
 $p(c_i) \leftarrow \mathbf{g}_s + (x, y, z)^T \mathbf{g}_d$ 
for all cells  $c_j$  within  $N \times N$ -neighbourhood44 do
   $x_2, y_2, z_2 \leftarrow$  3D index of 1D index  $j$ 
   $p(c_j) \leftarrow \mathbf{g}_s + (x_2, y_2, z_2)^T \mathbf{g}_d$ 
   $d \leftarrow \|p(c_j) - p(c_i)\|$ 
   $p_2 \leftarrow d(c_j)$ 
   $\vec{n}_2 \leftarrow \vec{n}(c_j)$ 
   $\vec{c}_2 \leftarrow \vec{c}(c_j)$ 
   $w \leftarrow \exp\left(-0.5 \left(\frac{d}{\sigma_r}\right)^2\right) \exp\left(-0.5 \left(\frac{p_2 - p}{\sigma_d}\right)^2\right)$ 
   $w' \leftarrow w$ 
   $p' \leftarrow p_2 w$ 
   $\vec{c}' \leftarrow \vec{c}_2 w$ 
   $\vec{n}' \leftarrow \vec{n}_2 w$ 
// Write weighted sum to smoothed SDF
if  $w > 0$  then
   $d_s(c_i) \leftarrow p' / w'$ 
   $\vec{c}_s(c_i) \leftarrow \vec{c}' / w'$ 
   $\vec{n}_s(c_i) \leftarrow \vec{n}' / (w' \|\vec{n}'\|)$ 
end procedure

```

▷ sum of weights<sup>43</sup>

<sup>43</sup>The weights could also be defined per channel, if one wanted them to not be equally smoothed, i.e. smooth distances stronger than colors, though my experiments showed no benefit to that.

▷ position of  $c_i$

▷ position of  $c_j$

▷ distance between  $c_i$  and  $c_j$

▷ raw SDF cell  $c_j$

▷ raw normal field cell  $c_j$

▷ raw color field cell  $c_j$

▷  $\sigma_r \neq 0, \sigma_d \neq 0$

<sup>44</sup> $N$  is odd, e.g.  $N = 11$  is a reasonable value, though it depends on  $\vec{r}$  and  $\sigma_r$  and  $\sigma_d$ . Padding is usually not necessary here, because I do not populate the SDF up to the bounds, and even when a neighbor cell causes a skip in a row or a column, those cells carry the same default value as other surrounding cells, and they will have a low weight due to the high distance.



Figure 7.23: Effect of normal smoothing.

Left No smoothing, normals are generated during SDF generation, incorrect normals lead to black spots.

Center Smoothing of distances, normals are generated after triangulation.

Right My smoothing well represents the original tooth model – no visible artifacts.

In Figure 7.21, the normals are not included, to see the effect that signed distance smoothing can have on its own. However, the smoothing effect does not look sufficient to reach the smoothness that real teeth exhibit. I have found that the much bigger impact on smoothness can be achieved from the surface normals (see Figure 7.23). For that, normals are generated during raw SDF generation (see Algorithm 7.9), by simply taking the direction from voxel center towards the center of the closest sphere. These normals do not work well without smoothing, as there is a considerable amount of noise, which comes from spheres being positioned on the inside of the object, relative to the voxel. This phenomenon is likely to happen near boundaries of the implicit object, like on the outside. However, smoothing of these noisy normals completely resolves this issue. Similarly, colors look much more convincing after smoothing, Figure 7.24 gives an indication on the results and also

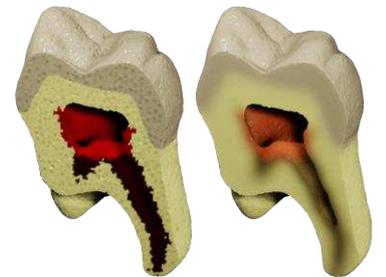


Figure 7.24: Color smoothing.

Left No smoothing.

Right Smoothing of colors (smoothed normals are not rendered, as the  $y$  not well defined during a cross-section).

the smoothing strength. When comparing the impact of smoothing of signed distances, normals and colors, there is a clear indication that signed distances need to be smoothed with larger smoothing factors. Leading to the need for individual  $\sigma_r, \sigma_d$  per field property that is being smoothed.

The process described calculates all distances in local space  $E$ , which might be differently scaled, dependent on the material. For example, the pelvic bone that I have received and worked with is in meters, whereas the tooth model that I designed is in millimeters. This is a difference in scale of factor 100, which results in considerably different values for distances<sup>45</sup>. With a small adjustment, the whole procedure can be made scale independent. The idea is to transform constants, such as  $\sigma_r$  and  $d_{\text{def}}$ , in a common space, such as world space  $W$ . Then, assuming the unit of  $W$  does not change, these parameters need to only be chosen once. Given the transformation from  $E$  to  $W$ ,  ${}^W H_E$ , adjust the constants

$$\sigma_r \leftarrow {}^W H_E^{-1} \sigma_r \tag{7.12}$$

$$d_{\text{def}} \leftarrow {}^W H_E^{-1} d_{\text{def}} \tag{7.13}$$

The resulting algorithm can handle the aforementioned pelvic bone and tooth model with  $\sigma_r = 3, \sigma_d = 1, d_{\text{def}} = -1$  with feasible results. See pelvic bone and femur in meters (Figure 7.26 and Figure 7.25) and tooth in millimeters (Figure 7.23). Of course, the best results can be achieved by tuning the parameters individually. However, without considering the scale, implausible results might be produced.

### 7.6.2 Fast Triangulation of the Signed Distance Field

The final major step in the visualization algorithm is the triangulation of the SDF. I have implemented a GPU program that uses the classic marching cubes algorithm to generate a list of triangles that represents the iso-surface at  $\alpha$ , at the chosen resolution (see Algorithm 7.11). Additionally, I perform trilinear interpolation of additional voxel data (normals and colors) in the same loop. I achieve fast parallelization by reserving the maximum amount of vertices  $|V|$  that can possibly occur:

$$|V| = x(\vec{g}_r) y(\vec{g}_r) z(\vec{g}_r) (5 \cdot 3) \tag{7.14}$$



<sup>45</sup>In fact, the pelvic bone model has such low coordinate values, that UE considers most vertices as duplicates and the normal generation consequently fails because of that.



Figure 7.25: Femur with unified parameters. The femur model converted to  $192^3$  SDF dynamic mesh, with vertices in meters, which results in considerably lower values than for tooth #16 (Figure 7.23). Still, the same set of parameters produces feasible results.

Top Static ground truth mesh.

Bottom Final dynamic mesh.

Figure 7.26: Pelvis with unified parameters. The pelvic bone model converted to  $128^3$  SDF dynamic mesh, with vertices in meters, which results in considerably lower values than for tooth #16 (Figure 7.23). However, with transformation of constants, a unified parameter range for the SDF generation and smoothing can be utilized ( $\alpha = 0, \sigma_r = 3, \sigma_d = 1, d_{\text{def}} = -1$ ).

Top left Static ground truth mesh.

Top right Spherepacking.

Bottom left Voxels with positive values.

Bottom right Final dynamic mesh.

since each voxel can at most result in 5 additional triangles, with 3 vertices each. At runtime, the actual amount of vertices is thread-safely incremented using `ATOMICADD()`, thereby guaranteeing unique memory locations for each vertex.

After generating the triangles, I perform a short postprocessing step to pool common vertices between separate triangles. The purpose of the pooling is to reduce the amount of individual vertices, to reduce the data that needs to be sent to the GE renderer. In practice, the largest slow-down occurs due to the amount of data that is sent to the GE renderer.

---

**Algorithm 7.11** Parallel Marching Cubes with Normals and Colors
 

---

**Input:** smoothed SDF

**Result:** coarse grid cells updated with new triangle-, normal-, color lists

```

procedure PARALLELMARCHINGCUBES
   $i_{l_0} \leftarrow x(b_{\min}) + y(b_{\min}) x(r) + z(b_{\min}) x(r) z(r)$ 
   $i_{h_1} \leftarrow x(b_{\max}) + y(b_{\max}) x(r) + z(b_{\max}) x(r) z(r)$ 
  for cells  $c_i \in [i_{l_0}, i_{h_1}]$  do in parallel ▷ assign set of cells to each thread
     $x \leftarrow i \% x(\vec{g}_r)$ 
     $y \leftarrow i / x(\vec{g}_r) \% y(\vec{g}_r)$ 
     $z \leftarrow i / (x(\vec{g}_r) y(\vec{g}_r))$ 
    if any component of  $(x, y, z)^T$  outside  $b_{\min}, b_{\max}$  then
      continue ▷ go to next cell assigned to this thread
     $(x_t, y_t, z_t)^T \leftarrow (x, y, z)^T / \vec{g}_r^t$ 
     $i_t \leftarrow x_t + y_t x(\vec{g}_r^t) + z_t x(\vec{g}_r^t) y(\vec{g}_r^t)$ 
    if all components of  $x_t, y_t, z_t$  within  $b_{\min}^t, b_{\max}^t$  46 then
      flag coarse cell  $i_t$  as dirty
      // Arrays for data samples at 8 voxel corners
       $v[j] \in V \subset \mathbb{R}^3, j \in [0, 8) \leftarrow \emptyset$  ▷ store 8 voxel vertices
       $\vec{n}[j] \in N \subset \mathbb{R}^3, j \in [0, 8) \leftarrow \emptyset$  ▷ store 8 voxel normals
       $\vec{c}[j] \in C \subset \mathbb{R}^3, j \in [0, 8) \leftarrow \emptyset$  ▷ store 8 voxel colors
       $p[j] \in P \subset \mathbb{R}, j \in [0, 8) \leftarrow \emptyset$  ▷ store 8 voxel distances
      // Arrays for resulting triangle vertices
       $v'[k] \in V' \subset \mathbb{R}^3, k \in [0, 15) \leftarrow \emptyset$  ▷ 5 triangles × 3 vertices = 15
       $\vec{n}'[k] \in N' \subset \mathbb{R}^3, k \in [0, 15) \leftarrow \emptyset$  ▷ store 15 result vertices
       $\vec{c}'[k] \in C' \subset \mathbb{R}^3, k \in [0, 15) \leftarrow \emptyset$  ▷ store 15 result normals
      ▷ store 15 result colors
      // Classify current voxel based on the corner values
       $i_f \leftarrow 0$  ▷ index which vertices are in or out (8 · 8 = 256)
      for all  $k \in [0, 8)$  do
         $i_c \leftarrow x + L_{\text{off}}[i_c, 0] + (y + L_{\text{off}}[i_c, 1]) x(\vec{g}_r) + (z + L_{\text{off}}[i_c, 2]) x(\vec{g}_r) y(\vec{g}_r)$ 
         $x(v[k]) \leftarrow x(\vec{g}_o) + x(\vec{g}_d) (x + L_{\text{off}}[k, 0])$ 
         $y(v[k]) \leftarrow y(\vec{g}_o) + y(\vec{g}_d) (y + L_{\text{off}}[k, 1])$ 
         $z(v[k]) \leftarrow z(\vec{g}_o) + z(\vec{g}_d) (z + L_{\text{off}}[k, 2])$ 
         $p[k] \leftarrow$  SDF value at 1D index  $i_c$ 
         $\vec{n}[k] \leftarrow$  normal field value at 1D index  $i_c$ 
         $\vec{c}[k] \leftarrow$  color field value at 1D index  $i_c$  47
        if  $p[k] \leq \alpha$  then ▷ current cube vertex is inside implicit object
           $i_f \leftarrow i_f | 1 \ll k$  ▷ set the  $k$ th bit of  $i_f$  to 1 48
      // Triangulate the current voxel
       $e_f \leftarrow L_{\text{edge}}[i_f]$  ▷ get edge flag from LUT at flag index
      if  $e_f \neq 0$  then
         $v_{\text{edge}}[l] \in V \subset \mathbb{R}^3, l \in [0, 12) \leftarrow \emptyset$  ▷ store up to 12 edge vertices
        for all  $i_{\text{edge}} \in [0, 12)$  do
          if  $e_f \& (1 \ll i_{\text{edge}})$  then ▷ is the  $i_{\text{edge}}$ th bit set?
             $i_{\text{edge}}^{\text{start}} \leftarrow L_{\text{edge}}[i_{\text{edge}}, 0]$ 
             $i_{\text{edge}}^{\text{end}} \leftarrow L_{\text{edge}}[i_{\text{edge}}, 1]$ 
             $\Delta p \leftarrow p[i_{\text{edge}}^{\text{end}}] - p[i_{\text{edge}}^{\text{start}}]$ 
             $\Delta v \leftarrow 0.5$ 
            if  $\Delta p = 0$  then
               $\Delta v \leftarrow (\alpha - p[i_{\text{edge}}^{\text{start}}]) / \Delta p$ 
             $v_{\text{edge}}[i_{\text{edge}}] \leftarrow v[p_{\text{edge}}^{\text{start}}] + (v[p_{\text{edge}}^{\text{end}}] - v[p_{\text{edge}}^{\text{start}}]) \Delta v$ 
          for all  $i_{\text{tri}} \in [0, 5)$  do
            if  $L_{\text{tri}}[i_f, 3 i_{\text{tri}}] < 0$  then
              break
             $i_{t_c} \leftarrow$  ATOMICADD( $N_v[i_t]$ , 3) ▷ up to 5 triangles are generated per voxel
            for all  $i_v \in [0, 3) \subset \mathbb{N}$  do ▷ LUT entry -1 indicates end of triangles
              ▷ vertex count of coarse cell 49 at  $i_t$ 

```

<sup>46</sup>The movement bounding box is here required in the coarser resolution, this transformation is done in `MESHUPDATEASYNC()`.

<sup>47</sup>This can be omitted when making a difference coloring (e.g. using binary classification), as the color is determined later.

<sup>48</sup>This will create a bitmask that is 1 for cube vertices that are inside, and 0 for those outside the implicit object, giving 256 possible cases which are handled in LUTs.

<sup>49</sup> $N_v[i_t]$  stores the information of how many vertices we added to the coarse cell, in order to know which of the data later on is useful, as sizes need to be predetermined, since memory allocations during GPU kernel would hinder performance.

```

ir ← 3 itc + iv
i'v ← Ltri[if; ir]
v'[ir] ← vvedge
▷ store resulting vertex at local index ir

// Trilinear interpolation of 8 normals and colors
 $\vec{d} \leftarrow \frac{v_{\text{edge}}[i'_v] - v[0]}{v[6] - v[0]}$ 
 $\vec{d}^{-1} \leftarrow (1, 1, 1)^T - \vec{d}$ 
 $\vec{n}'[i_r] \leftarrow \vec{n}[0] x(\vec{d}^{-1})y(\vec{d}^{-1})z(\vec{d}^{-1}) + \vec{n}[1] x(\vec{d})y(\vec{d}^{-1})z(\vec{d}^{-1})$ 
+  $\vec{n}[2] x(\vec{d})y(\vec{d}^{-1})z(\vec{d})$  +  $\vec{n}[3] x(\vec{d}^{-1})y(\vec{d}^{-1})z(\vec{d})$ 
+  $\vec{n}[4] x(\vec{d}^{-1})y(\vec{d})z(\vec{d}^{-1})$  +  $\vec{n}[5] x(\vec{d})y(\vec{d})z(\vec{d}^{-1})$ 
+  $\vec{n}[6] x(\vec{d})y(\vec{d})z(\vec{d})$  +  $\vec{n}[7] x(\vec{d}^{-1})y(\vec{d})z(\vec{d})$ 

 $\vec{c}'[i_r] \leftarrow \vec{c}[0] x(\vec{d}^{-1})y(\vec{d}^{-1})z(\vec{d}^{-1}) + \vec{c}[1] x(\vec{d})y(\vec{d}^{-1})z(\vec{d}^{-1})$ 
+  $\vec{c}[2] x(\vec{d})y(\vec{d}^{-1})z(\vec{d})$  +  $\vec{c}[3] x(\vec{d}^{-1})y(\vec{d}^{-1})z(\vec{d})$ 
+  $\vec{c}[4] x(\vec{d}^{-1})y(\vec{d})z(\vec{d}^{-1})$  +  $\vec{c}[5] x(\vec{d})y(\vec{d})z(\vec{d}^{-1})$ 
+  $\vec{c}[6] x(\vec{d})y(\vec{d})z(\vec{d})$  +  $\vec{c}[7] x(\vec{d}^{-1})y(\vec{d})z(\vec{d})$ 

// Add vertex, normal & color to encompassing coarse grid cell ir
write v'[ir] to coarse distance grid cell ir at array position ir
write  $\vec{c}'[i_r]$  to coarse color grid cell ir at array position ir
write  $\vec{n}'[i_r]$  to coarse normal grid cell ir at array position ir

end procedure
    
```

The algorithm creates a hash map that maps the 3D position vector that represents the vertex to some additional data: normal, color, index of the corresponding vertex (the vertex itself is omitted here, as it can be read from the key). This hashmap is created for each coarse grid cell, as this is the extend of the pooling. One might think that increasing the pooling extend, e.g. to be global, would have a large effect on the data efficiency. In fact, we are only losing pooled vertices along coarse grid borders, i.e. for  $g_d^t = 8$ , we are losing around  $\frac{8 \cdot 8 \cdot 6 + 4}{10 \cdot 7^3} = 11.31\%$ <sup>50</sup> of memory efficiency on average. Additionally, this would complicate the algorithm, as the vertex indices are expected in a local manner by the GE renderer. Still, this is a possible avenue for further improvement.

<sup>50</sup>I considered the amount of neighbouring cells on coarse cell borders vs. the amount of neighbouring cells inside a coarse cell.

### Algorithm 7.12 Vertex Pooling

**Input:** 3D grid with redundant vertex-data

**Result:** non-redundant vertex-data; indexGrid that maps to vertex-data

```

procedure VERTEXPOOLING
  for all dirty coarse grid cells ci do
    bi ← empty hash map (3D Vector ↦ VoxelData)
    for all vertices vj in ci do
      if bi does not contain vj then
         $d_j \leftarrow \{\vec{c}(v_j), \vec{n}(v_j), i(v_j)\}$ 5152
        insert key-value pair  $\mathbf{p}(v_j) \mapsto d_j$  into bi
      else
▷ Found another vertex vk with  $\mathbf{p}(v_k) = \mathbf{p}(v_j)$ 
        indexGrid[id][j] ← i(vk)
    ic ± 1
    for all entries vj in bi do
      triangleGrid[id][i(vj)] ←  $\mathbf{p}(v_j)$ 
      normalGrid[id][i(vj)] ←  $\vec{n}(v_j)$ 
      colorGrid[id][i(vj)] ←  $\vec{c}(v_j)$ 

end procedure
    
```

<sup>51</sup>The local index to reference this vertex from within the same coarse grid cell.

<sup>52</sup>I omit  $\mathbf{p}(v_j)$  in the value, since I store it as the key.

**Part II**

**Application**

RELATED PUBLICATIONS

**P11.** *Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks* . . . . . 272

OUTLINE

8.1 Bi-Manual VR Dental Simulator . . . . . 93

8.1.1 Synthesizing the Sound of a Dental Handpiece . . . . . 95

8.1.2 Simulating Optical Magnification inside VR . . . . . 99

8.1.3 Improved Eye Tracking for User Gaze Analysis . . . . . 100

8.2 Large Forces for Hip Surgery Training Simulator . . . . . 103

8.2.1 Rendering of Extreme Forces during Surgical Hammering . . . . . 105

“Mine eyes are made the fools o’ the other senses.”

— William Shakespeare

IN this chapter, I want bridge the gap between the so far theoretical content and the actual integration into a full simulator, including integration into a GE. I have integrated my methods into two simulators, which I will describe further:

1. Dental simulator (see [Section 8.1](#))
2. Hip surgery simulator (see [Section 8.2](#))

Both simulators also necessitated the development of additional features to further improve the realism of the experience when using them. Of course, there is no general solution to developing simulators, and the details of the use-case will almost always differ and thus, require individual solutions.

### 8.1 BI-MANUAL VR DENTAL SIMULATOR

In cooperation with the Mahidol University, I have developed a VR-based simulator, based on the haptic rendering method that I have developed. The simulator supports virtually performing the material removal of the two following surgical procedures

1. **Caries removal** is the removal of tooth tissue that is infected by caries. The infected tooth has certain regions of enamel and sometimes dentin infected with caries. The tissue can be distinguished by the darkened color and reduced specular reflectance (see [Figure 8.1](#)). Additionally, the haptic sensation when drilling is softer and “sticky”, as it was described to me by a dentist.
2. **Root-canal** is necessary to do when caries has reached far into the tooth and the pulp is infected. It involves removing infected pulp and nerves inside the root canals. The simulated material removal is involved in the preparation, during the *access opening*, where an access hole is drilled into the top surface of the infected tooth (see [Figure 8.2](#)). Such a hole ought to be centered, have a smooth outline, and ought to be just large enough that all root orifices are visible from distinct viewpoints<sup>1</sup>. It is especially important to leave all walls and floor intact and thick enough to have structural integrity, otherwise the tooth might break in the future<sup>2</sup>.

Development started in 2017, based on UE 4.16. I have created a virtual operating room by placing medical assets, such as patient seat and lamp, and a patient based on UE’s digital human (see [Figure 8.3](#)), which is the predecessor to the more recent meta human, in which the user can design how the human shall look exactly. I modified the human model in the following way:

- rigged into the lying pose
- a surgical cloth was generated by using Blender’s cloth simulation to generate a static mesh
- the mouth inside was lined with red-textured gum-geometry
- added a simple dentation model, with tooth #26 removed – this hole was filled with the dynamic mesh that is generated at run-time

The simulator already supported bi-manual haptic interaction, with a mirror and dental handpiece being attached to the left and right haptic devices.



Figure 8.1: Carious tooth. Tooth model with caries – the dark coloration indicates carious tissue.



Figure 8.2: Root-canal procedure. Tooth model during root-canal – red color indicates pulp floor and dark colors indicate orifices.

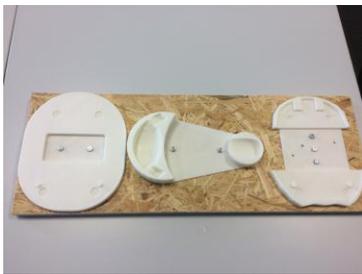
<sup>1</sup>Requiring a common viewpoint at which all orifices are visible would often require a hole that is too large.

<sup>2</sup>This happens in around 8% of cases after 5 years [Chen et al., 2008]. Still, root-canals are the only alternative to extraction, thus, it is relevant to try to increase their success rate.



Figure 8.3: Initial dental simulator patient. The patient model, showing the custom dentation, including tooth #26 with correct anatomy.

**Figure 8.4: Initial dental VR simulator.** The patient is located in a roughly modeled hospital room, which includes dental chair, and mobile dental stool. The patient is covered by a static mesh, generated from cloth simulation. The patient was modeled using UE’s digital human “Buck” (which seemingly is no longer available online). Epic Games used to regularly release new versions of digital humans, of which Buck was the latest around 2018. Today, these iterations are replaced by the Metahuman platform, which allows user customizable human models. As the earlier digital humans had no mouth anatomy, I added the inside of the mouth (see Figure 8.3). The environment is kept relatively simple, and there are no windows, which is not rather uncommon.



**Figure 8.5: Initial dental simulator registration.** I first designed the setup to mount the VR controller in the middle. However, it turned out to be better to mount it to either side, as it produces more reliable tracking and the distance between the haptic devices can be shortened.

The haptic devices were also registered inside the VR system by a simplified method, of which the math is detailed in Section 6.2. The simplification does not support modifying the device separation distance. The idea is to mount both haptic devices at a constant offset to a VR controller (see Figure 8.5), whose pose is naturally being tracked by a proprietary VR tracking system. Thus, by linear transformation of that pose, based on the constant offset, I can infer the pose of both haptic devices in the VR system. Finally, the virtual camera pose is offset, such that the poses of virtual tools and real devices are equal.

The dental VR simulator was later redone from scratch in 2021 in the newer UE 4.27<sup>3</sup>, with a new virtual patient based on UE’s meta human. During this redone, all of the previously mentioned features are included in the bi-manual VR simulator (see Figure 8.6). The following new features were included, of which most will be explained in this section:

1. Registration with arbitrary virtual and real offsets – these can be independently set and midpoint is registered to minimize the offset<sup>4</sup>.

**Figure 8.6: VR Simulator.** A dental student of Thammasat University is using my bi-manual VR dental simulator with two force-feedback devices, as part of a user study. The view of the student is overlaid in the top left corner.



2. Synthesis of drilling sounds to increase immersion.
3. Screen space zooming in VR to resemble optical magnification of surgical loupes, which some dentists utilize in reality.
4. Eye-tracking with a gaze accuracy tests procedure – The integration into UE was rewritten by me, improving upon the official manufacturer’s plugin for UE. Of course, my eye-tracking fully supports the custom zooming (to avoid problems as shown in [Figure 8.13](#)). In fact, eye-tracking accuracy when doing ray casts into the scene is even increased during zooming, as the world appears enlarged on screen.

<sup>3</sup>This version is close to UE 5, which should simplify a future version upgrade.

<sup>4</sup>A small mismatch between virtual tools and real devices proved useful to me to reduce probability of interference of left and right haptic devices, as (1) the end-effector is larger, and (2) the haptic arm up until the end-effector do not exist inside the simulator. In such a case the virtual separation is chosen larger as the real separation.

Additionally, the graphical quality was considerably increased by using higher quality assets, many of which were custom made by me. More importantly, the human model is based on UE’s meta human. I adjusted the dentation by turning the face texture atlas into a transparency-enabled material and painting the spot of tooth #26 and #36 to have alpha set to 0. Consequently, the meta human dentation is missing those teeth, which I then fill up by overlaying the anatomically correct tooth models (details in [Section 7.3.2](#)).

### 8.1.1 *Synthesizing the Sound of a Dental Handpiece*

The VR simulator feeds the visual and haptic sense well, though sound is not at all considered so far, thus, its inclusion would considerably increase the simulator’s immersion. Moreover, sound is an important sensation during dental surgery, as the surgeon can easily tell when material is being removed. A diamond bur is rotating at a high speed of 365 000 RPM, which creates a very distinct high-pitched sound<sup>5</sup> upon contact with the tooth material, which gives the surgeon a clear indication that material is being removed. According to [Altinöz et al.](#), the frequency of a dental handpiece with bur in free space is around 5.5 kHz, and increases to around 8 kHz when removing material, though the latter value varies greatly when the measurement is repeated without changing any of the materials. There are even measurements with a lower pitch at around 4.6 kHz. The great variation during contact and my own experience lead me to believe, that the exact pitch of the drilling sound might also provide a hint as to how large the contact area between bur and removable material is, for which I have found further evidence when looking at detailed video recordings<sup>6</sup>. Additionally, I have evaluated a frequency analysis of a sound sample recorded inside a dental operating room during dental drilling (see [Figure 8.7](#)). There, I discovered the distinct starting and stopping sound signature of the dental handpiece. The pitch of the sound quickly ramps up as the rotation speed increases. Similarly, when the motor is turned off the pitch of the sound quickly ramps down to zero as the rotation speed equally decreases to zero.

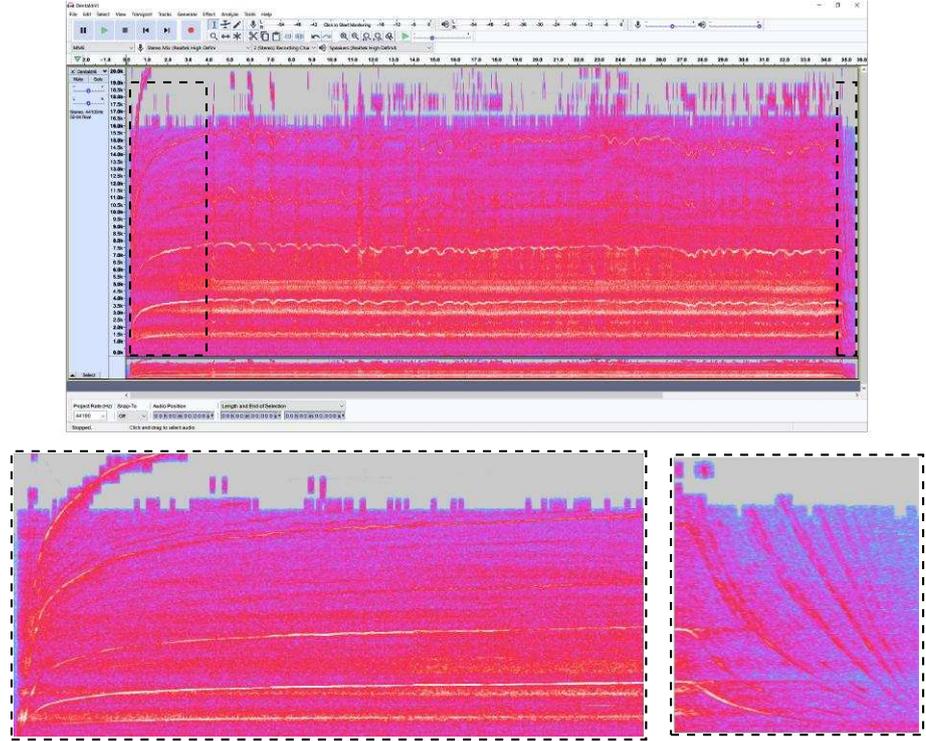
<sup>5</sup>Fear of dental procedures is a common fear, and some people report feeling uneasy when they hear this kind of sound [[Antoniadou et al., 2022](#)].

<sup>6</sup>Unfortunately, their copyright prevents me from including them here.

I have used these observations to design a sound synthesis that meets these requirements. My idea was to generate two separate, long (around 1 minute or longer, such that repetition is easily implementable) sound samples – one for each of the two frequencies: spinning in free space, and removing material through drilling contact. Both sound samples are a collection of white noises that are each centered around different frequency (by using a notch filter). The set of noise frequencies are chosen such that the overall sample frequency is centered around 4 kHz, to allow further up-pitching at runtime, based on dynamic parameters. The free spinning sound starts playing wherever the drilling button is pressed on the haptic device. Volume and pitch of the sample are ramped up by custom functions, dependent on the time  $t \in [0, 1]$ <sup>7</sup>, to create a realistic ramp up sound.

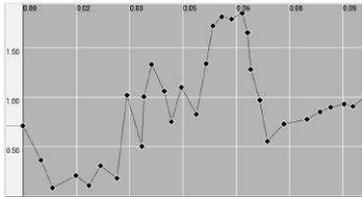
<sup>7</sup> $t$  is normalized in the functions, however, the time that I used to stretch the full ramp-up in real-time is 2 s.

**Figure 8.7: Sound sample frequency.** I have taken a sound recording from inside a dental clinic and analyzed the frequency distribution. This has revealed that the starting up of a handpiece has a distinct frequency raise function. A set of main frequencies (shown in brighter color) reach a certain level, as soon as the handpiece motor was on for around four seconds, at which point the high speed of the motor is reached and the pitches settles at the respective highest points. When the motor is turned off, the rotation speed drops off rapidly; after not even one second all movement stops. Consequently, an equally sharp drop in the frequencies is visible on the frequency diagram. The sound inbetween turning the motor on and off closely resembles noise around each of the main frequencies.



$l$	$b$	$c_{\text{down}}$	$c_{\text{up}}$
0.4	1.3	0.5	0.00001

**Table 8.1: Spin sound parameters.** The parameter set that I have used to implement the spinning sound modulation.



**Figure 8.8: Volume ramp up.** The volume increase during dental drill start-up. Manually designed via keyframes in the UE editor, based on sound samples. The  $x$ -axis shows time in seconds, and the  $y$ -axis the volume as a factor.

<sup>8</sup>For the pitch  $p_s^-$ , UE does not allow a pitch factor less than 0.4, therefore this function will not approach 0 (as  $v_s^-$  does), but 0.4. This can be achieved by setting  $p_s^-(0)$  to the desired value end value, after the timeline has terminated.

The pitch ramp-up  $p_s^+$ , when the motor starts up, follows the function

$$p_s^+(t) = l + (b - l) t^{1-c_{\text{up}}} \quad (8.1)$$

with all parameters shown in Table 8.1. The volume ramp-up follows a custom key-framed function, which I manually designed based on sound samples (see Figure 8.8).

The ramping-down, when the motor is turned off and slows down, is a bit more complicated, as the speed of the motor (i.e. how long the user has been holding the drilling button) needs to be taken into consideration. The down-ramp pitch  $p_s^-$  and volume  $v_s^-$ <sup>8</sup> are defined by following function, dependent on the passed time  $t$ :

$$p_s^-(t) = v_s^-(t) = \frac{\frac{t}{c_{\text{down}}^{\frac{1}{t_{\text{max}}}}} - 1}{c_{\text{down}} - 1} [p_s^-(0) - p_s^+(t_{\text{max}})] + p_s^+(t_{\text{max}}) \quad (8.2)$$

In both cases, the length of the timeline  $t_{\text{max}}$  is determined by the timeline progression of the ramp-up when the event was fired. For example, if the ramp-up timeline has progressed to 40% of the duration, i.e.  $t = 0.4$  s, then the down-ramp will take in total 0.4 s, as the motor tends to spin down quicker if it was spun up for a shorter duration. Consequently, the down-ramp will start off where the ramp-up has stopped, which is especially relevant if it was stopped before the full pitch was reached (see in Figure 8.10, that each premature stop still looks plausible, as the pitch does not jump when stopping). This ensures, that the modulation is done continuously, without any instantaneous changes in pitch or volume. Otherwise, the down-ramp would always start with  $t = 0$  at the highest pitch, even if the ramp-up has not reached that pitch yet. I programmed this into UE by using blueprint events that are fired when the haptic device buttons for the drilling are pressed (see Figure 8.9). A frequency response, similar to the real sample (see Figure 8.7), can now be produced by parameterizing the noise sample with the pitch and volume at runtime, based on the passed time (see Figure 8.10).

The next part of the synthesis is the actual drilling contact that takes away material through cutting. Here, I will make use of my own observation on contact area and sound pitch, based on watching and listening to videos of tooth cutting:

## 8.1. BI-MANUAL VR DENTAL SIMULATOR

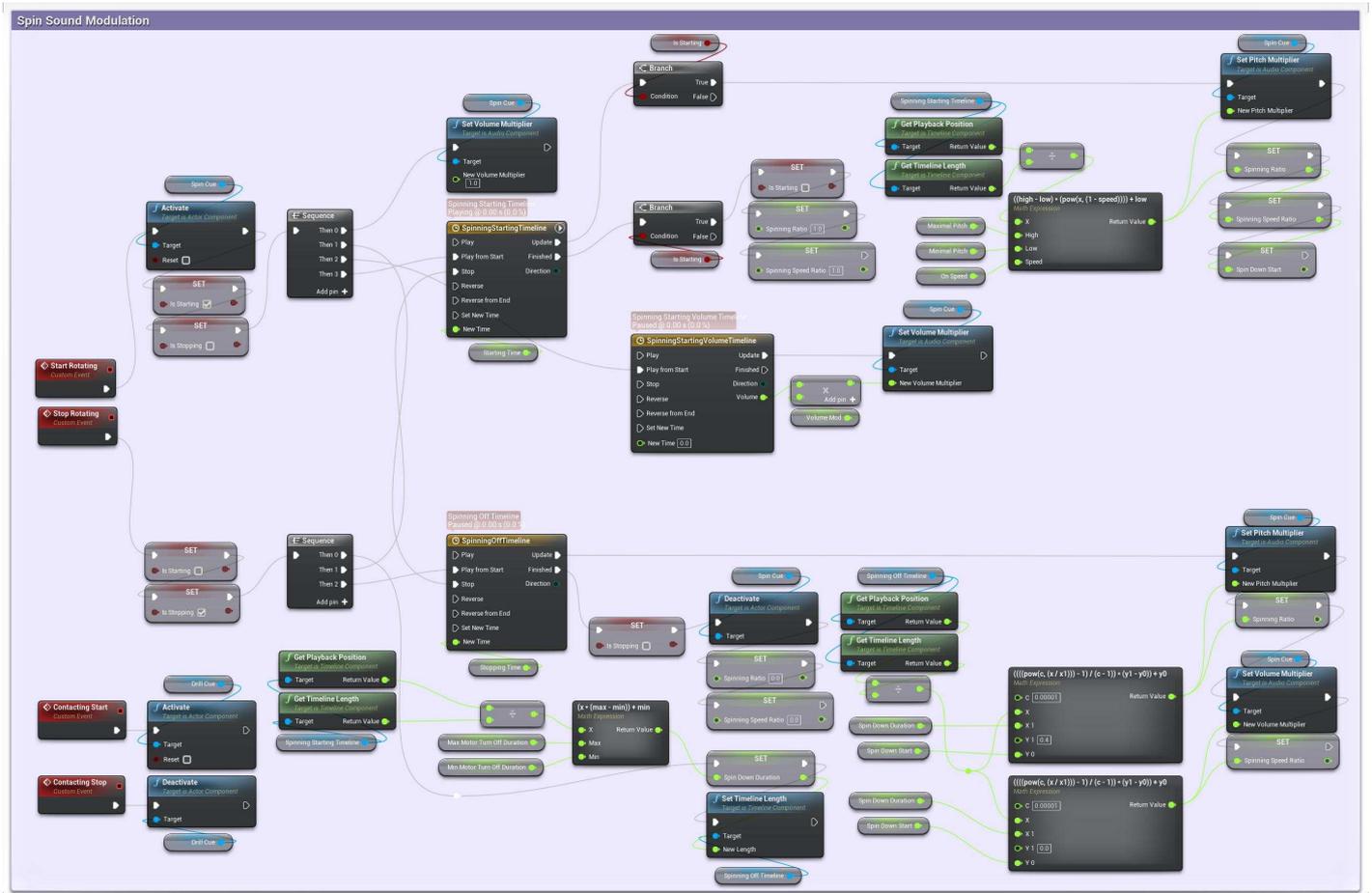


Figure 8.9: Spinning sound synthesis blueprint. The UE blueprint that performs the spinning sample modulation via pitch and volume. When the drill button is pressed and released, the events Start- and Stop Rotation are called, respectively.

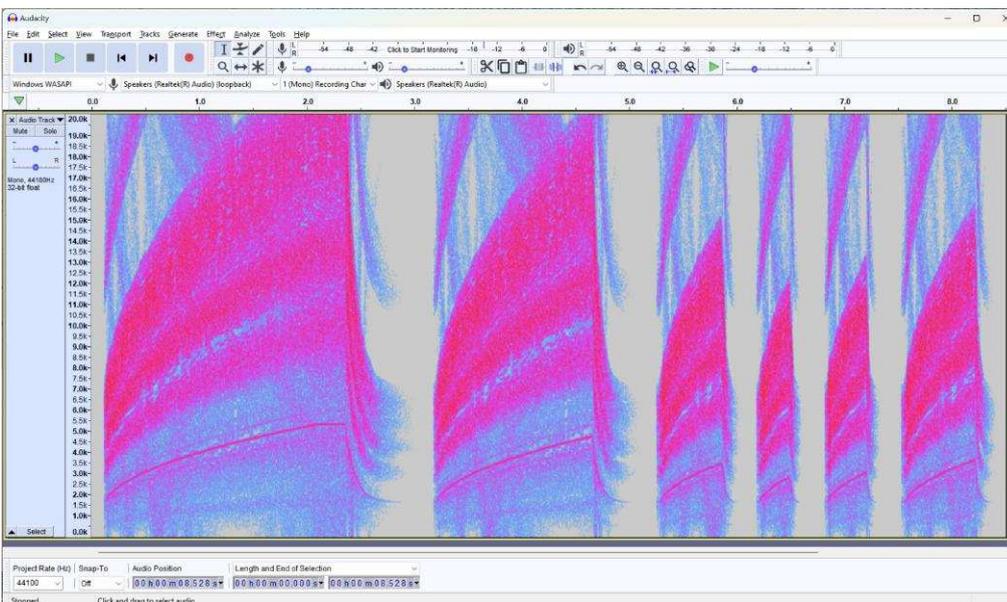


Figure 8.10: Simulated sounds. A custom static spinning sample is modulated by dynamically changing the pitch at runtime. The characteristic ramp-up and ramp-down and overall frequency response resembles the real sound recording (see Figure 8.7). Notice also, that prematurely stopping the motor will result in plausibly adjusted (both in initial pitch and duration) motor down-ramp sound progression.

1. larger contact area (i.e. more material being removed) results in lower pitch – presumably because the motor can not overcome the cutting resistance as easily and the actual rotation speed is lowered
2. larger contact area (i.e. more material being removed) results in slightly higher volume – presumably because the larger contact area results in increased cutting friction, which results in more sound waves

These observation build the basics of my sound synthesis for the cutting sound. The amount of material that is removed during one time-step can be easily tracked by summing up the  $d$  value (before division by two) of all threads inside `MATERIALREMOVALPARALLEL()` (see Algorithm 7.2). The sum  $\sum d = r$  describes the overall distance that all radii of the environment spheres were accumulatively reduced by during this time-step. This value can fluctuate a lot, therefore using a moving average provides a better approximation, which I will call  $\bar{r}$ . The pitch of the drilling sound  $p_d(\bar{r})$  is then inversely linearly interpolated, as follows:

$$p_d(\bar{r}) = p_d^{t=1} - \left[ p_d^{t=0} + \left( \bar{r} - r_0 \right) \frac{p_d^{t=1} - p_d^{t=0}}{r_1 - r_0} \right] \quad (8.3)$$

$p_d^{t=0}$	$p_d^{t=1}$	$v_d^{t=0}$	$v_d^{t=1}$	$r_0$	$r_1$
2	3	0	0.3	1	1.5

Table 8.2: Drill sound parameters. The parameter set that I have used to implement the drilling sound modulation. Especially the removal values will likely be very different for other simulations, as this heavily depends on the world scale.

Consequently, the pitch  $p_d(\bar{r})$  gets lower, as the amount of material being removed  $\bar{r}$  increases, which fulfills the first requirement, mentioned earlier. The volume of the drilling sound  $v_d(\bar{r})$  is also linearly interpolated, as follows:

$$v_d(\bar{r}) = v_d^{t=0} + \left( \bar{r} - r_0 \right) \frac{v_d^{t=1} - v_d^{t=0}}{r_1 - r_0} \quad (8.4)$$

Thus, the volume  $v_d(\bar{r})$  gets higher, as the amount of material being removed  $\bar{r}$  increases, which fulfills the second requirement. Additionally, the volume multiplier from the spin sound modulation is also used to scale the drilling sound volume. This improves realism and coherence, as the drill can not produce loud noises as the drill motor starts up or slows down, as it would in the real-world. I programmed these equations into a UE blueprint (see Figure 8.11) by listening to the Tick-event, which is fired during each new render frame, in which pitch and volume of the drilling sound are adjusted accordingly (used parameters are shown in Table 8.2).

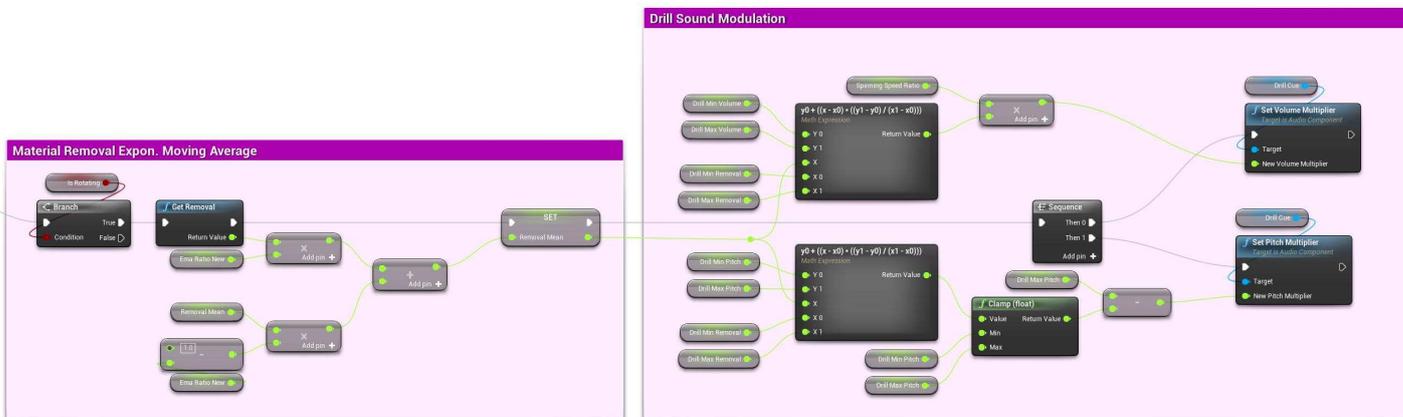


Figure 8.11: Drilling sound synthesis blueprint. The UE blueprint that performs the drilling sample modulation via pitch and volume. The amount of material being removed determines volume and inversely determines pitch.

8.1.2 *Simulating Optical Magnification inside VR*

In reality, dentists regularly make use of magnifying binoculars during their procedures, as even in reality, seeing small details is a challenge for the eye. Consequently, rendering such small detail using a pixel raster, which is not even close to the limits of the human eye, leads to considerable further difficulties for users' vision. Therefore, I implemented a similar feature in my VR simulator that allows users to switch between different levels of optical magnification on the fly (1x, 2x, 4x, 8x, 16x). However, the UE does not allow developers to make any changes to the camera projection when the scene is rendered in an HMD, including field of view (FOV), which likely would have simplified the implementation process. In most cases, it makes sense to control the rendering parameters, in order to prevent motion sickness due to parameters that contradict the physical world, e.g., by an incorrect field-of-view. However, I believe this case warrants an exception for optical magnification, for the following reasons:

1. There is a clear physical meaning behind the magnification since dentists often use surgical binoculars with optical magnification, thus, it does not contradict the physical world.
2. The users are themselves in control of the magnification. If they happen to dislike it, or feel sick, they can quickly decide to disable it with ease.
3. The user is always seated, and during the procedure, only very small, and very controlled head movements are necessary.
4. The feature enables the user to see the tooth and, in particular, the root canal more clearly, since most current HMDs do not offer enough resolution to render such details clearly<sup>9</sup>.

To implement the optical magnification, it was necessary for me to modify the source code of UE, as no capable interface is available by default. The optical zoom of a factor  $m$  is implemented by manipulating UE's default stereo projection matrix  ${}^S\mathbf{H}_W$  (which transforms a 3D point in camera space to a 2D point in screen space) as follows:

$${}^S\mathbf{H}_W \leftarrow \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & m & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^S\mathbf{H}_W \quad (8.5)$$

I implemented this modification of UE in the function `GetStereoProjectionMatrix` of SteamVR and OpenXR. Thus, a point that would normally be projected onto  $\mathbf{p}$  in screen space is instead projected onto  $m\mathbf{p}$ . This effectively zooms the screen image from the screen center by a scaling factor of  $m$ .

Additionally, I needed to adjust the positions of both eyes, which are used in SteamVR and OpenXR's `GetRelativeEyePose` (see [Code 8.1](#) for the adjusted version). In that function, the eye-to-head transformation is created from some translation  $\vec{t}$  and a rotation. The distance of the eyes to the HMD origin needs to be scaled down by

$$\vec{t} \leftarrow \vec{t} \frac{1}{m} \quad (8.6)$$

where  $m$  depends on the magnification factor chosen by the user. Without this correction, my magnification would also incur an increased stereopsis, which needs to remain unchanged, in order to produce correct spatial impressions<sup>10</sup>. The choice of  $m$  defaults to 1 (in which case  $m\mathbf{p} = \mathbf{p}$  is the unmodified screen position) and can be adjusted by the user by pressing the front or back button of the left 3D Systems Touch. By pressing the forward

<sup>9</sup>This was clearly demonstrated in the prior user-study (see [Section 9.1.1](#)), and thus, lead me to implement this feature.

We will see in the data later-on that most users make frequent use of the magnification (see [Section 9.1.3](#)).

<sup>10</sup>My impression was that objects felt closer than they should be. I have not thoroughly investigated this effect, but subjectively the problem was readily apparent. Likewise, the eye position adjustment subjectively completely fixed this issue, and no users have reported feeling that something is "off", even though around 50 people have used the simulator over an extended period of time.

<sup>11</sup>Previously, I had implemented magnification with a stepless slider, that can be moved by pressing the up and down buttons. However, this overcomplicated the zooming process, as it was slower and previous settings were not easily repeatable. Additionally, real surgical loupes typically only have one or two distinct magnification levels, as each level requires a separate lens.

button, the factor  $m$  is increased until the next higher number in the series (1, 2, 4, 8, 16), and the back button decreases to the next lower number in the series<sup>11</sup>.

```
bool FSteamVRHMD::GetRelativeEyePose(int32 DeviceId, EStereoscopicPass
Eye, FQuat& OutOrientation, FVector& OutPosition)
{
    if (DeviceId != IXRTrackingSystem::HMDDeviceId || !(Eye ==
        eSSP_LEFT_EYE || Eye == eSSP_RIGHT_EYE))
    {
        return false;
    }
    auto Frame = GetTrackingFrame();

    vr::Hmd_Eye HmdEye = (Eye == eSSP_LEFT_EYE) ? vr::Eye_Left :
        vr::Eye_Right;
    vr::HmdMatrix34_t HeadFromEye = VRSystem->
        GetEyeToHeadTransform(HmdEye);

    // grab the eye position, currently ignoring the rotation
    // supplied by GetHeadFromEyePose()
    OutPosition = FVector(-HeadFromEye.m[2][3], HeadFromEye.m
        [0][3], HeadFromEye.m[1][3]) * Frame.WorldToMetersScale *
        (1.0/g_zoomFactor);
    FQuat Orientation(ToFMatrix(HeadFromEye));

    OutOrientation.X = -Orientation.Z;
    OutOrientation.Y = Orientation.X;
    OutOrientation.Z = Orientation.Y;
    OutOrientation.W = -Orientation.W;

    return true;
}
```

Code 8.1: SteamVR's eye pose calculation

OutPosition is scaled by  $\frac{1}{m} = \frac{1}{g\_zoomFactor}$ . Thereby normalizing stereo separation when zooming in ( $m > 1$ ), making objects appear at the same distance as they were without zooming in terms of binocular depth cues. Of course, they still appear larger.

In my user study (see Section 9.1.3), novices spent on average 77% and experts 60% of their whole training time in the optically magnified mode. In fact, most trials, 103 out of 198 trials (52%), were performed with nearly all of the time in zoomed mode (over 95% of the operating time), with 2x being the most common setting. This is a remarkably high number and shows that most users can not sufficiently see the tooth detail without magnification<sup>12</sup>. Two of 36 participants reported experiencing discomfort when using optical magnification, and explained that the head tracking becomes too sensitive when zoomed in. However, this can hardly be circumvented, as seeing a smaller region of the screen in higher detail automatically makes the rendering more sensitive to head rotation<sup>13</sup>. Of course, those participants could simply disable the zoom and proceed with the task.

<sup>12</sup>It would be interesting to investigate the visibility of small detail inside HMDs of varying resolutions.

<sup>13</sup>Imagine looking through binoculars on close targets (typically they are only used to see far away objects) – clearly, it would be more difficult to control where exactly you are looking at, compared to looking without binoculars.

### 8.1.3 Improved Eye Tracking for User Gaze Analysis

The HTC VIVE Pro Eye has built-in sensors that track the user's eyes at a frequency of 120 Hz, with an accuracy of 0.5–1.1°, by hardware specifications. I use the SRanipal software development kit (SDK) to communicate with the eye sensors, which works very well. This SDK also provides a UE plugin; however, I have found two notable shortcomings<sup>14</sup> of that plugin, which I explain in the following.

(i) **Low Frequency of Sensor Readings** The SRanipal UE plugin works synchronously with the rendering thread. Therefore, sensor updates are bound by the rendering performance of my application. In my case, I usually run between 60–90 Hz, as I am rendering a demanding VR scene with geometry that is constantly updated at run-time. This means I

<sup>14</sup>When inspecting the source code, there seems to be relatively little effort put into its development, compared to the Unity plugin, which has a complete transformation of gaze rays. My implementation was developed by considering the Unity plugin's source code, as it is more complete.



Figure 8.12: Eyetracking in dental simulator. Eye sensors inside the HMD track the pupil position which is used to compute the gaze direction. The gaze direction is transformed into the UE world and a raycast gives rise to what the user is looking at (displayed as a blue line). Intersections with the mirror are reflected, as that is what the user sees.

could only record eye tracking data at around 75 Hz, as opposed to the advertised 120 Hz. Leube et al. [Leube et al., 2017] have shown that there is a significant decrease in saccade detection when going from 120 Hz to 60 Hz. In addition, the net frequencies will be even lower, as some frames need to be rejected because of incorrect sensor readings.

(ii) **Gaze Origin Inaccuracy** The SRanipal UE plugin provides a function to compute a ray-cast from the user's cyclops eye (the midpoint between the left and right eye) into the virtual scene, to determine which virtual 3D point the user is focusing on. However, upon inspecting the source code for this function, I found it incorrectly assumes the user's cyclops eye at  $\vec{0}$  inside the camera's local frame during the ray-cast. Thus, the gaze origin is simply replaced by the camera position (the gaze direction is correctly transformed to the UE world). Obviously, a ray-cast from an incorrect origin will decrease the accuracy of the position the user fixates. This could be one of the reasons why several studies that examined the VIVE's eye-tracking accuracy found significant deviations from the advertised hardware accuracy:  $1.71^\circ$  by [Schuetz and Fiehler, 2022] and  $4.16^\circ$  by [Sipatchin et al., 2021].

To alleviate both issues, I have implemented the sensor communication in a C++ library that I access from UE. My C++ library is based on the SRanipal C++ SDK, but it allows for running the eye tracking in a separate thread that runs asynchronously to the game engine. Thus, I achieve the maximum tracking frequency of 120 Hz. Additionally, I corrected the incomplete gaze origin transformation from sensor space to UE world space. The gaze direction transformation needed to be adjusted to incorporate the changes to the stereo projection matrix described in Equation 8.5 and Equation 8.6.

Given a gaze origin  $\mathbf{o}_S$  and direction  $\vec{\mathbf{d}}_S$  that are defined in the HMD's local sensor frame,  $S$ , I compute the gaze origin and direction  $\mathbf{o}_W, \vec{\mathbf{d}}_W$  in the world frame by first transforming them into camera space using

$${}^C\mathbf{H}_S = S({}^W l_R) S\left(\frac{1}{m}\right) S((-1, 1, 1)) R_z\left(-\frac{\pi}{2}\right) R_x\left(\frac{\pi}{2}\right) S(0.001) \quad (8.7)$$

where  ${}^W l_R$  is the UE world-to-meter property, which I set to 50, and  $S$  is a scaling transform. This transformation is constant since the sensor does not move relative to the camera.

However, this transformation is only complete for the gaze origin  $\mathbf{o}^{15}$ . In the case the user has switched on the magnification (i.e., dental loupe), the gaze direction  $\vec{\mathbf{d}}_C$  needs to be rotated to adjust for the different positions that objects assume on screen (see Figure 8.13 as an example). I create a rotation  ${}^x\mathcal{R}_d$  that rotates the gaze direction  $\vec{\mathbf{d}}_C$  towards

<sup>15</sup>The gaze origin is unaffected by zooming, as zooming does not incur a change in the location of the cyclops-eye.

Figure 8.13: Optical magnification example. An exemplary screen capture that shows the same screen region at different magnification levels. When looking at the same screen position, the occupying object changes when the level of magnification changes.

Left No magnification (1x).

Center 2x magnification.

Right 4x magnification.



$\vec{x} = (1, 0, 0)^T$  (in UE, this is the forward direction in camera space). I then compute the corrective rotation through spherical interpolation by  $1 - \frac{1}{m}$  of the rotation towards the 0-rotation

$${}^d\Delta R_d = \text{slerp}\left({}^x\Delta R_d, \mathbf{0}, 1 - \frac{1}{m}\right) \quad (8.8)$$

$$\vec{d}_C = {}^d\Delta R_d \vec{d}_S \quad (8.9)$$

where the direction is defined as:  $\text{slerp}(\mathbf{A}, \mathbf{B}, 0) = \mathbf{A}$  and  $\text{slerp}(\mathbf{A}, \mathbf{B}, 1) = \mathbf{B}$ .

Finally I transform from camera space to world space by  ${}^W H_C$  (this matrix is dynamic and can be retrieved from the UE scene node). The total transformation then is

$$\mathbf{o}_W = {}^W H_C {}^C H_S \mathbf{o}_S \quad (8.10)$$

$$\vec{d}_W = {}^W H_C {}^d\Delta R_d {}^C H_S \vec{d}_S \quad (8.11)$$

### 8.1.3.1 Performance Metrics for Indirect Gaze

In order to analyze the gaze behavior of participants, I implemented a logging system that logs data of the user's gaze and the current simplified simulation state. The logs are then processed afterward to generate descriptive statistics on the behavior related to mirror placement, visual focus, etc.

I am, in particular, interested in two cases: (i) when the user is looking at the tooth or the bur, and (ii) when the user is inspecting the tooth. I will denote case (i) by the term *Vision*; I can detect this by casting the eye gaze ray into the scene and checking whether it hits the tooth or the bur geometry; in case it hits the mirror, I follow the reflected ray. This will also easily catch occlusions of the bur or the tooth by the handpiece, or incorrect placement of the mirror. Obviously, *Vision* is a desirable state for the full length of the procedure. Case (ii) is denoted by *Inspection* and can be detected by checking for a hit of the ray with the tooth while the bur is not removing material from the tooth.

For the analysis of users' gaze behavior, I implemented the detection of the events *Vision* and *Inspection*, which are marked by breaks in temporal coherence, as follows:

- **Vision break:** if there was continuously no *Vision* (of either bur or tooth) during the previous second. A break is finished, once a new gaze frame is present with *Vision* of tooth or bur. Obviously, a state of *Vision* is beneficial, and no *Vision* with the bur removing material is to be minimized. Therefore, I track vision breaks overall, and vision breaks during drilling.
- **Inspection:** this event is recorded as soon as there was no material removal for three seconds and the user's gaze hits the tooth surface. Inspection time is ended once material removal started again or there is no *Vision* of the tooth for one second. Therefore, dentists are generally not inspecting, and make use of *Inspection* occasionally to inspect their progress more closely.

I keep track of count, frequency, and average length of Vision breaks and Inspections, to be used in the analyses later. Additionally, I will also refer to the overall ratio of unobstructed indirect vision during drilling in relation to the whole time drilling under the term “correct mirror pose in drilling”. I have introduced these new metrics, as metrics presented in previous work are not suitable to this particular task. For example, expert classification based on gaze benefits from metrics suitable for machine learning [Hosp et al., 2021; Yin et al., 2020]. These metrics are intended for classification, and they are usually low-level, and thus, provide less valuable insights to humans. [Yang et al., 2022] used dwell time to analyze gaze behavior during dental check-ups. They divided the mouth into regions that act as areas of interest (AOI). To quantize the gaze behavior, dwell time on each AOI is used. This metric is less useful in my case, as we have only the tooth surface and possibly the bur tip as AOIs, which both are located at roughly the same location and very small in size. Therefore, vision is always concentrated at the same focus points, just the correct indirection through the mirror and avoidance of visual obstruction between mirror and AOI. I believe my metrics could be suitable for analysis of other surgical training experiments. For example, experiments on other surgical tasks, where avoiding the obstruction of vision is important, might utilize vision breaks. In tasks that are typical performed with regular inspections, inspections frequency and inspection break length might be used. The relevance of these metrics depend on the details of the other surgical tasks that the experiments are trying analyze.

My eye gaze recognition was overall quite reliable. The eye sensors provided valid eye-tracking data in 74.31% of all measurements. However, a few participants had lower sensor validity, primarily due to participants not wearing the HMD properly. I decided to exclude those participants from the analyses in order to minimize noise in the eye-tracking data.

In order to ensure the validity of the recordings of the eye tracking data, I conducted accuracy checks, at the beginning of each trial. This check consisted of presenting  $3 \times 3$  red dots (see Figure 8.14) in order while instructing the participants to fixate them. These red dots were located on a plane located 30 cm in front of the camera. They deviated from the central viewing direction by  $2.86\text{--}14.04^\circ$ , with  $8.85^\circ$  on average. The dot pattern was biased downwards by  $5.71^\circ$  to better sample the relevant viewing directions since dentists tend to look mostly downwards (relative to the central viewing direction) during the procedure. During the check, I recorded the median angle between the gaze ray and the ray towards the currently shown red dot. Each dot was shown for 3 s, of which I discarded the first second and the last 0.5 second to allow the participants enough time to change fixation targets. Thereby, I only regard accuracy data for 1.5 s after the dot was already visible for 1 s. Of all those measurements, I calculated the median value for each dot, and took the average of all dots’ median to represent the whole trial’s accuracy. In my study, participants achieved an accuracy of  $0.29\text{--}3.58^\circ$ , with on average  $1.21^\circ$ . In order to further minimize noise, I excluded trials if their accuracy error exceeded  $\epsilon_a = 1.27^\circ$ . I derived this threshold,  $\epsilon_a$ , by taking  $1/3$  of the diameter of the top surface of the tooth at a distance of 25.5 cm. The rationale for this is that I want gaze rays near the middle of the tooth surface, perpendicular to it, to be correctly recognized by the eye tracking. In practice, the recognition depends on many details such as the head position and view direction, mirror pose, handpiece pose and especially the optical magnification. For example, the eye tracking is much more accurate the higher the optical magnification is set to.

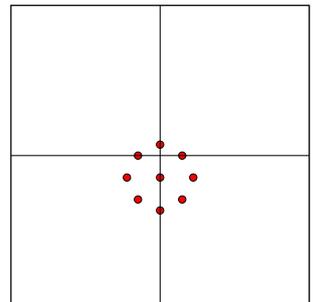


Figure 8.14: Accuracy test pattern. The accuracy test consists of 9 fixation targets that are distributed in the shown pattern. The pattern is biased downwards by  $5.71^\circ$  to be more in line with dentists typical viewing direction – they are thought to have good posture, which requires them to rotate the eyes downwards. Relative dimensions are mostly to scale.

## 8.2 LARGE FORCES FOR HIP SURGERY TRAINING SIMULATOR

I have participated in two externally financed research projects “HIPS” and “DynamicHIPS”, in which we have developed a simulator for practising performing THA on a virtual patient, with haptic feedback, which I developed. I have used the methods that I described in the previous chapters to implement the haptic feedback.



Figure 8.15: HIPS on KUKA robot. Showing me using the HIPS simulator, which is controlled via and renders haptic forces to the KUKA LBR iiwa 7 robot.



Figure 8.16: HIPS on Haption Virtuose. Showing someone using the HIPS simulator, on a Haption Virtuose.

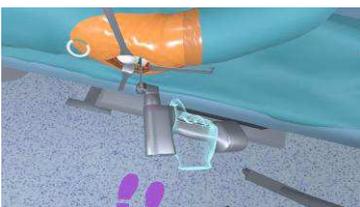


Figure 8.17: THA task #1. Sawing off the femoral head to allow access to the spongy bone.



Figure 8.18: THA task #2. Reaming of the pelvic bone to round shape of specific diameter.

The THA encompasses six surgical tasks, which the simulator can render haptically:

1. Cutting off the femur head using a bone saw (see [Figure 8.17](#))
2. Reaming the hip acetabulum to the implant diameter (see [Figure 8.18](#))
3. Inserting the hip implant by hammering it in
4. Rasping out the femur inside to implant diameter
5. Inserting the femur implant by hammering it in

where tasks 1 and 2 are rendered using traditional kinaesthetic (see [Chapter 6](#)) and vibrotactile force feedback, and tasks 3,4 and 5 are rendered using a novel hammering device (see [Section 8.2.1](#)). The kinaesthetic haptics work well on a host of haptic devices, predominantly the robots KUKA LBR iiwa 7 and 14 (see [Figure 8.15](#)), as well as commercial haptic devices by Haption, such as the Virtuose (see [Figure 8.16](#)). Haption devices can exert forces up to 70 N, and the KUKA robots support even higher forces of up to 137 N, which requires stable operation of the simulation in all possible situations. I have managed to run the simulator at a stiffness of up to  $12 \frac{\text{kN}}{\text{m}}$  on a Haption Virtuose prototype device, which surpasses high stiffnesses reported in literature, such as  $5 \frac{\text{kN}}{\text{m}}$  [[Desai et al., 2019](#); [Ostaduy and](#)

Lin, 2006] or theoretical  $10 \frac{\text{kN}}{\text{m}}$  [Wang et al., 2021]. It should be noted that this stiffness was only stable in the haptic device sweet-spot, and moving outside of this sweet-spot could occasionally result in undesirable vibrations.

During development, it proved very useful to have an additional safety mechanism put in place, as to (1) avoid damaging these devices and (2) to improve the user safety. The idea is to track the Haptic Tool's translational velocity  $\vec{v}(T_H)$  and angular velocity  $\vec{\omega}(T_H)$  and check if they exceed a set threshold  $\delta_t = 5 \frac{\text{m}}{\text{s}}$  and  $\delta_r = 2000 \frac{\circ}{\text{s}}$ . At that point, I essentially suspend the output of forces during haptic rendering for 3 s – the whole simulation continues running regularly, with the difference that calculated forces are zeroed before sending them to a device. When such an event occurred, and the suspension runs out, the sent force magnitude is gradually increased, such that a reactivation of the safety mechanism will reinitiate with the lowest velocity possible. The translational and angular velocities are calculated by looking through the history of the Haptic Tool pose. This procedure is similar to the way the velocity of all kinematic bodies is calculated (see Section 4.3.1), however, the passed time between the old pose and the current pose must be calculated in real time, not simulation time<sup>16</sup>.

### 8.2.1 Rendering of Extreme Forces during Surgical Hammering

The rest of the surgical tasks of the THA are performed using a steel hammer and steel impact tools. To reiterate these tasks encompass the following

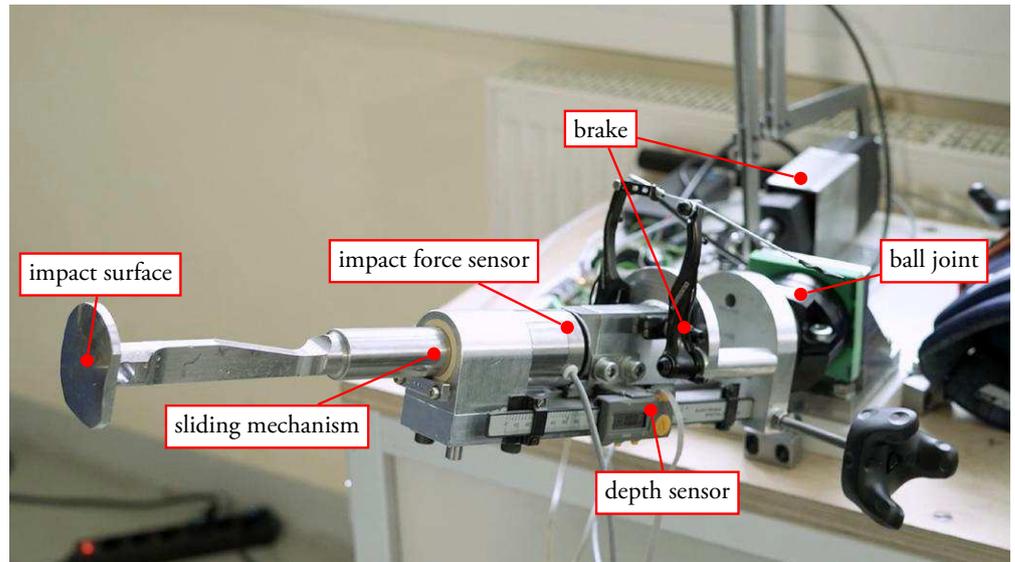
1. Inserting the hip implant by hammering it in the previously reamed out acetabulum.
2. Rasping out space for the femur implant inside the femur bone by repeatedly hammering rasps of increasing size inside and out of the femur cortical bone.
3. Inserting the femur implant by hammering it in the previously rasped out femur inside.

Rendering the forces that are generated during hammering of steel hammer onto steel objects is technically impossible on traditional kinaesthetic haptic devices. The reason for this is, that the forces can reach over 40 kN, whereas I have never heard of kinaesthetic haptic devices that can render more than around 140 N – most are far below, which is at least a difference of two to three orders of magnitude. Moreover, traditional haptic devices have a limited working range and some put limits on the motion path that can be realized. The first idea I had to implement such a scenario is to use the traditional approach, which is to mount the hammer onto the haptic arm and represent the impact tool as part of the virtual environment. However, this approach is not feasible due to the working range and motion restrictions. The second idea was to mount a secondary device onto the haptic device arm that receives the impact, meaning the virtual hammer is not rendered by the device. Instead, the haptic devices renders the passive impact tool, and the hammer is a physical dummy with dampened hammer head. However, this implementation would seriously risk damaging the haptic device, either because the arm experiences too large of an impact or because the user misses the secondary device and hits possibly sensitive parts of the haptic device. Therefore, implementing rendering of hammer impacts on kinaesthetic haptic devices is currently not possible, without overcoming these issues.

For my hip surgery simulator, the project partner from Technical University Chemnitz (TUC) has developed a new type of haptic device, called hammer device, which can render such high forces (see Figure 8.19). The idea is to have a steel impact surface, which can move uniaxially on a steel slide. This impact surface can withstand the high forces that occur during hammering, as it is made out of steel. The advancement of the impact surface on the slide is controlled via a bicycle brake mounted to the slide, which is actuated by

<sup>16</sup>If the safety threshold is checked against a velocity based on simulation time, a slowed down simulation would essentially raise the effective safety threshold, thereby negatively impacting safety. Similarly, a fast simulation would raise the effective safety threshold, possibly leading to false positive force suspension.

**Figure 8.19: Hammering device.** Photograph of the hammering device. The whole arm is mounted on a balljoint to a heavy table. The arm has a fixed slide with a brake mounted to it. On that slide, a sliding element is inserted that can fully slide on it. The impact surface is friction mounted, therefore, it can manually be inserted and removed by the user, in VR. The device has an impact force sensor to measure the force that the hammer impact, however, it did not work properly, and a depth sensor that can sense how far the sliding mechanism has progressed along the fixed slide.



<sup>17</sup>I have removed 1 femur rasping and 2 femur implanting experiments, as the data had anomalies that I could not work around.



**Figure 8.20: Biomedical experimental setup.** A donor acetabulum is mounted to a vise, the real implantation tool is hammered into it, while data is recorded.



**Figure 8.21: Biomedical marker setup.** The organ pose is tracked by three markers, and the implant by another marker.

an electric motor by mechanically pulling on the brake line. This advancement along the slide is tracked by a depth sensor and is reported via a USB controller to the host computer, which also runs the simulator. This combination of technology enables the simulation to dynamically change the braking force, which adjusts the amount of force needed to advance the impact surface along the slide.

I have designed an algorithm to dynamically adjust this braking force to render the haptical experience during the three hammering tasks during THA. The project partner TUC has conducted a set of biomechanical experiments related to these hammering tasks, during which they gathered several datasets that I will be working with in the following. More specifically, I have received 43 datasets of femur rasping, 21 of femur implanting and 16 hip implanting experiments<sup>17</sup>. For each experiment, the specific task was performed with the real tools on donor organs. The datasets are organized by tracking specific data, which is recorded after each hammer hit. The following data, dependent on the specific experimental task, was tracked:

- **Femur rasping**

1.  $f \in \mathbb{R}$ : the force magnitude of a hammer impact (in newton)
2.  $d_1 \in \mathbb{R}^3$ : the rasp marker location (in millimeters)
3.  $\theta \in \mathbb{R}^3$ : the orientation of the rasp marker (in degrees)<sup>18</sup>

- **Femur implanting**

1.  $f \in \mathbb{R}$ : the force magnitude of a hammer impact (in newton)
2.  $d_1 \in \mathbb{R}^3$ : the implant marker location (in millimeters)
3.  $\theta \in \mathbb{R}^3$ : the orientation of the implant marker (in degrees)

- **Hip implanting**

1.  $f \in \mathbb{R}$ : the force magnitude of a hammer impact (in newton)
2.  $d_i \in \mathbb{R}^3, i \in [1, 3]$ : three implant marker locations (in millimeters)
3.  $\theta_i \in \mathbb{R}^3, i \in [1, 3]$ : the orientations of implant markers (in degrees)

To design an algorithm for each task, I first want to unify these datasets. However, when analyzing the datasets for this, several issues arose:

1. Some data rows had missing data – seemingly due to the marker tracking malfunctioning.

2. The exact location of the reference points was not documented and might possibly differed between experiments.
3. The coordinate axes were often not congruent when comparing experiments, meaning one experiment's world  $x$ -axis could be another experiment's world  $y$ -axis (see Figure 8.22). For example, some hip implanting samples move in negative direction, some in positive. Additionally, the position also changes on axes.
4. The amount of hammer hits differed between experiments.
5. The positional differences vary greatly between samples, i.e. a rasp could be inserted between 3 mm or 25 mm.

<sup>18</sup>All orientations are relative to a marker attached to the organ.

To remedy issue 1, I excluded samples with incomplete data rows from further analysis, leading to the exclusion three data samples. The rest of the issues will be dealt with during the formulation of the material model. Firstly, I will create a new feed metric, which defines how far the impact tool has advanced due to hammer hits. I define the feed  $e$  as the average<sup>19</sup> positional difference to 0:

$$\vec{e} = \frac{1}{i} \sum_i \sqrt{x(\mathbf{d}_i)^2 + y(\mathbf{d}_i)^2 + z(\mathbf{d}_i)^2} \quad (8.12)$$

<sup>19</sup>I generally take the average of all available impact tool markers, if one could not be detected in an image it is discarded for that row.

Additionally, this change also solves both issues 2 and 3, as the axes labeling and relative marker locations are no longer relevant.

Next, I will analyze the overall position behavior and hit counts by normalizing the data. I do this in two steps: first, I normalize both variables independently to check to get absolute values of the respective other variable. Afterwards, I also normalize both variables at the same to possibly discover general trends in the functions. The independent normalization of the hit count (see Figure 8.24) revealed the usual insertion depths, depending on the task. For the femur rasping task, I observed a mean of 9.1 mm, and a median of 7.8 mm at completion. During the femur implant insertion, I observed a mean of 18.8 mm, and a median of 18.0 mm at completion. Finally, for the hip implant insertion, I observed a mean of 4.7 mm, and a median of 5.0 mm at completion. The independent normalization of the insertion depth (see Figure 8.24) revealed the common amount of hits per task. During the femur rasping task, I observed a mean of 5.8 hits, and a median of 5.5 hits until completion. For the femur implant insertion task, the mean was 6.7 hits, and a median of 5.0 hits to reach completion. The hip implant insertion exhibited a mean of 7.2 hits, and a median of 8.0 hits for completion of the task.

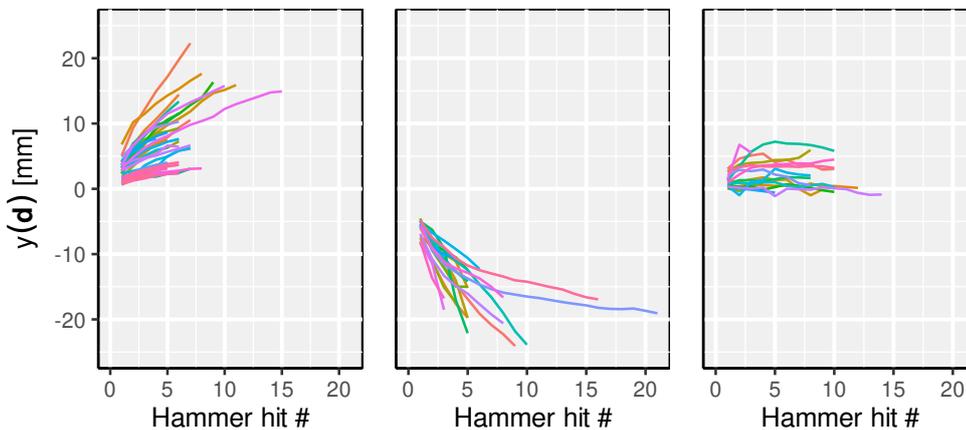


Figure 8.22: Biomedical hammering data, raw data. The raw data that TUC gathered for the hammering tasks during hip surgery. The  $y$ -axis shows the impact tool marker's  $y$ -position after the hammer hit. Each line represents a distinct dataset. Notice the many irregularities in the recorded data: hit count, initial position, end position and positional difference vary between samples.

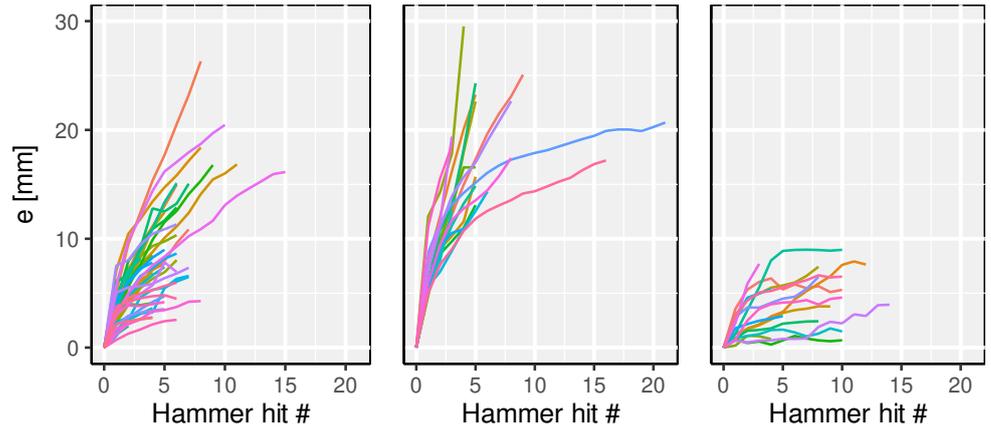
Left Rasping of femur trabecular.

Center Implanting in femur.

Right Implanting in pelvis.

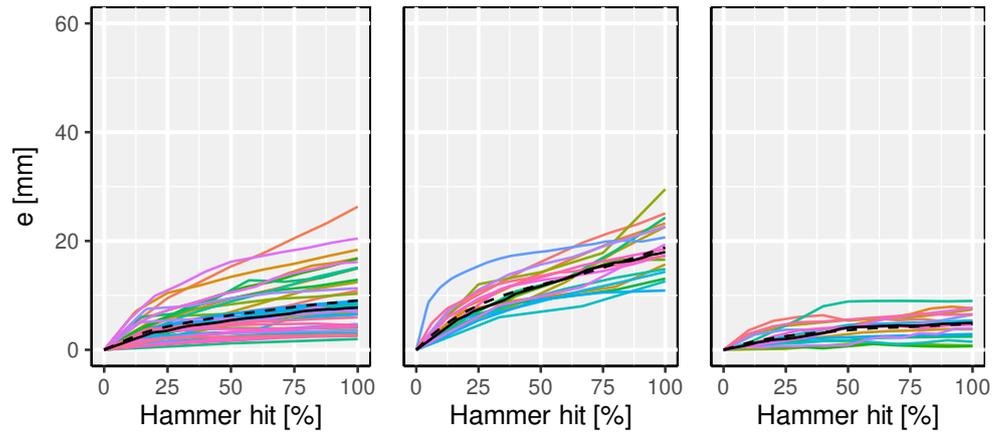
**Figure 8.23: Biomedical hammering data, custom feed metric.** The new feed metric  $e$  is used (see Equation (8.12) for a definition), which helps improve the dataset uniformity.

**Left** Rasping of femur trabecular.  
**Center** Implanting in femur – the inverted axis is corrected.  
**Right** Implanting in pelvis – the previously inconsistent trajectory is now more uniform.



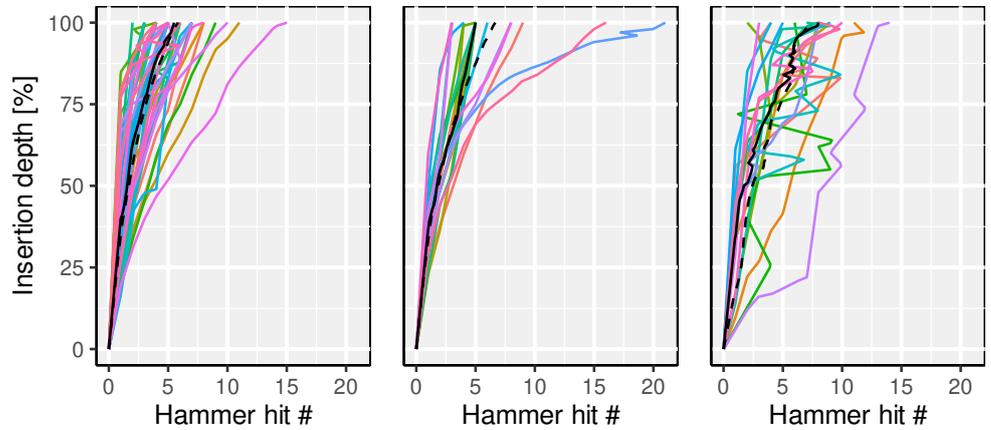
**Figure 8.24: Biomedical hammering data, normalized hit count.** The amount of hits is normalized to [0%, 100%]. This way, the full insertion depth can be calculated by averaging the samples at 100% hits. Black dashed line shows average; black solid line shows the median.

**Left** Rasping of femur trabecular, with a full insertion depth of 8 mm.  
**Center** Implanting in femur; the insertion depth is 10 mm.  
**Right** Implanting in pelvis; a full insertion depth of 20 mm.



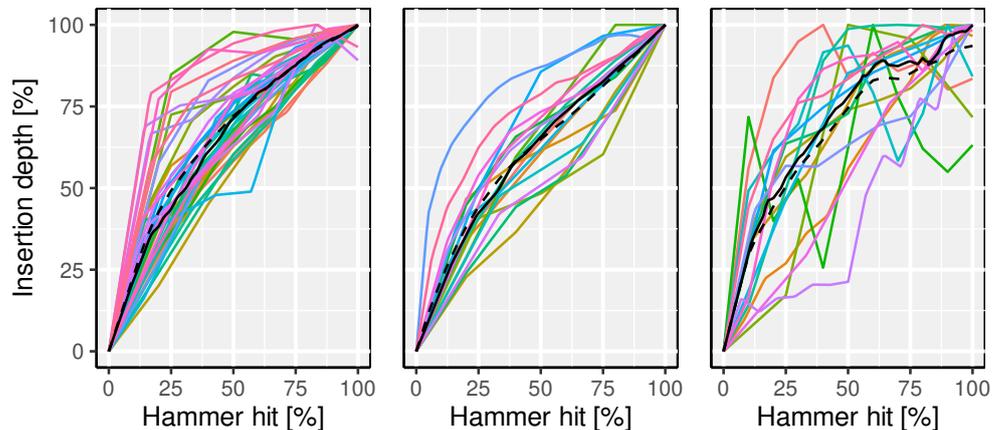
**Figure 8.25: Biomedical hammering data, normalized insertion depth.** The depth of insertion is normalized to [0%, 100%]. This way, the amount of hammer hits for completion can be calculated by averaging the samples at 100% insertion. Black dashed line shows average; black solid line shows the median.

**Left** Rasping of femur trabecular takes around 6 hits.  
**Center** Implanting in femur takes around 6 hits.  
**Right** Implanting in pelvis takes around 8 hits.



**Figure 8.26: Biomedical hammering data, fully normalized.** Both variables are normalized to [0%, 100%]. This shows the overall trends of the change in insertion depth over the task, quantized in hammer hits. One can clearly see that the overall trend is similar for all tasks, except for initial resistance, as signified by the steeper initial slope.

**Left** Rasping of femur trabecular shows a steep slope – the first few hits often do most insertion work.  
**Center** Implanting in femur shows a shallow slope – the insertion depth changes almost linearly  
**Right** Implanting in pelvis shows a steep slope – though the trend is less clear.



Finally, I normalized both variables (see Figure 8.26) to [0 %, 100 %]. This revealed the general shape of the feed dependent on the number of hammer hits. This shape can be simplified to

$$g(b) = \frac{c^b - 1}{c - 1} \quad (8.13)$$

where  $b \in [0, 1]$  is the hit count and  $c \in (0, 1)$  controls the initial slope. Equation (8.13) describes the insertion depth in relation to the hammer hit count, which is not something we can directly control. However, when taking the derivative of Equation (8.13),

$$\frac{d}{db}g(b) = \frac{c^b \log(c)}{c - 1} \quad (8.14)$$

we get the change in insertion depth, which could be better described as the feed rate, depending on the hit count. We can control the feed rate indirectly, by actuating the brake. This relation is generally inverse: to slow the feed rate, the brake needs to be actuated more strongly. Such a brake rate could be derived as such:

$$1 - \frac{d}{db}g(b) = 1 - \frac{c^b \log(c)}{c - 1} \quad (8.15)$$

In fact, when plotting this function, it shares many characteristics with Equation (8.16): a high initial slope, which is then monotonically decreasing and to be nearly flat, asymptotically approaching some value. However, some key differences remain, which are, that generally,  $1 - \frac{d}{db}g(0) \neq 0$  and  $1 - \frac{d}{db}g(1) \neq 1$ . Additionally, we are no longer considering the hit count, but use the insertion depth  $i$  read by the depth sensor instead. I do this, as it supports a more general model with emergent braking behavior, since a novice might as well use too little force, which would likely not cause the same effect as a sufficiently strong hit. I implemented this horizontal stretch by introducing the maximum insertion depth  $i_1$ , which for our device is limited to 52 cm, as the sliding mechanism does not reach further. These inconsistencies can be rectified by augmenting Equation (8.13) to

$$b(i) = \frac{c^{\frac{i}{i_1}} - 1}{c - 1} (b_1 - b_0) + b_0 \quad (8.16)$$

which now allows the control of other parameters, such as the initial braking force  $b_0$ , final braking force  $b_1$  and the final insertion depth  $i_1$ <sup>20</sup>. Thus, full parameterization of the braking function to meet the absolute value requirements discovered in Figure 8.25 and Figure 8.24 can be realized. This equation with the found initial values was used to drive the hammering device brake, for further tuning. Through repeated testing and tuning of the parameters with an expert, we have derived a set of parameters which were satisfactory to the expert's

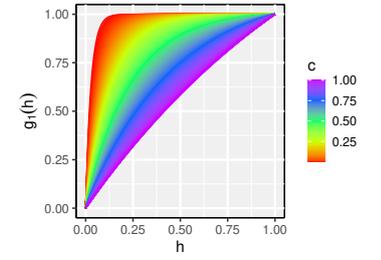


Figure 8.27: Feed rate during hammering. The general shape of the feed rate during hammering tasks (see Equation (8.13)), with  $b_{\text{end}} = 2$ . By varying the values for  $c$ , the general shape of feed rate (see Figure 8.26) can be modeled well.

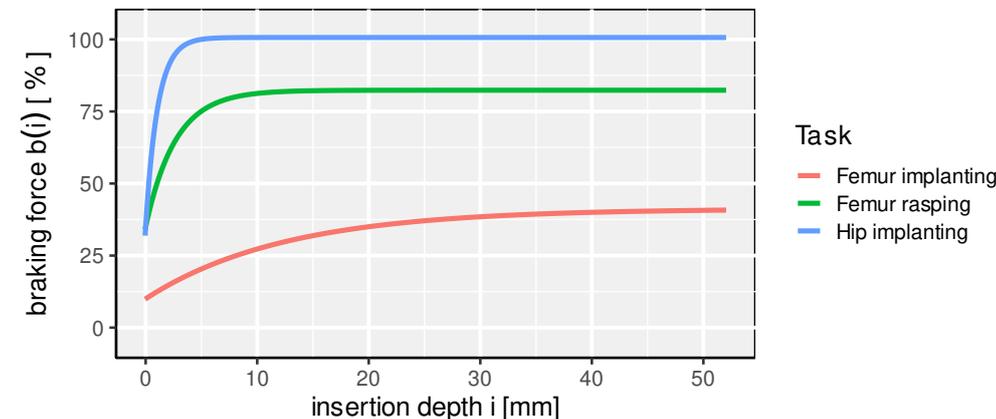


Figure 8.28: Hammering device braking model. The braking force for all tasks, dependent on the insertion depth, measured by the depth sensor of the hammering device. These parameters were found by repeated tuning with an expert in THA. The braking force is

haptic sensation, and lead to three distinct braking models (see [Figure 8.28](#)). For the femur rasping task, the parameter tuning lead to the following values:  $b_0 = 0.35$ ,  $b_1 = 0.8$ ,  $c = 0.05$  and  $i_1 = 8$ . Concerning the femur implant insertion task, the parameter tuning lead to the following values:  $b_0 = 0.1$ ,  $b_1 = 0.35$ ,  $c = 0.2$  and  $i_1 = 20$ . Finally, for the hip implant insertion task, the parameter tuning lead to the following values:  $b_0 = 0.35$ ,  $b_1 = 1.0$ ,  $c = 0.01$  and  $i_1 = 5$ .



## RELATED PUBLICATIONS

<b>P7.</b>	<b>A Shared Haptic Virtual Environment for Dental Surgical Skill Training . . .</b>	<b>212</b>
<b>P8.</b>	<b>The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training . . . . .</b>	<b>220</b>
<b>P9.</b>	<b>Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study . . . . .</b>	<b>228</b>
<b>P10.</b>	<b>The effect of 3D stereopsis and hand-tool alignment on learning effectiveness and skill transfer of a VR-based simulator for dental training . . . . .</b>	<b>246</b>
<b>P11.</b>	<b>Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks . . . . .</b>	<b>272</b>

## OUTLINE

9.1	Teaching Dental Surgery . . . . .	113
9.1.1	The Impact of HMDs in Virtual Simulators . . . . .	113
9.1.2	Guided Teaching by Remote Haptic Guidance . . . . .	115
9.1.3	Dental Mirror Handling and the Implications on Students' Skill . . . . .	117
9.2	Simulating Hip Surgery . . . . .	117
9.2.1	Ideal Rendering Methods for THA . . . . .	118
9.2.2	Experts' Impressions of the Simulator . . . . .	119

## EVALUATING SIMULATORS IN PRACTICE

“What you have spoke, it may be so perchance.”

— William Shakespeare

Simulators’ usefulness can not be demonstrated by their use of the latest technologies. Instead, their ultimate goal is to be an effective tool for helping students learn surgical procedures. My simulators’ effectiveness in this regard can only be demonstrated, with certainty, by systematic evaluation. In this chapter, I will provide an overview of my evaluation studies and their results. Additionally, digital simulators offer the unique ability to easily track and log complicated behavioral data of its users, which might reveal novel facts or connections related to the particular matter. I will lay out what novel findings I have discovered during these evaluation studies.

The chapter is split up by the surgical discipline (dentistry and orthopedics) and by associated evaluation study that I performed. Each discipline has, due to the different physical circumstances, different challenges and solutions, which I will highlight.

### 9.1 TEACHING DENTAL SURGERY

Dentistry encompasses many difficult tasks that require the use of complex bi-manual hand-eye coordination. Thus, dental students ought to acquire the necessary psycho-motor skills to master such difficult tasks during their training. However, the acquisition is particularly difficult and can only be achieved by repeated hands-on practice. Moreover, traditional simulators require the use (and their irrevocable wear) of expensive tools, such as anatomically correct plastic teeth and diamond burs. Thus, a conflict emerges which forces stakeholders to choose between cost saving and training success, as they are directly in competition. The use of VR technologies helps circumvent this issue, as they are reusable. This avoids the high operational costs and, thus, ultimately mitigates the conflict of interest. In cooperation with the Mahidol University and the Thammasat University, I conducted several user studies to test the effectiveness of my simulator in regards to skill acquisition. Additionally, I have looked at how unique features of VR, such as stereoscopy and realistic spatial relations influence the simulators’ effectiveness (see Section 9.1.1). In another evaluation study, I evaluated the effectiveness of remote haptic guidance for active teaching of caries removal (see Section 9.1.2). Lastly, I have conducted a study that tracked dental students tool use and eye gaze to evaluate their use of the dental mirror and how it relates to performance and learning effectiveness (see Section 9.1.3).

#### 9.1.1 The Impact of HMDs in Virtual Simulators

(based on [10], [8])

Due to the recent proliferation of virtual simulators in dental training, I wondered what the impact of the unique technological features of VR on simulators is. These features encompass *stereoscopy* and *realistic spatial relations*, both of which make up the unique selling points of HMDs.

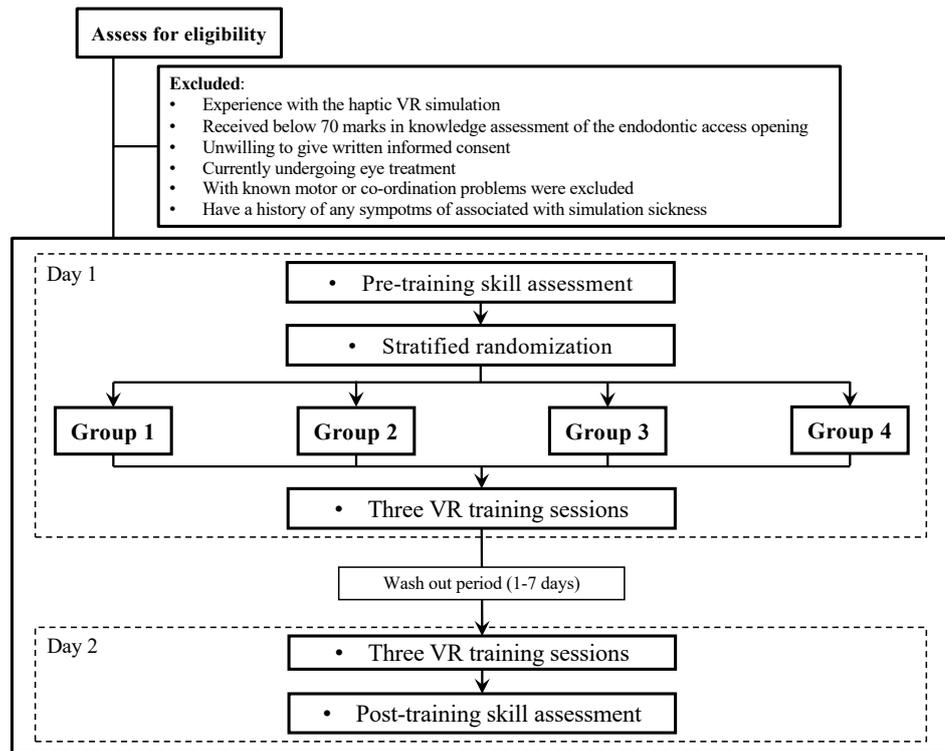
Therefore, I designed a user study in which I will systematically control for these variables. For this, I have firstly implemented a method to effectively disable stereoscopy inside VR in UE. The basic idea is to scale up the world, while also scaling up VR movement, such as from the head and controllers, such that movement feels natural, but actual scale is nearly infinitely large, resulting in a relative IPD that is nearly zero<sup>1</sup>. One condition involved normal stereoscopic 3D rendering, while the other used the nearly zero IPD, resulting in monoscopic 3D. Secondly, I have implemented one-to-one alignment of hands and tools,



**Figure 9.1: Registration error.** The dimensions of the registration error: the real devices are translated towards the user by 40 cm and downwards by 20 cm. This introduces an inverse registration error, effectively moving the virtual tools away and upwards from the user, in resemblance of desktop-based virtual simulators.

<sup>1</sup>UE does not allow the modification of the IPD used to render the VR image, which necessitated this workaround.

**Figure 9.2: HMD study flowchart.** Flowchart that shows the individual steps of the user study procedure. Participants were randomly assigned one of four groups, which determined their experimental conditions. Each participant underwent two training sessions, each encompassing three trials. Additionally, participants' skill was assessed pre- and post-training on real plastic teeth, using real tools. The outcomes were afterwards blindly scored by independent experts (in regards to who did them, and whether it was pre- or post-training) to have real-world skill measurements.

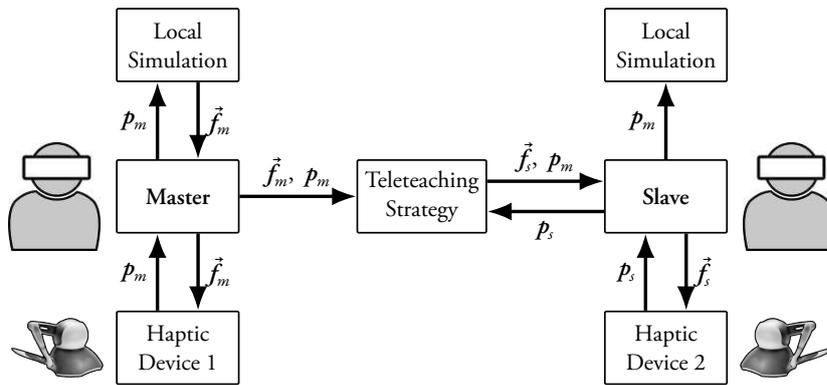


by the registration method described in [Section 6.2](#). In one condition, I have used the correct registration (called aligned condition); in another, I have introduced a large deliberate registration error, further called misaligned condition. The error dimensions were chosen, such that the misaligned condition best resembles the spatial relations of desktop-based virtual simulators, in which you see the scene at head height in front of you (see [Figure 9.1](#)). I constructed four groups based on a combination of these factors, which were given the following numbers:

1. Stereoscopic 3D, aligned hands and tools
2. Stereoscopic 3D, misaligned hands and tools
3. Monoscopic 3D, aligned hands and tools
4. Monoscopic 3D, misaligned hands and tools

Each participant was randomly assigned one these groups. In principle, they underwent the same experimental procedure (see [Figure 9.2](#)), except that the simulator was configured differently, according to the assigned group.

This study is the first to analyze the effect of different aspects of VR technologies on the transferability of dental skills from VR simulators training to real-world skill and vice versa. The data analysis revealed, that the alignment of physical and virtual tools has a positive impact on students' learning effect, compared to learning using misaligned physical and virtual tools. Additionally, alignment of physical and virtual tools is also associated with improved simulator usability, making it better suited to use such simulator setting for virtual assessment of skill. Surprisingly, I found that monoscopic 3D rendering resulted in larger learning gains, compared to stereoscopic 3D rendering. Thus, in my study, monoscopic 3D rendering provided students with more helpful training. However, further analysis suggests that this finding could be confounded by the eye-to-tooth distance, which was significantly lower for the monoscopic condition. On the other hand, my data suggests that stereoscopic 3D enforced a minimal distance at around 15 cm, which was rarely undershot. This was not the case for the monoscopic 3D condition, suggesting that the inter-ocular disparity of



**Figure 9.3: Guidance system.** Diagram to show the data flow and relevant actors in my guidance system. There are two asymmetrical actors that interact: the *master* and the *slave*. The master system runs the regular dental simulator (see Section 8.1) with regular force rendering. On the other hand, the slave system will render a guidance force, which depends on the used strategy.

stereoscopic 3D was the reason for the enforced minimal distance. Future studies focusing on stereoscopy should control for the user eye-to-tooth variable, e.g. by blurring the screen when coming too close to the AOI. Otherwise, the monoscopic condition can move the virtual camera closer, resulting in a higher resolution projection of the AOI. Still, stereoscopic 3D had a positive impact on the transfer of skill from real-world to the simulator, suggesting this condition is better suited for skill assessment using a simulator.

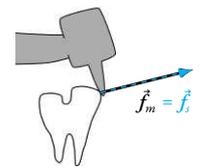
### 9.1.2 Guided Teaching by Remote Haptic Guidance

(based on [7])

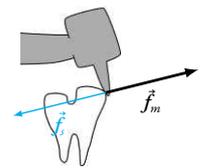
The high level and cost of resources required to provide clinical training in medicine and dentistry (e.g. over \$350,000 per completing dentistry student in Australia in 2016 [Segal et al., 2017]) necessitates concentration of clinical training programs in relatively few universities in any given country. In low- and middle-income countries, these universities are typically located in the main urban areas, requiring those outside the urban centers to travel there for training. This is a particularly problematic constraint for continuing education in advanced techniques where not only equipment but also expertise may be scarce. In recent years, online learning has become an effective approach to reach students who may not be able to travel to university campuses for various reasons. Its use has also dramatically increased during the recent COVID-19 pandemic with social distancing and lock-down requirements. However, online education has thus far been primarily limited to teaching of knowledge and cognitive skills. There is yet almost no use of online education for teaching of physical clinical skills. An advantage of VR simulators, which has not yet been explored for dental training, is the possibility of linking VR environments over a network so that a teacher and student may share a common simulator environment, in which they can teach these physical clinical skills.

In an effort to explore this solution, I have developed a shared haptic virtual environment for dental surgical skill training (see Figure 9.3). The system provides the teacher and student with a shared virtual environment containing a virtual dental station with patient, a dental drill controlled by a haptic device, and a virtual drillable tooth (see Section 8.1). The teacher can demonstrate a procedure while the student observes, and the student can then practice the procedure while the teacher observes. Since the kinaesthetic sense (the sense of force and motion) is essential in carrying out many dental procedures [Steinberg et al., 2007; Thomas et al., 2001], and surgical procedures in general [Skoczylas et al., 2012], an important question is how to best teach this in a virtual environment and how to make the haptic sensations “observable”. In order to explore how best to communicate kinaesthetic information, our system includes three modes for transmitting haptic sensations from the user performing the procedure to the user observing:

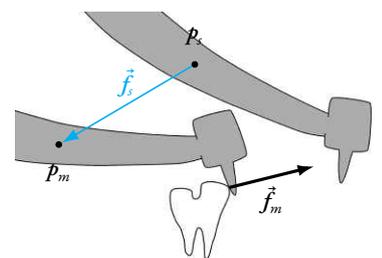
- **Same force:** transmit the force that the tooth is exerting on the master’s tool (see Figure 9.4).



**Figure 9.4: Same strategy.** Teleteaching force strategy *same*: the slave system renders the *environment force* that the master system computed to the haptic device of the slave system.

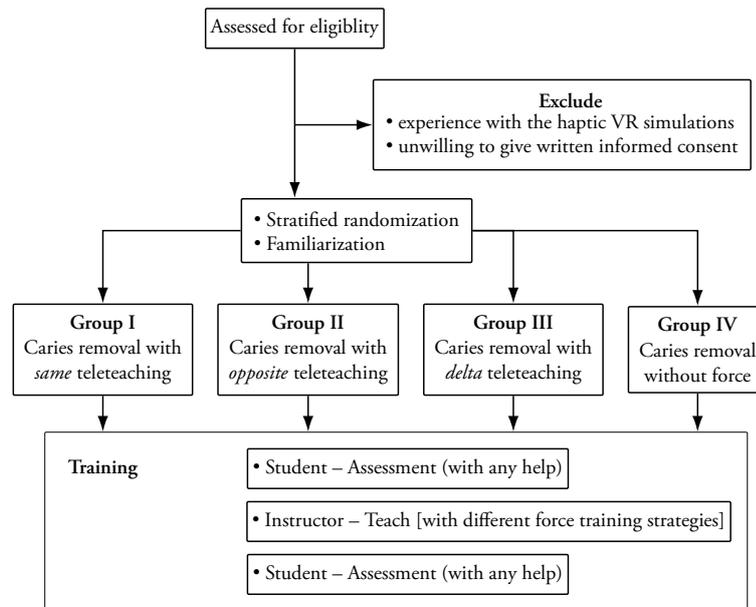


**Figure 9.5: Opposite strategy.** Teleteaching force strategy *opposite*: the slave system renders the inverted *environment force* that the master system computed to the haptic device of the slave system.

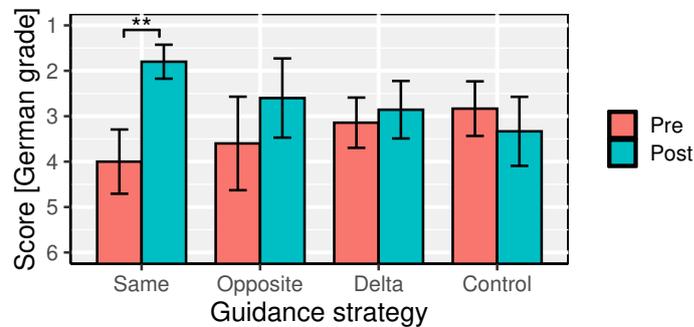


**Figure 9.6: Delta strategy.** Teleteaching force strategy *delta*: the difference of the tool poses is used to generate a spring force that will move the slave’s haptic device towards the master’s haptic device pose.

**Figure 9.7: Guidance study flowchart.** Flowchart that shows the individual steps of the user study procedure. Participants are randomly assigned one of four groups, which determined their experimental conditions. Each participant underwent a training session, in which a practising dentist controls the master system and performs the caries removal procedure, while the student receives guidance based on the group condition. Additionally, participants' skill was assessed pre- and post-training by letting them do the procedure inside the simulator themselves. The outcomes were recorded and blindly graded by the dentist.



**Figure 9.8: Guidance study result.** The performance improvement that resulted from the training. I have found a significant improvement in the performance score for the *same*-condition. All other studies had no statistically significant improvement, even though the *opposite*-condition showed a notable improvement. The score is a German grade in [1, 6], where 1 is the best and 6 the worst score.



- **Opposite force:** transmit the force that the master exerts on the tooth (see Figure 9.5).
- **Delta force:** transmit a force that will move the slave device along the same trajectory that the master device is moving along (see Figure 9.6).

The first two modes are designed to teach the amount of force to use, while the third is designed to teach the movement of the drill.

To investigate these strategies, I have designed a user study (see Figure 9.7) which compares their respective learning effectiveness during learning of caries removal. 23 participants without prior knowledge of dentistry were trained by being guided by an expert dentist with one of the three methods (with an additional group with only visual guidance, as a control). The participants skill was assessed before and after training by letting them perform the caries removal procedure without any guidance.

The analysis of the gathered data revealed that the same force, which renders the force that the tooth is exerting on the master's tool, on average produced the best caries removal outcomes (see Figure 9.8). For the *same*-condition, I found the post-training score ( $1.8 \pm 0.84^2$ ) to be significantly better than the pre-training score ( $4 \pm 1.58$ ), with statistical significance, as shown by a paired t-test ( $t(4) = -3.773, p < .01$ ). For the *opposite*-condition, I found the post-training score ( $2.6 \pm 1.95$ ) to be notably better than the pre-training score ( $3.6 \pm 2.3$ ), though not statistically significant ( $t(4) = -0.609, p = .29$ ). All other strategies resulted in little to no improvements during the training. Participants in the *delta*-condition improved slightly, going from  $3.14 \pm 1.46$  pre-training to  $2.86 \pm 1.68$ , without statistical significance ( $t(6) = -0.275, p = .40$ ). In the *control*-condition, participants even got slightly worse scores, going from  $2.83 \pm 1.47$  pre-training to  $3.33 \pm 1.86$  post-training, without statistical significance ( $t(5) = 0.425, p = 0.66$ ).

<sup>2</sup>Note that the grade here is the German school grade, where 1 is the best score and 6 is the worst score.

Thus, the conditions *same-force* and *opposite-force* were the most promising, with the *same-condition* resulting in a significant learning effect by just a single training session. Further testing with larger sample sizes is needed to verify whether the *opposite-force* strategy is an effective option to help students learn psycho-motor skills for dental procedures.

### 9.1.3 Dental Mirror Handling and the Implications on Students' Skill

(based on [11])

In the previous two studies, the task did not require the use of the dental mirror, although the simulator fully supported it. While I was conducting the experiments, I noticed that the majority of participants still did not utilize the dental mirror. Since this provided me with no data on how they would use the mirror, and what influence mirror use has on students' performance and possibly their learning. To circumvent this issue, I have designed a user study around a task which requires users to make use of the dental mirror. An appropriate task is the upper jaw root canal treatment (RCT), as teeth on the upper jaw can only be viewed through indirection using a mirror. Of course, it is not impossible to view upper jaw teeth directly, however, indirection is necessary to keep a healthy posture and good ergonomics, which dental educators impart as highly important. Additionally, I used my eye tracking (see Section 8.1.3) with gaze redirection along the mirror plane. I also added a zooming feature (see Section 8.1.2) for two reasons: (1) it is akin to the popular real-world optical magnification and (2) it increases the eye tracking accuracy.

The user study procedure (see Figure 9.10) encompasses two training sessions, with three RCAO simulator trials, meaning each trial consisted of doing the whole RCAO procedure to completion. Before and after training, I assessed their RCAO skill level, by performing the same procedure on a physical simulator (see Figure 9.9). The results revealed several novel findings regarding indirect vision. Most importantly, I have shown that indirect vision can predict users' performance and to some extent the personal learning effectiveness of the training. I came up with three novel indirect vision metrics:

1. **Indirect vision during drilling**, measured by the ratio of drilling with vision of important AOIs (tooth surface or bur tip): Moderately correlates with performance and learning effect.
2. **Frequency of detailed inspections<sup>3</sup>**: Weakly correlates with performance, but not with learning effect.
3. **Consistency of vision**, measured by prolonged breaks of vision: Strongly correlates with performance and weakly correlates with learning effect.

Some of which are to some extent specific to surgical procedures that involve redirection using a handheld mirror in narrow spaces and occlusion. I also discovered that students had significant learning effects in all of these novel metrics, meaning I could show that training using my simulator improved their proficiency of indirect vision. In most of these metrics, students could even reach proficiency levels which are comparable to experts.

## 9.2 SIMULATING HIP SURGERY

In hip surgery, the surgeon experiences forces which are multiple orders of magnitude larger<sup>4</sup>, which necessitated the integration specialized hardware and software solution. In this chapter I will present the specialized hardware and software solutions that I have developed to handle such large forces.

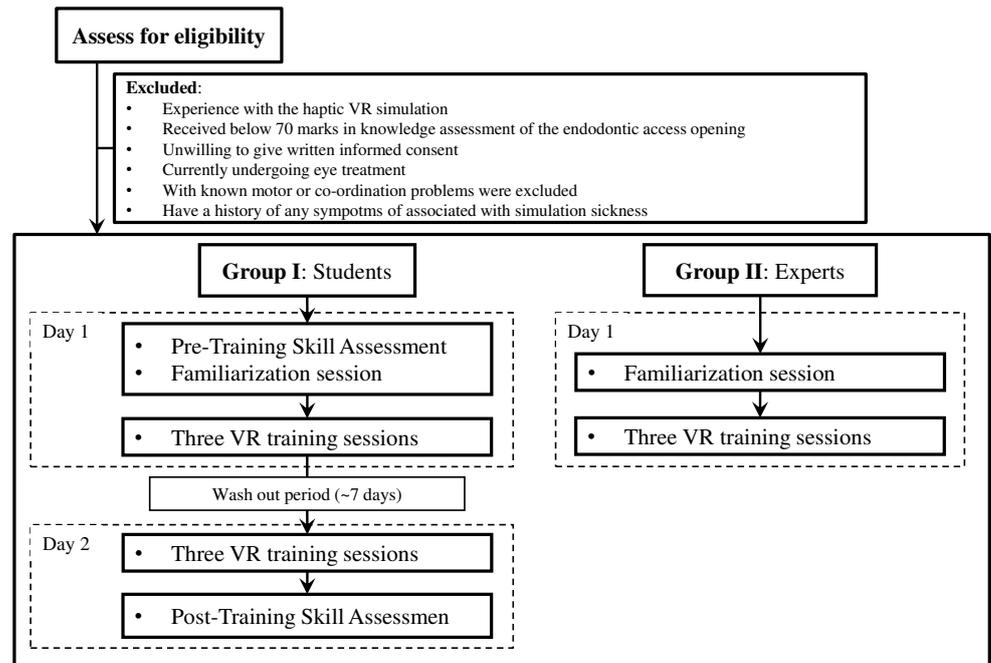


**Figure 9.9: Plastic tooth assessment.** A plastic tooth #36 is inserted into a dummy dentation. This tooth has correct internal anatomy and convincing physical properties, such that drilling them mimicks the real tooth materials. These teeth can be used once and cost roughly \$10. Additionally, a similarly expensive diamond bur is used to drill them, which also experiences wear.

<sup>3</sup>An inspection is characterized by no recent drilling activity, removal of the handpiece from the tooth area and vision of the tooth, typically focusing on the tooth inside, to inspect RCAO progress.

<sup>4</sup>Forces that occur in dental surgery generally do not exceed 3 N, whereas in THA, constant forces can reach a magnitude of 200 N and sporadic forces can reach magnitudes of up to 20 000 N.

**Figure 9.10: Mirror study flowchart.** Flowchart that shows the individual steps of the user study procedure. Participants were grouped based on expertise, where students were 5<sup>th</sup> year dental students and experts were fully trained dentists. The student group underwent two training sessions, each consisting of three trials, with a wash-out period of around one week between the sessions. To measure their performance, they underwent pre- and post-training assessment, which consisted of performing the procedure on real plastic teeth, which were later blindly rated by independent experts. The experts performed three trials, to provide a reference dataset to compare the students' data to.



### 9.2.1 Ideal Rendering Methods for THA

(based on [9], [4])

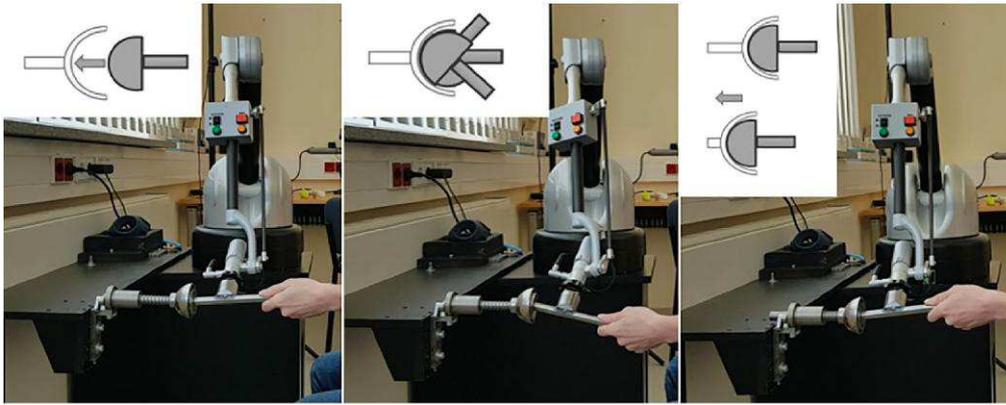
The research area of kinaesthetic haptic rendering has brought forth many force rendering schemes with different benefits and downsides. These can be very broadly categorized into direct rendering schemes, and proxy-based rendering schemes. Where the former are generally much simpler conceptually and programmatically. During my development of the VR THA simulator (see Section 8.2), I was challenged with developing realistic and safe haptic rendering of the different steps of the procedures (again, refer to Section 8.2). Ideally, I intended to choose an existing force rendering scheme that is ideally equipped to handle haptical rendering of simulation of THA. After a thorough review of the literature, there was no obvious candidate that is ideal for my use-case, even for the first two steps (sawing and reaming), during which the occurring forces are not as high as in the later three steps (implanting and rasping). To get a better idea of the suitability of the different force rendering schemes, I have designed a user study around the hip reaming task.

The hip reaming task essentially involves inserting the perfectly round reamer head into the hip acetabulum (see Figure 7.5 and Figure 7.14) and pushing with a relatively large force and possibly rotate in a specific manner. I wanted to analyze this surgical task. In fact, I have even made the study task more strict, to hopefully generalize the results to other surgical task that involve high forces, as this area is quite sparsely researched. I abstracted the hip reaming to use a perfect semi-sphere steel tool head and perfectly round steel cup. The abstracted task was then subdivided into the following subtasks (visualized in Figure 9.11):

1. Inserting the semi-sphere into the cup
2. Rotating the tool while the tool head is inserted into and thus bound to the cup
3. Pushing into the cup until a threshold force magnitude is reached<sup>5</sup>

<sup>5</sup>I chose a value near the haptic device maximum force magnitude of 60 N as the push completion threshold.

These task encompass all aspects that make hip reaming difficult to be haptically rendered. The insertion involves light force, which tests the stiffness of light contact and orientation with contact forces. The rotation task tests the haptic rendering of persistent contacts on a large lever and movement despite this persistent contact. Lastly, the pushing task tests the ability of the haptic rendering method to handle high forces. Using these tasks, I evaluated



**Figure 9.11: Abstracted THA task.** I abstracted the surgical THA task of hip reaming into these three tasks. All use a steel tool, which has a perfect semi-sphere at the end, instead of the hip reamer and a perfectly round steel cup instead of the hip acetabulum.

**Left** Inserting the tool into the cup with light force.

**Center** Rotating the tool while being in contact with the cup.

**Right** Pushing of the tool into contact with increasing force.

the suitability of haptic rendering approaches in a user study. The study procedure involved participants performing each task with each haptic rendering method (in a stratified random order) and rating their realism on a scale of 0–100% side-by-side to a rig where the same steel tool and cup were setup as physical objects. Thereby, I ensure that all participants have an accurate and immediate haptic sensation that they can compare against.

The results show that direct penalty-based rendering is overall the worst method for these tasks, and impulse-based rendering performed best for hard contacts. For the other tasks, rotation in contact and pushing during contact, the methods that were rated as producing the most realistic rendering are constraint-based and rigid-body-based force rendering. Thus, I conclude that a hybrid approach between impulse- and constraint-based or impulse- and rigid-body-based force rendering is the best general solution<sup>6</sup>.

### 9.2.2 Experts' Impressions of the Simulator

Based on the results of the previously mentioned evaluation study, I developed a training simulator for THA (see Section 8.2) using a hybrid haptic rendering approach. This simulator supports most steps of THA, of which most have not been evaluated in the previous study. Thus, I set out to evaluate the complete simulator by letting orthopedic surgeons try it out, to gather experts' feedback. This led to a demo setup inside a hospital, to make it as easy as possible to recruit as many surgeons as possible. Naturally, surgeons are bound by more time constraints, which necessitated this setup and only a single pass of the procedure inside the simulator.

Eighteen surgeons participated in the study (16 male, 2 female; mean age of 39 years) (see Table 9.1). The mean height of participants was 1.88 m. Participants reported to perform an average of 57 THA procedures per year. Nine participants were attending surgeons while the other nine were residents<sup>7</sup>. Attending surgeons had a mean of 10 years of experience compared to 3 years for residents. Five participants reported having prior VR experience, four reported having some VR experience, and nine had no prior VR experience. Nine participants had previously used VR for medical applications while the other nine had not.

The evaluation procedure firstly encompassed introducing the surgeons to the simulator, this was, to some extent, done on a per task basis, as the control scheme occasionally changes when switching between tasks (see Figure 9.12). For steps 1 and 2 (femur head sawing and acetabular reaming), a conventional kinaesthetic haptic device is used for control and to render moderate 6D forces ( $\sim 70$  N). On the other hand, for steps 3, 4 and 5 (hip implanting, femur rasping and femur implanting), the novel hammering device is used for control and to render the high impact forces (5–20 kN). The surgeons were briefly instructed what to do, and then proceeded to do the task to completion. Some participants wanted to redo a specific tasks, which we complied with, otherwise we moved on to the next

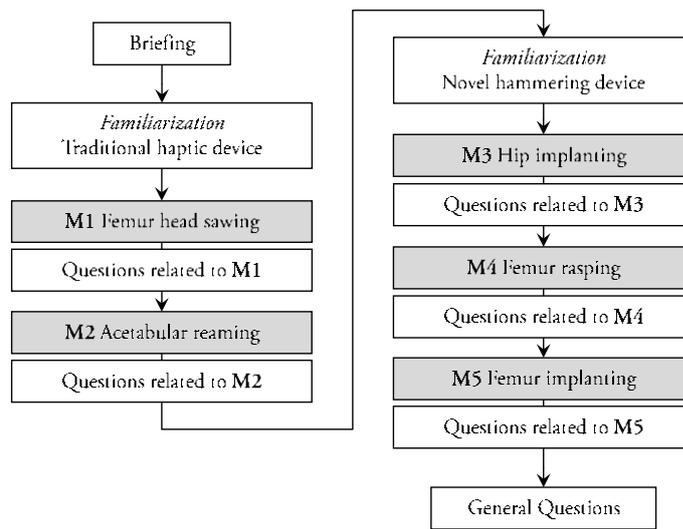
<sup>6</sup>I implemented such a hybrid approach in Section 5.3.3.

Demographic	Value
N	18
Sex	
Male	16
Female	2
Age	39 yr
Height	1.88 m
THA performed	57 $\frac{1}{\text{yr}}$
Attending Surgeon?	
No	9
Yes	9
Residents' experience	3 yr
Attending's experience	10 yr
VR experience	
Yes	5
Somewhat	4
No	9
Medical VR experience	
Yes	9
No	9

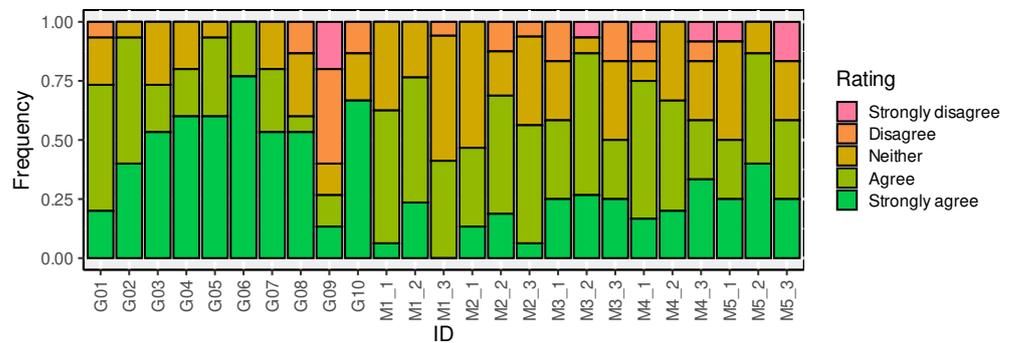
**Table 9.1: THA study demographics.** An overview over the demographics of the participants, which were a mix of residents and attending surgeons of a German hospital. The value column describes the mean answer, except for N it is a sum and if there were multiple choices, then it is the frequency of that answer.

<sup>7</sup>Residents are doctors employed by a hospital, which are, however, still under supervision and thus, only perform routine procedures. "Resident physician" roughly translates to the German qualification level of "Assistenzarzt". An attending surgeon on the other hand, is a doctor that has practical experience for 5 or more years and is allowed to treat patients, even for difficult procedures. "Attending surgeon" roughly translates to German qualification level of "Facharzt" or "Oberarzt".

**Figure 9.12: THA study design.** The surgeons were first briefed on how the simulator is used and what the scope of the simulated procedure is. Then, we started by familiarizing them with the kineasthetic haptic feedback device (Haption Virtuose), which is used for the first two modules. The same is done for the novel hammering device (see Figure 8.19) before module 3, as all following modules will be rendered using the hammering device. After the familiarization, we let them do the specific tasks inside the simulator and encouraged them to verbalize any immediate feedback they had, which we noted down. Additionally, we asked them a set of questions related to the module (the exact questions can be seen in Table 9.2, Table 9.3 and Table 9.4), before proceeding to the next module. In the end, we asked them another set of general questions (see Table 9.2).



**Figure 9.13: THA study feedback.** Statistical distribution of how much experts agreed or disagree with a given statement. The bar height visualizes the relative frequency at which the given answer was supplied, relative to the other answers, such that the sum comes up as 1. Statements are explained in Table 9.2, cross-referenced by the corresponding ID.



task. When transitioning from task 2 to 3, the physical location and equipment needed to be adjusted, which we manually helped with by guiding them. When all tasks were done, we asked each surgeon a set of questions. These questions were formulate together with project partners, and are based on a set of questions from Fang et al. [Fang et al., 2014], which aims at rating the acceptance of a technology:

- 10 questions concerning the quality and acceptance of the simulator as a whole (see Table 9.2). Answer distribution is shown in Figure 9.13.
- 11 questions each for modules 1 and 2 (femur sawing and acetabular reaming) concerning their simulation quality (see Table 9.2 and Table 9.4) and possible tuning of the related simulation parameters (see Table 9.3). Answer distributions are shown in Figure 9.13 and Figure 9.14.
- 3 questions each for modules 3,4 and 5 (hammering of hip implant, femur rasping and femur implant) concerning their simulation quality (see Table 9.2). Answer distribution is shown in Figure 9.14.

The analysis of the supplied answers revealed a set of findings, which I outline in the following. Experts accept the simulator as a whole, with respect to the majority of aspects, except for its relevance for the attending surgeons as a practicing tool, in which case a bit over 50% disagreed. All other general acceptance questions were agreed to with 60% or more. This shows, in particular, that experts in orthopedics collectively regard the simulator as a easy-to-use, realistic, useful learning tool for THA, for medical students and even residents. Therefore, any doctor that is still in training could benefit from training within my simulator, which demonstrates its efficacy as a training tool.

ID	Statement
G01	The 3D visualization of the situs was realistic.
G02	The 3D visualization of the OR was realistic.
G03	The HIPS simulator helps in learning the implantation of a hip prosthesis.
G04	Time was going by fast while I was using the HIPS simulator.
G05	Getting accommodated to using the HIPS simulator was easy.
G06	Using the HIPS simulator gave me great pleasure.
G07	I would recommend the HIPS simulator to medical students (last year of study).
G08	I would recommend the HIPS simulator to residents (in training at hospital).
G09	I would recommend the HIPS simulator to attending surgeons (completed training).
G10	The HIPS simulator should support preparation of patient-specific operations.
M1_1	The sawing of the femur head was realistic.
M1_2	Regarding the visualization: the sawing of the femur head looked realistic.
M1_3	Regarding the haptics: the sawing of the femur head felt realistic.
M2_1	The reaming of the acetabulum was realistic.
M2_2	Regarding the visualization: the reaming of the acetabulum looked realistic.
M2_3	Regarding the haptics: the reaming of the acetabulum felt realistic.
M3_1	The hammering of the hip implant was realistic.
M3_2	Regarding the visualization: the hammer of the hip implant looked realistic.
M3_3	Regarding the haptics: the hammer of the hip implant felt realistic.
M4_1	The hammering of the femur rasp was realistic.
M4_2	Regarding the visualization: The hammer of the femur rasp looked realistic.
M4_3	Regarding the haptics: the hammer of the femur rasp felt realistic.
M5_1	The hammering of the femur implant was realistic.
M5_2	Regarding the visualization: the hammer of the femur implant looked realistic.
M5_3	Regarding the haptics: the hammer of the femur implant felt realistic.

**Table 9.2: THA study questions.** A set of statements we designed to which the experts could disagree or agree to on a 5-point Likert scale, providing us with feedback on the quality of the simulator. It is a mix of statements concerning the usability, acceptance rate and realism (in terms of visuals and haptics) of the simulator. The statements starting with **Mx** are concerning the module **x**, meaning the numbered surgical step of the procedure. On the other hand, statements starting with **G** are concerning the general simulator impression, not specific to one step. The answer distribution is shown in Figure 9.13.

ID	Question	Feedback		
		Decrease	OK	Increase
M1_q1	How did you feel about the freedom of motion of the saw?	0	13	4
M1_q2	How did you feel about the friction amount of the saw?	4	11	2
M1_q3	How did you feel about cutting speed of the saw?	0	7	9
M1_q4	How did you feel about you influence of your pressure on the cutting speed of the saw?	0	4	10
M1_q6	Regarding your impression on the haptics of femur sawing: how was the cortical bone hardness?	2	11	4
M1_q7	Regarding your impression on the haptics of femur sawing: how was the spongy bone hardness?	2	13	2
M2_q1	How did you feel about the freedom of motion of the reamer?	0	13	0
M2_q2	How did you feel about the friction amount of the reamer?	1	9	2
M2_q3	How did you feel about cutting speed of the reamer?	2	11	2
M2_q4	How did you feel about you influence of your pressure on the cutting speed of the reamer?	1	12	1
M2_q6	Regarding your impression on the haptics of acetabular reaming: how was the cortical bone hardness?	0	8	5
M2_q7	Regarding your impression on the haptics of acetabular reaming: how was the spongy bone hardness?	0	9	2

**Table 9.3: THA study tuning directions.** A set of questions aimed at determining the fit and possible improvements of the simulation parameters. Here, I focus on kinesthetic haptic rendering and material removal, which are used in module 1 and 2 (femur sawing and acetabular reaming). The simulation's emergent behaviour is controlled by parameters, which I aim at optimizing through the experts' feedback. A statement's ID starts with **Mx**, indicating it is a question concerning the module **x**, meaning the numbered surgical step of the procedure (e.g. **M1\_1** is a question regarding module 1, which is the femur head sawing). The results are shown in Figure 9.14 (left).

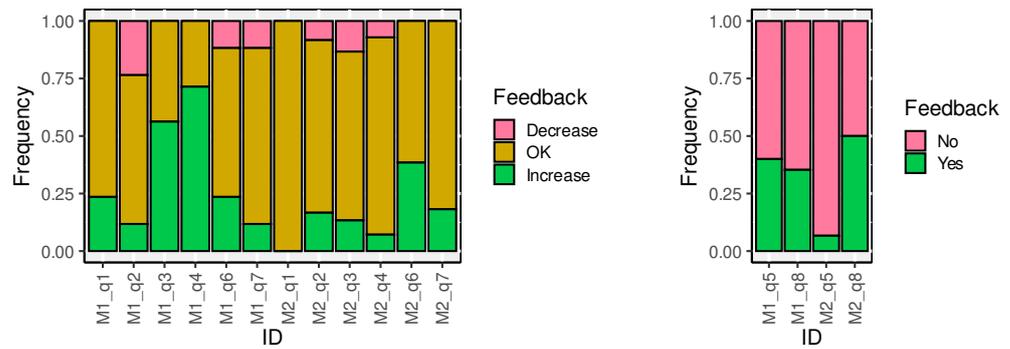
ID	Question	Feedback	
		Yes	No
M1_q5	Were you able to feel that the cortical bone was being cut?	6	9
M1_q8	The simulator does not support vibration: did this irritate you or was it missing?	6	11
M2_q5	Were you able to feel that the cortical bone was being cut?	1	14
M2_q8	The simulator does not support vibration: did this irritate you or was it missing?	7	7

**Table 9.4: THA study tuning.** A set of questions aimed at determining the quality and possible improvements of the kinesthetic haptic rendering and material removal, which are used in module 1 and 2 (femur sawing and acetabular reaming). A statement's ID starts with **Mx**, indicating it is a question concerning the module **x**, meaning the numbered surgical step of the procedure (e.g. **M1\_1** is a question regarding module 1, which is the femur head sawing). The results are shown in Figure 9.14 (right).

**Figure 9.14: THA study tuning results.** Statistical distribution of how experts answered a given statement. The bar height visualizes the relative frequency at which the given answer was supplied, relative to the other answers, such that the sum comes up as 1. Statements are explained in Table 9.3 and Table 9.4, cross-referenced by the corresponding ID.

**Left** The distribution of answers supplied for the parameter tuning questions (detailed in Table 9.3).

**Right** The distribution of answers supplied for the yes/no quality questions concerning modules 1 and 2 (detailed in Table 9.4).



The haptic rendering was rated as relatively realistic, where only about 20% or less disagree that the haptic rendering felt realistic. For the traditional kinaesthetic haptic feedback, the agreement is even higher, as only around 10% of participants disagreed (the earlier chapters explain the haptic rendering method used; see Chapter 4 and Chapter 6). Similarly, the visualization was rated as mostly realistic, for all modules, with a maximum of 20% disagreeing (see Section 7.6 for details on the algorithm).

Since I had not done any parameter tuning with experts, this was the first opportunity to get a well-sampled feedback on the parameter choice. Consequently, the questions in Table 9.3 and Table 9.4 were aimed getting feedback from the experts to improve the parameter choice. I have found that, most parameters were already in an acceptable range. However, there were also some notable problems:

1. The cutting speed of the saw was too slow, and the influence of the lateral force on cutting speed was rated as too low.
2. The experts could not tell by feeling, whether they were cutting cortical bone or spongy bone, which they ideally should

Both issues were since solved by adjusting the parameters of the cutting (as listed in Section 8.2) and modifying the material removal algorithm<sup>8</sup> (see Section 7.4). The manuscript already includes the final versions of all algorithms, including all of the modifications. In conclusion, the HIPS simulator was already regarded as helpful and realistic to experts in orthopedics, even at a developmental stage, without prior parameter tuning with an experts. With further parameter tuning the experts would likely rate it even higher in terms in realism and helpfulness.

<sup>8</sup>If the reader is wondering what was the problem for the cutting model: the issue is somewhat elaborate, but it boils down to not changing the cutting radius on a per-sphere basis. When deciding how much to cut from a sphere, based on its density, this will result in cutting spheres in soft layers earlier, even when there are spheres of the harder layer in front of it. This happens, since the harder spheres in front of softer spheres are cut much slower, due to their high density, however, the CCD makes the tool unable to advance. Therefore, whenever there is just one hard sphere within the contact set, the tool can not advance further, and thus, the cutting speed will feel too slow. I got better results by using the average density weighted by contact volume to compute a global cutting radius. When more spheres of a soft layer are in contact, all spheres will be cut more quickly. This might seem counter-intuitive, as the density contribution of a single sphere gets has a precise influence, however, the contact margin is very small, in the low millimeter range. Thus, this approach results in a gradual change in cutting speed as we change layers from hard to soft, and a single hard sphere will not hinder the tool advancement.

## 9.2. SIMULATING HIP SURGERY

OUTLINE

10.1 Future Work ..... 125

## CONCLUSION

---

**I**N this work, I have firstly layed out a modular system for a VR-based, haptic-enabled physics simulation. The modularity of my system stems from a structured view on the coordinate space transformations (see [Section 6.1](#)), that are necessary when so many individual parts are involved (virtual worlds, rendering backend, eye sensors, models, devices), each of which should be able to be easily exchanged for another. This enabled the integration of many different virtual tools, haptic devices (including impedance and admittance control), and virtual world, including completely different game-engines (Unreal Engine and Unity3D). This simulation supports continuous, realistic 6 DOF force responses (see [Section 5.3](#)) and material removal (see [Section 7.4](#)), while guaranteeing intersection-free movement (see [Section 5.1](#)). The volumetric models (see [Chapter 7](#)), which can be modified through material removal, can be visualized at interactive rates (see [Section 7.6](#)). These features, and many other features, elevate the state-of-the-art of virtual haptic VR simulators, and required the development of novel algorithms. I implemented these novel algorithms in a simulation library (see [Chapter 3](#)), and used it to develop two medical training simulators; one for THA (see [Section 8.2](#)) and one for dental surgeries (RCAO and caries removal (CR)) (see [Section 8.1](#)). The THA simulator supports the simulation of five individual surgical steps, with three completely different tools: bone saw, acetabular reamer (see [Section 7.1](#)), and hammer (see [Section 8.2.1](#)). For these simulators, I have implemented more novel features, such as automated VR registration (see [Section 6.2](#)), sound synthesis from material removal (see [Section 8.1.1](#)), VR zoom (see [Section 8.1.2](#)) and more accurate Unreal Engine eye tracking (see [Section 8.1.3](#)), and many more. I have received a lot of positive feedback and enthusiasm for my simulators from experts, including some feedback, that was formally documented (see [Section 9.2.2](#)).

I have shown through experiments, that training with my simulators has a significant learning effect on students, as evaluated by independent experts (see [Section 9.1.2](#)). In fact, more experiments have also shown, that this learning effect transfers from VR to real-world, which I measured by assessing the skill using traditional methods that use anatomical plastic teeth and real tools (see [Section 9.1.1](#) and [Section 9.1.3](#)). My dental simulator made it possible to discover the effect, that individual technological aspects of HMD VR have on the learning effectiveness and transferability of training simulators: hand-tool alignment (see [Section 6.2](#)) was shown to improve bi-directional transfer of skill, whereas stereopsis only improves transfer from real-world to VR (see [Section 9.1.1](#)). My simulators were also crucial instruments for analyzing dental students training behavior, which has revealed novel findings on how dental students use indirect vision, showing how indirect vision relates to performance and learning (see [Section 9.1.3](#)). This was possible by developing and measuring novel metrics on indirect vision, that correlate with performance: indirect vision during drilling, inspection frequency and consistency of vision. These novel insights were only possible to discover inside my dental simulator, as it enables natural and realistic training of RCAO, while tool trajectories and eye gaze can be accurately tracked.

### 10.1 FUTURE WORK

Through accurate tracking of tool trajectories, head movement and eye gaze, it is now possible to analyze aspects of medical students' training. Since I have already developed these systems, I want to conduct more experiments with medical students. For example, displaying indirect vision metrics after a trial on upper jaw teeth could help students get immediate feedback, based on which they can accelerate the learning process. It would also be possible

to generate a recording of trials, which could be played back to review the student interaction in detail with an expert. This would enable the expert to highlight concrete issues during the trial, which might improve students' learning effect even more, as knowledge of performance has been shown to be more important [Perry et al., 2015]. Another interesting possibility is, to implement a guidance system that will calculate the ideal mirror pose (based on the poses of mirror, target tooth, head and handpiece) and display it in a semi-transparent visual style, to help students move the mirror to maximize tooth visibility, given the current circumstances. The effect of these interventions on general skill learning and indirect vision performance could be evaluated in a user study.

There has been some interest in making one or more of my simulators into a product. To make a compelling product, several technological advancement would have to be achieved: (1) the scene setup would have to allow for multiple drillable organs, and (2) the limitation of having to use multiple GPUs would likely have to be removed. Multiple drillable organs is mostly a challenge for the collision detection (see Chapter 5) – a broad-phase would have to be implemented; and more so for the SDF generation (see Section 7.6), as the amount of required GPU memory would become too large for many GPUs – at least based on how it is currently programmed. The GPU memory currently allocated fields for signed distance, normals and color for each possible voxel; to reduce this, one would have to create a voxel pool that can dynamically grow as more voxels occupy the volume border, since many voxels in the grid do not contribute to the final mesh. The second GPU currently solves collision detection and material removal, as this device is not occupied by other tasks, such as rendering or dynamic mesh generation – the latter tasks are done on the primary GPU. If one GPU was to be removed, most of the collision detection and material removal would have to be computed on the CPU, as having too many tasks on the GPU creates a bottleneck. This occasionally leads to a slow simulation step, which noticeably impacts haptic rendering smoothness. Therefore, solving this issue well is a crucial future task to improve the portability of the simulation, while not negatively impacting the haptic sensation.

On the more novel technological front, an exciting new feature would be to add the possibility for deformation to the physics simulation. This would enable the simulation of deformable tools, such as dental files<sup>1</sup> or catheters, which are used to treat cardiovascular issues. Additionally, this could open up an avenue to also support deformation of materials, such as deformable skin, which would enable the two-way coupling of cuttable rigid and deformable organs, such as cuttable bone that is surrounded by deformable skin or muscle tissue. I have noticed that dentists like to push the inside of the cheek of a patient with the dental mirror to increase the space and thus vision inside the mouth; such interaction was not possible, but could be realized with skin deformation, coupled with rigid body contact. Further, this could lead to the exciting possibility to develop a method that also supports cutting of the deformable volume. Whether the intersection-free property of the simulation can be maintained under deformations, is unclear to me at this point. To the best of my knowledge, cuttable and deformable simulation has never been achieved before at haptic rates<sup>2</sup> and using a volumetric representation<sup>3</sup>. I hypothesize that such deformations could be achieved with the current volumetric representation (sphere packings), by introducing a graph over the spheres that represent the connectedness of neighboring spheres (I have already shown that such a graph is feasible for haptics, even when traversing large parts [Kaluschke et al., 2017]). The deformation will balance external forces that act upon a sphere with internal forces that are enforced by the interconnections between spheres. Cutting would then remove connections, when part of the blade collides with either a connection or neighboring spheres with sufficient normal force or sufficient lateral speed. Deformation of the visual mesh could be solved by defining vertex to sphere correspondence, which could be used for linear blending of vertex positions. A completely different approach would be to use the position-based dynamics method on the visual mesh (or a proxy mesh with fewer polygons); the downside here would be that intersections would likely be unavoidable, which could introduce instability into the simulation.

<sup>1</sup>Dental files are small flexible files that dentists use to enlarge and smooth out individual root canals after cutting the access opening.

<sup>2</sup>There are many works on integration of complex methods, such as finite element method (FEM) or material point method (MPM), into virtual simulators. The result is usual not fast enough for convincing haptics, but the quality of the simulation is very convincing.

<sup>3</sup>Surface representations have been successfully to haptically render deformable surfaces with convincing haptic sensation.

**Part III**

**Appendix**



## BIBLIOGRAPHY

- Agarwal, N., To, K., and Khan, W. (2021). Cost effectiveness analyses of total hip arthroplasty for hip osteoarthritis: A PRISMA systematic review. *International Journal of Clinical Practice*, 75(2):e13806.
- Ahmed, K., Keeling, A. N., Fakhry, M., Ashrafian, H., Aggarwal, R., Naughton, P. A., Darzi, A., Cheshire, N., Athanasiou, T., and Hamady, M. (2010). Role of Virtual Reality Simulation in Teaching and Assessing Technical Skills in Endovascular Intervention. *Journal of Vascular and Interventional Radiology*, 21(1):55–66.
- Aïm, F., Lonjon, G., Hannouche, D., and Nizard, R. (2016). Effectiveness of Virtual Reality Training in Orthopaedic Surgery. *Arthroscopy*, 32(1):224–232.
- Alaker, M., Wynn, G. R., and Arulampalam, T. (2016). Virtual reality training in laparoscopic surgery: A systematic review & meta-analysis. *International Journal of Surgery (London, England)*, 29:85–94.
- Altinöz, H. C., Gökbudak, R., Bayraktar, A., and Belli, S. (2001). A pilot study of measurement of the frequency of sounds emitted by high-speed dental air turbines. *Journal of Oral Science*, 43(3):189–192.
- Antoniadou, M., Tziouva, P., and Antoniadou, C. (2022). The Effect of Sound in the Dental Office: Practices and Recommendations for Quality Assurance—A Narrative Review. *Dentistry Journal*, 10(12):228.
- Arbabafti, M., Moghaddam, M., Nahvi, A., Mahvash, M., and Rahimi, A. (2008). Haptic and visual rendering of virtual bone surgery: A physically realistic voxel-based approach. In *2008 IEEE International Workshop on Haptic Audio Visual Environments and Games*, pages 30–35.
- Badash, I., Burtt, K., Solorzano, C. A., and Carey, J. N. (2016). Innovations in surgery simulation: A review of past, current and future techniques. *Annals of Translational Medicine*, 4(23):453.
- Bajo, J. M., Patow, G., and Delrieux, C. A. (2020). Realistic Buoyancy Model for Real-Time Applications. *Computer Graphics Forum*, 39(6):217–231.
- Baraff, D. (1994). Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '94*, pages 23–34, Not Known. ACM Press.
- Bender, J. (2007). Impulse-based dynamic simulation in linear time. *Computer Animation and Virtual Worlds*, 18(4-5):225–233.
- Berndt, I., Torchelsen, R., and Maciel, A. (2017). Efficient Surgical Cutting with Position-Based Dynamics. *IEEE Computer Graphics and Applications*, 38:24–31.
- Blinn, J. F. (1982). A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256.
- Bosker, B. H., Verheyen, C. C. P. M., Horstmann, W. G., and Tulp, N. J. A. (2007). Poor accuracy of freehand cup positioning during total hip arthroplasty. *Archives of Orthopaedic and Trauma Surgery*, 127(5):375–379.

- Buchanan, J. A. (2001). Use of simulation technology in dental education. *Journal of Dental Education*, 65(11):1225–1231.
- Ceylan, H. H., Vahedi, H., Azboy, I., Aali Rezaie, A., and Parvizi, J. (2020). Mini-Open Femoroacetabular Osteoplasty: Risk Factors for Failure and Conversion to Hip Arthroplasty. *JBJS*, 102(12):e59.
- Chami, G., Ward, J. W., Phillips, R., and Sherman, K. P. (2008). Haptic Feedback Can Provide an Objective Assessment of Arthroscopic Skills. *Clinical Orthopaedics and Related Research*, 466(4):963–968.
- Chan, S. (2014). *Haptic Rendering of Medical Image Data for Surgical Rehearsal*. PhD thesis, Stanford University, Stanford, CA, USA.
- Chan, S., Conti, F., Blevins, N. H., and Salisbury, K. (2011). Constraint-based six degree-of-freedom haptic rendering of volume-embedded isosurfaces. In *2011 IEEE World Haptics Conference*, pages 89–94, Istanbul. IEEE.
- Chan, S., Li, P., Locketz, G., Salisbury, K., and Blevins, N. H. (2016). High-fidelity haptic and visual rendering for patient-specific simulation of temporal bone surgery. *Computer Assisted Surgery*, 21(1):85–101.
- Chen, S.-C., Chueh, L.-H., Wu, H.-P., and Hsiao, C. K. (2008). Five-year follow-up study of tooth extraction after nonsurgical endodontic treatment in a large population in Taiwan. *Journal of the Formosan Medical Association = Taiwan Yi Zhi*, 107(9):686–692.
- Chu, F., Zheng, J., Wang, Q., Lu, X., Chen, Y., Zhong, Y., Li, Y., Shi, J., Jiang, Y., Zhang, W., Liu, L., and Sun, W. (2023). Mirror training device improves dental students' performance on virtual simulation dental training system. *BMC Medical Education*, 23(1):315.
- Colgate, J., Stanley, M., and Brown, J. (1995). Issues in the haptic display of tool use. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 140–145 vol.3.
- Das, A. K. J. (2020). Das-seng innomed instruments for anterior approach total hip arthroplasty. Orthopaedic Video Theater. [https://www.aaos.org/videos/video-detail-page/22972\\_\\_Videos](https://www.aaos.org/videos/video-detail-page/22972__Videos). Published December 15, 2020. Accessed on 2024-04-04.
- Desai, I., Gupta, A., and Chakraborty, D. (2019). Rendering Stiff Virtual Walls Using Model Matching Based Haptic Controller. *IEEE Transactions on Haptics*, 12(2):166–178.
- Dhaimy, S., Bedida, L., Merini, H. E., Benkiran, I., Dhaimy, S., Bedida, L., Merini, H. E., and Benkiran, I. (2019). External and Internal Root Canal Anatomy of the First and Second Permanent Maxillary Molars. In *Human Teeth - Key Skills and Clinical Illustrations*. IntechOpen.
- Dhingra, M., Goyal, T., Yadav, A., and Choudhury, A. K. (2021). One-year mortality rates and factors affecting mortality after surgery for fracture neck of femur in the elderly. *Journal of Mid-Life Health*, 12(4):276–280.
- Du, W., Zhang, J., and Hu, J. (2018). A Method to Determine Cortical Bone Thickness of Human Femur and Tibia Using Clinical CT Scans. page 11.
- Eve, E. J., Koo, S., Alshihri, A. A., Cormier, J., Kozhenikov, M., Donoff, R. B., and Karimbux, N. Y. (2014). Performance of dental students versus prosthodontics residents on a 3D immersive haptic simulator. *Journal of dental education*, 78(4):630–637.

- Faieghi, M., Atashzar, S. F., Tutunea-Fatan, O. R., and Eagleson, R. (2020). Parallel Haptic Rendering for Orthopedic Surgery Simulators. *IEEE Robotics and Automation Letters*, 5(4):6388–6395.
- Fang, Y., Qureshi, I., Sun, H., McCole, P., Ramsey, E., and Lim, K. (2014). Trust, Satisfaction, and Online Repurchase Intention: The Moderating Role of Perceived Effectiveness of E-Commerce Institutional Mechanisms. *Management Information Systems Quarterly*, 38(2):407–427.
- Freyberger, F., Kuschel, M., Färber, B., and Buss, M. (2007). A presence measure for virtual reality and telepresence based on multimodal conflicts. In *Proceedings of the ISPR Presence 2007. The 10th Annual International Workshop on Presence*.
- Ge Yu, Dangxiao Wang, Yuru Zhang, and Jing Xiao (2015). Simulating Sharp Geometric Features in Six Degrees-of-Freedom Haptic Rendering. *IEEE Transactions on Haptics*, 8(1):67–78.
- Giudice, J. S., Poulard, D., Nie, B., Wu, T., and Panzer, M. B. (2018). A Cortical Thickness Mapping Method for the Coxal Bone Using Morphing. *Frontiers in Bioengineering and Biotechnology*, 6.
- Gloy, K., Weyhe, P., Nerenz, E., Kaluschke, M., Uslar, V., Zachmann, G., and Weyhe, D. (2022). **Immersive Anatomy Atlas: Learning Factual Medical Knowledge in a Virtual Reality Environment.** *Anatomical Sciences Education*, 15(2):360–368.
- Go, C. C., Kyin, C., Maldonado, D. R., and Domb, B. G. (2020). Surgeon Experience in Hip Arthroscopy Affects Surgical Time, Complication Rate, and Reoperation Rate: A Systematic Review on the Learning Curve. *Arthroscopy: The Journal of Arthroscopic & Related Surgery*, 36(12):3092–3105.
- Golahmadi, A. K., Khan, D. Z., Mylonas, G. P., and Marcus, H. J. (2021). Tool-tissue forces in surgery: A systematic review. *Annals of Medicine and Surgery*, 65:102268.
- Gradl, R., Zhanette, I., Ruiz-Yaniz, M., Dierolf, M., Rack, A., Zaslansky, P., and Pfeiffer, F. (2016). Mass Density Measurement of Mineralized Tissue with Grating-Based X-Ray Phase Tomography. *PLoS ONE*, 11(12):e0167797.
- Guendelman, E., Bridson, R., and Fedkiw, R. (2003). Nonconvex Rigid Bodies with Stacking. page 8.
- Hosp, B., Yin, M. S., Haddawy, P., Watcharopas, R., Sa-ngasoongsong, P., and Kasneci, E. (2021). Differentiating surgeons' expertise solely by eye movement features. In *Companion Publication of the 2021 International Conference on Multimodal Interaction, ICMI '21 Companion*, page 371–375, New York, NY, USA. Association for Computing Machinery.
- Hou, X. and Sourina, O. (2012). Stable Dynamic Algorithm Based on Virtual Coupling for 6-DOF Haptic Rendering. In *2012 International Conference on Cyberworlds*, pages 157–164, Darmstadt, Germany. IEEE.
- Hulin, T., Sagardia, M., Artigas, J., Schaetzle, S., Kremer, P., and Preusche, C. (2008). Human-Scale Bimanual Haptic Interface. In *Proceedings, Pisa (Italien)*.
- James, T. P., Kelly, M. P., Lannin, T. B., Pearlman, J. J., and Saigal, A. (2013). Sagittal Bone Saw With Orbital Blade Motion for Improved Cutting Efficiency. *Journal of Medical Devices*, 7(1):011009.

- Kaluschke, M., Weller, R., Hammer, N., Pelliccia, L., Lorenz, M., and Zachmann, G. (2020). **Realistic Haptic Feedback for Material Removal in Medical Simulations**. In *2020 IEEE Haptics Symposium (HAPTICS)*, pages 920–926.
- Kaluschke, M., Weller, R., Yin, M. S., Hosp, B. W., Kulapichitr, F., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2024). **Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks**. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*.
- Kaluschke, M., Weller, R., and Zachmann, G. (2017). **A volumetric penetration measure for 6-DOF haptic rendering of streaming point clouds**. In *2017 IEEE World Haptics Conference (WHC)*, pages 511–516.
- Kaluschke, M., Yin, M. S., Haddawy, P., Srimaneekarn, N., Saikaew, P., and Zachmann, G. (2021). **A Shared Haptic Virtual Environment for Dental Surgical Skill Training**. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 347–352, Lisbon, Portugal. IEEE.
- Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2022). **The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training**. In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, pages 449–455, Rochester, MN, USA. IEEE.
- Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2023). **The Effect of 3D Stereopsis and Hand-Tool Alignment on Learning Effectiveness and Skill Transfer of a VR-based Simulator for Dental Training**. *PLOS ONE*, 18(10):e0291389.
- Kinoshita, H., Nagahata, M., Takano, N., Takemoto, S., Matsunaga, S., Abe, S., Yoshinari, M., and Kawada, E. (2016). Development of a Drilling Simulator for Dental Implant Surgery. *Journal of Dental Education*, 80(1):83–90.
- Koopman, P., Buis, J., Wesselink, P., and Vervoorn, M. (2010). Simodont, a haptic dental training simulator combined with courseware. *Bio-Algorithms and Med-Systems*, 6(11):117–122.
- Leube, A., Rifai, K., and Rifai, K. (2017). Sampling rate influences saccade detection in mobile eye tracking of a reading task. *Journal of Eye Movement Research*, 10(3):10.16910/jemr.10.3.3.
- Li, Y., Ye, H., Ye, F., Liu, Y., Lv, L., Zhang, P., Zhang, X., and Zhou, Y. (2021). The Current Situation and Future Prospects of Simulators in Dental Education. *Journal of Medical Internet Research*, 23(4):e23635.
- Lorenz, M., Hoffmann, A., Kaluschke, M., Ziadeh, T., Pillen, N., Kusserow, M., Perret, J., Knopp, S., Dettmann, A., Klimant, P., Zachmann, G., and Bullinger, A. C. (2023). **Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study**. *Scientific Reports*, 13(1):11230.
- McCormack, J. and Sherstyuk, A. (1998). Creating and Rendering Convolution Surfaces. *Computer Graphics Forum*, 17(2):113–120.
- McNeely, W. A., Puterbaugh, K. D., and Troy, J. J. (1999a). Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling. page 8.
- McNeely, W. A., Puterbaugh, K. D., and Troy, J. J. (1999b). Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, page 401–408, USA. ACM Press/Addison-Wesley Publishing Co.

- Mirtich, B. and Canny, J. (1995). Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics - SI3D '95*, pages 181–ff., Monterey, California, United States. ACM Press.
- Mordechai, F., Tali, C., Jonathan, M., Ori, P., Yaron, B., Ram, S., and Guy, T. (2022). The effect of type of specialty (periodontology/oral surgery) on early implant failure: A retrospective “Big-Data” study from a nation-wide dental chain in Israel. *Clinical Oral Investigations*, 26(10):6159–6163.
- Müller, M., Heidelberger, B., Hennix, M., and Ratcliff, J. (2007). Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118.
- Nauffal, D., Ballester, M., Reyes, R. L., Jiménez, D., Otero, R., Quintavalla, R., Monreal, M., and Investigators, T. R. (2012). Influence of recent immobilization and recent surgery on mortality in patients with pulmonary embolism. *Journal of Thrombosis and Haemostasis*, 10(9):1752–1760.
- Oeltze, S. and Preim, B. (2004). Visualisierung von Gefäßsystemen mit Convolution Surfaces. In Tolxdorff, T., Braun, J., Handels, H., Horsch, A., and Meinzer, H.-P., editors, *Bildverarbeitung für die Medizin 2004*, pages 189–193. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Ortega, M., Redon, S., and Coquillart, S. (2007). A Six Degree-of-Freedom God-Object Method for Haptic Display of Rigid Bodies with Surface Properties. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):458–469.
- Ortmaier, T., Weiss, H., Döbele, S., and Schreiber, U. (2006). Experiments on robot-assisted navigated drilling and milling of bones for pedicle screw placement. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2(4):350–363.
- Otaduy, M. and Lin, M. (2006). A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation. *IEEE Transactions on Robotics*, 22(4):751–762.
- Parker, M., Myles, J., Anand, J., and Drewett, R. (1992). Cost-benefit analysis of hip fracture treatment. *The Journal of Bone and Joint Surgery. British volume*, 74-B(2):261–264.
- Pelliccia, L., Lorenz, M., Heyde, C.-E., Kaluschke, M., Klimant, P., Knopp, S., Schleifenbaum, S., Rotsch, C., Weller, R., Werner, M., Zachmann, G., Zajonz, D., and Hammer, N. (2020). **A Cadaver-Based Biomechanical Model of Acetabulum Reaming for Surgical Virtual Reality Training Simulators.** *Scientific Reports*, 10(1):14545.
- Perry, S., Bridges, S. M., and Burrow, M. F. (2015). A Review of the Use of Simulation in Dental Education. *Simulation in Healthcare*, 10(1):31.
- Pohlenz, P., Gröbe, A., Petersik, A., von Sternberg, N., Pflessner, B., Pommert, A., Höhne, K.-H., Tiede, U., Springer, I., and Heiland, M. (2010). Virtual dental surgery as a new educational tool in dental school. *Journal of Cranio-Maxillofacial Surgery*, 38(8):560–564.
- Polce, E. M., Kunze, K. N., Williams, B. T., Krivicich, L. M., Maheshwer, B., Beletsky, A., Cohn, M. R., Kogan, M., and Chahla, J. (2020). Efficacy and Validity of Orthopaedic Simulators in Surgical Training: A Systematic Review and Meta-Analysis of Randomized Controlled Trials. *JAAOS - Journal of the American Academy of Orthopaedic Surgeons*, 28(24):1027.

- Sagardia, M. and Hulin, T. (2016). A fast and robust Six-DoF god object heuristic for haptic rendering of complex models with friction. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology - VRST '16*, pages 163–172, Munich, Germany. ACM Press.
- Satava, R. M. (2008). Historical Review of Surgical Simulation—A Personal Perspective. *World Journal of Surgery*, 32(2):262.
- Schuetz, I. and Fiehler, K. (2022). Eye Tracking in Virtual Reality: Vive Pro Eye Spatial Accuracy, Precision, and Calibration Reliability. *Journal of Eye Movement Research*, 15(3):10.16910/jemr.15.3.3.
- Segal, L., Marsh, C., and Heyes, R. (2017). The real cost of training health professionals in Australia: It costs as much to build a dietician workforce as a dental workforce. *Journal of Health Services Research & Policy*, 22(2):91–98.
- Sendyk, D. I., Chrcanovic, B. R., Albrektsson, T., Wennerberg, A., and Zindel Deboni, M. C. (2017). Does surgical experience influence implant survival rate? A systematic review and meta-analysis. *International Journal of Prosthodontics*, 30(4).
- Seymour, N. E. (2008). VR to OR: A review of the evidence that virtual reality simulation improves operating room performance. *World Journal of Surgery*, 32(2):182–188.
- Sherman, W. F., Flick, T. R., Dranoff, C. S., Weintraub, M. J., Kale, N. N., Sommi, C., and Sanchez, F. L. (2021). Variability of Cutting and Thermal Dynamics Between New and Used Acetabular Reamers During Total Hip Arthroplasty. *Arthroplasty Today*, 7:91–97.
- Sipatchin, A., Wahl, S., and Rifai, K. (2021). Eye-Tracking for Clinical Ophthalmology with Virtual Reality (VR): A Case Study of the HTC Vive Pro Eye’s Usability. *Healthcare*, 9(2):180.
- Skoczylas, L. C., Littleton, E. B., Kanter, S. L., and Sutkin, G. (2012). Teaching techniques in the operating room: The importance of perceptual motor teaching. *Academic Medicine: Journal of the Association of American Medical Colleges*, 87(3):364–371.
- Statistisches Bundesamt (2022). Current population of Germany. [https://www.destatis.de/EN/Themes/Society-Environment/Population/Current-Population/\\_node.html#sprg478526](https://www.destatis.de/EN/Themes/Society-Environment/Population/Current-Population/_node.html#sprg478526). Accessed on 2024-04-17.
- Statistisches Bundesamt (2023). Fallpauschalenbezogene Krankenhausstatistik (DRG). Operationen und Prozeduren an vollstationären Patienten: Deutschland, Jahre, Operationen und Prozeduren (1-4-Steller Hierarchie). <https://www-genesis.destatis.de/genesis//online?operation=table&code=23141-0101>. Accessed on 2024-04-04.
- Steinberg, A. D., Bashook, P. G., Drummond, J., Ashrafi, S., and Zefran, M. (2007). Assessment of faculty perception of content validity of PerioSim, a haptic-3D virtual reality dental training simulator. *Journal of Dental Education*, 71(12):1574–1582.
- The World Bank Group (2024). Life expectancy at birth, total (years). <https://data.worldbank.org>.
- Theiler, R. (2002). Arthrose - Epidemiologie, Diagnose und Differentialdiagnose, Abklärung und Dokumentation. In *Swiss Medical Forum - Schweizerisches Medizin-Forum*.
- Thomas, G., Johnson, L., Dow, S., and Stanford, C. (2001). The design and testing of a force feedback dental simulator. *Computer Methods and Programs in Biomedicine*, 64(1):53–64.

- Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, Bombay, India. Narosa Publishing House.
- Wan, M. and McNeely, W. (2003a). Quasi-static approach approximation for 6 degrees-of-freedom haptic rendering. In *IEEE Visualization, 2003. VIS 2003.*, pages 257–262.
- Wan, M. and McNeely, W. A. (2003b). Quasi-Static Approximation for 6 Degrees-of-Freedom Haptic Rendering. page 6.
- Wang, D., Zhang, X., Zhang, Y., and Xiao, J. (2013). Configuration-Based Optimization for Six Degree-of-Freedom Haptic Rendering for Fine Manipulation. *IEEE Transactions on Haptics*, 6(2):167–180.
- Wang, Q., Chen, H., Wu, W., Qin, J., and Heng, P. A. (Fourth 2012). Impulse-Based Rendering Methods for Haptic Simulation of Bone-Burring. *IEEE Transactions on Haptics*, 5(4):344–355.
- Wang, Q., Qin, J., Wang, W., Shan, J., Zhang, J., Liu, X., and Heng, P.-A. (2015). Haptic rendering of drilling process in orthopedic surgical simulation based on the volumetric object. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pages 1098–1101.
- Wang, Y., Feng, L., and Andersson, K. (2021). A Position-Control Based Approach to Haptic Rendering of Stiff Objects. *IEEE transactions on haptics*, 14(3):646–659.
- Weller, R. and Zachmann, G. (2009b). A unified approach for physically-based simulations and haptic rendering. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, pages 151–159, New Orleans Louisiana. ACM.
- Weller, R. and Zachmann, G. (2009-06-28, 2009a). Inner sphere trees for proximity and penetration queries. In *Robotics: Science and Systems V. Robotics: Science and Systems Foundation*.
- Weyhe, D., Uslar, V., Weyhe, F., Kaluschke, M., and Zachmann, G. (2018). **Immersive Anatomy Atlas — Empirical Study Investigating the Usability of a Virtual Reality Environment as a Learning Tool for Anatomy.** *Frontiers in surgery*, 5(73):73.
- Xu, H. and Barbic, J. (2014). Continuous Collision Detection Between Points and Signed Distance Fields. *Workshop on Virtual Reality Interaction and Physical Simulation*, page 7 pages.
- Xu, H. and Barbic, J. (2016). Adaptive 6-DoF Haptic Contact Stiffness Using the Gauss Map. *IEEE Transactions on Haptics*, 9(3):323–332.
- Yang, E., Park, S., Ryu, J., and Lim, T. (2022). How does dental students' expertise influence their clinical performance and perceived task load in a virtual dental lab? *Journal of Computing in Higher Education*, pages 1–27.
- Yin, M. S., Haddawy, P., Hosp, B., Sa-ngasoongsong, P., Tanprathumwong, T., Sayo, M., Yangyuenpradorn, S., and Supratak, A. (2020). A study of expert/novice perception in arthroscopic shoulder surgery. In *Proceedings of the 4th International Conference on Medical and Health Informatics, ICMHI '20*, page 71–77, New York, NY, USA. Association for Computing Machinery.
- Young, D. P., Stadler, S., Cutnell, J. D., and Johnson, K. W. (2018). *Cutnell & Johnson Physics*. John Wiley & Sons.

- Zhao, D., Li, Y., and Barbic, J. (2018). 6-DoF Haptic Rendering of Static Coulomb Friction Using Linear Programming. *IEEE Transactions on Haptics*, 11(3):325–337.
- Ziadeh, T., Perret, J., Kaluschke, M., Knopp, S., and Lorenz, M. (2020). **Review of Haptic Rendering Techniques for Hip Surgery Training.** In *EuroVR 2020 Application, Exhibition & Demo Track: Proceedings of the Virtual EuroVR Conference*, number 381 in VTT Technology, pages 29–39. VTT Technical Research Centre of Finland.
- Zilles, C. and Salisbury, J. (1995). A constraint-based god-object method for haptic display. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 146–151, Pittsburgh, PA, USA. IEEE Comput. Soc. Press.

## VARIATION OF CLUSTERING ALGORITHM

The following variation (see [Algorithm A.1](#)) on the presented clustering algorithm (see [Algorithm 5.8](#)) has a linear run-time of  $O(n) = n$ , with  $n$  being the number of present contact points, even in the worst case. The original algorithm (see [Algorithm 5.8](#)) can have a run-time complexity of  $O(n) = n^2$  in the worst case.

---

### Algorithm A.1 Linear Contact Clustering

---

**Input:**  $C$

**Result:**  $L$

**procedure** CONTACTCLUSTERINGLINEAR

**if**  $P_{\text{ClusterThreshold}} \leq 0$  **then**

$L \leftarrow C$

$\triangleright$  no clustering  $\Rightarrow$  a cluster for each contact with  $N = 1$

**return**

**for all**  $c_i \in C$  **do**

$d_i \leftarrow \infty$

$\mathbf{p} \leftarrow \mathbf{p}(c_i)$

$\tilde{\mathbf{n}} \leftarrow \tilde{\mathbf{n}}(c_i)$

$dop \leftarrow dop(c_i)$

$V \leftarrow V(c_i)$

$V_D \leftarrow V_D(c_i)$

    found\_cluster  $\leftarrow$  false

**for all**  $l_j \in L$  **do**

$d_{ij} \leftarrow |\mathbf{p} - \frac{\mathbf{p}(l_j)}{N(l_j)}|$

**if**  $d_{ij} < d_i$  **then**

$k \leftarrow j$

**if**  $d_{ij} < P_{\text{ClusterThreshold}}$  **then**

        found\_cluster  $\leftarrow$  true

$\mathbf{p}(l_j) \leftarrow \mathbf{p}(l_j) + \mathbf{p}$

$\tilde{\mathbf{n}}(l_j) \leftarrow \tilde{\mathbf{n}}(l_j) + \tilde{\mathbf{n}}$

$dop(l_j) \leftarrow \max(V(l_j), dop)$

$V(l_j) \leftarrow V(l_j) + V$

$V_D(l_j) \leftarrow V_D(l_j) + V_D$

$N(l_j) \leftarrow N(l_j) + 1$

**if not** found\_cluster **then**

**if**  $|L| < K$  **then**

$\triangleright$  no cluster near  $\Rightarrow$  create new cluster

$\mathbf{p}(l) \leftarrow \mathbf{p}$

$\tilde{\mathbf{n}}(l) \leftarrow \tilde{\mathbf{n}}$

$dop(l) \leftarrow dop$

$V(l) \leftarrow V$

$V_D(l) \leftarrow V_D$

$N(l) \leftarrow 1$

$L \leftarrow L \cup \{l\}$

**else if**  $d_i < \infty$  **then**

$\triangleright$  too many clusters  $\Rightarrow$  settle for nearest

$\mathbf{p}(l_k) \leftarrow \mathbf{p}(l_k) + \mathbf{p}$

$\tilde{\mathbf{n}}(l_k) \leftarrow \tilde{\mathbf{n}}(l_k) + \tilde{\mathbf{n}}$

$dop(l_k) \leftarrow \max(V(l_k), dop)$

$V(l_k) \leftarrow V(l_k) + V$

$V_D(l_k) \leftarrow V_D(l_k) + V_D$

$N(l_k) \leftarrow N(l_k) + 1$

**for all**  $l_j \in L$  **do**

$\triangleright$  average all properties except penetration depth

```
    |    $\hat{p}(l_j) \leftarrow \hat{p}(l_j)/N(l_j)$   
    |    $\hat{\mathbf{n}}(l_j) \leftarrow \hat{\mathbf{n}}(l_j)/N(l_j)$   
    |    $V(l_j) \leftarrow V(l_j)/N(l_j)$   
    |    $V_D(l_j) \leftarrow V_D(l_j)/N(l_j)$   
end procedure
```

---

## MATH NOTATION

---

### TYPES OF VALUES

Throughout this work, I typeset math variables with different value-ranges differently, to increase clarity and legibility. In the following (see Table B.1), I list the value categories and the typesetting convention used to display variable of each category.

Notation	Value type
$a, b \in \mathbb{R}$	Scalar
$p, c \in \mathbb{R}^3$	Point
$\Delta p, \Delta t$	Difference to previous state
$\dot{p}$	Derivative of point $p$
$\vec{v}, \vec{d} \in \mathbb{R}^3$	Vector
$q, R \in \mathbb{R}^4$	Quaternion
$M \in \mathbb{R}^{3 \times 3}$	Matrix
$H \in \mathbb{R}^{4 \times 4}$	Homogeneous matrix
${}_L p, {}_W \vec{v} \in \mathbb{R}^3$	A point or vector in a specific space $L$ or $W$ (this is generally omitted)

Table B.1: Value types. Typesetting conventions that I employ for various value types.

### COMMON OPERATIONS

I have also created short typesetting conventions for math operations that are frequently used; again, this is done to increase clarity and legibility. In the following (see Table B.2), I list all such operations and their respective typesetting convention.

Notation	Operation
$\frac{\vec{v}}{\ \vec{v}\ }, \hat{v}, \hat{d} \in \mathbb{R}^3$	Normalized vector
$x(\vec{v}), \vec{v}_x \in \mathbb{R}$	Components of vector
$M^{-1} \in \mathbb{R}^{3 \times 3}$	Inverse of matrix $M$
$M^T \in \mathbb{R}^{3 \times 3}$	Transpose of matrix $M$
$M_{1,2} \in \mathbb{R}$	Matrix element at row 1 and column 2
$M_{2,i} \in \mathbb{R}^3$	Matrix row 2
$M_{i,3} \in \mathbb{R}^3$	Matrix column 3
$T(t)^1$	Translation by $t$
$T(H)$	Extracted translation of matrix $H$
$S(s)$	Scale by $s$
$S(H)$	Extracted scale of matrix $H$
$R_x(\theta)$	Rotation of angle $\theta$ around axis $x$
$R(H)$	Extracted rotation of matrix $H$
$\theta(R)$	Extracting a torque vector from a rotation $R$ . This is a vector parallel to the rotation axis, with a magnitude proportional to rotation angle.

Table B.2: Common operations. Math typeset conventions that I follow for common math operations.

<sup>1</sup>I occasionally omit the set that the result lies in, in case there is ambiguity. E.g. when extracting the rotation from a matrix, the result could be a  $3 \times 3$  matrix,  $4 \times 4$  matrix, or a quaternion (so  $\mathbb{R}^4$ ).

CATEGORIZING TRANSFORMATION TYPES

Some chapters in this document heavily revolve around transformation matrices to convey more detailed information regarding the math and implementation. I have found that most transformations can be categorized into two kinds:

- **Change of basis:** represent a point in a different space (typeset as  ${}^B\mathbf{H}_A$ )
- **Delta transformation:** move a point from one pose to another pose (typeset as  ${}^B\Delta\mathbf{H}_A$ )

To make these transformations easier to distinguish and to create a better intuitive understanding of both types, I have constructed two new notations for both types. I will explain how they are constructed and provide motivation as to how each of them can be used, in the following.

CHANGE OF BASIS

**Definition**

I express a change of basis (a transformation matrix that transforms a point from one coordinate space  $A$  to another coordinate space  $B$ ) as

$${}^B\mathbf{H}_A = {}^W\mathbf{H}_B^{-1} {}^W\mathbf{H}_A$$

where  ${}^W\mathbf{H}_A$  is the local to world transformations that describes the pose of a body  $A$  inside a world  $W$ .

Given  $A$ 's position  $\mathbf{p}(A) \in \mathbb{R}^3$ , scale  $s(A) \in \mathbb{R}$  and orientation  $\mathbf{q}(A) \in \mathbb{R}^4$ , we construct  ${}^W\mathbf{H}_A$  in the following manner:

$${}^W\mathbf{H}_A = T(\mathbf{p}(A)) R(\mathbf{q}(A)) S(s(A))$$

Such a transformation can transform a local point (defined in space  $A$ ) to a global point (e.g. defined in  $W$ ). With the above definition, some point  $\mathbf{p}$  in an arbitrary space  $A$  can be transformed to a point  $\mathbf{p}'$  in any other arbitrary space  $B$  by

$${}^B\mathbf{p}' = {}^B\mathbf{H}_A {}^A\mathbf{p}$$

As such, these kinds of transformations do not move a point  $\mathbf{p}$ ; instead, its values are adjusted, to express the same location, relative to a different coordinate system  $B$ , which is represented by the new point  $\mathbf{p}'$ .

DELTA TRANSFORMATION

**Definition**

A transformation that actually moves a point  $\mathbf{p}$  from one pose  $A$  to another pose  $B$  inside a common coordinate space  $W$ , will be expressed as

$${}^B\Delta\mathbf{H}_A = {}^W\mathbf{H}_B {}^W\mathbf{H}_A^{-1}$$

This transformation can be used to generate a translation vector  $\vec{\mathbf{t}}$  and rotation axis  $\vec{\mathbf{a}}$  between the two pose origins<sup>2</sup>

$$\vec{\mathbf{t}} = T({}^B\Delta\mathbf{H}_A)$$

<sup>2</sup>If the body poses  ${}^W\mathbf{H}_A$  and  ${}^W\mathbf{H}_B$  are relative to the bodies' center of mass, the translation and rotation extracted from  ${}^B\Delta\mathbf{H}_A$  can be used to generate a force and torque that propels body  $A$  towards body  $B$  (this is used in Section 6.3).

$$\vec{a} = \mathbf{R}({}^B\Delta\mathbf{H}_A)$$

where the magnitude of  $\vec{a}$  is proportional to the difference in orientation.

A delta transformation can also be used to modify a local-to-world transformation matrix, e.g.

$${}^W\mathbf{H}_B = {}^B\Delta\mathbf{H}_A {}^W\mathbf{H}_A$$

which can be easily shown to be true by

$${}^W\mathbf{H}_B = {}^B\Delta\mathbf{H}_A {}^W\mathbf{H}_A = {}^W\mathbf{H}_B {}^W\mathbf{H}_A^{-1} {}^W\mathbf{H}_A = {}^W\mathbf{H}_B$$

which might seem not a useful property, since  ${}^W\mathbf{H}_B$  is necessary to construct  ${}^B\Delta\mathbf{H}_A$  in the first place. However, given this setup,  ${}^B\Delta\mathbf{H}_A$  can be modified, e.g. halving the translation, rotation and scale, to get the exact mid-way point of the two poses  ${}^W\mathbf{H}_A$  and  ${}^W\mathbf{H}_B$ , or any other in-between pose. Moreover, having a compact notation also improves legibility.

## COMMON SYMBOLS

Some symbols are recurring throughout this document, and I try to have consistent meaning assigned to a certain symbol. In the following (see [Table B.3](#)), I provide a small selection of such recurring symbols; however, this list is not exhaustive.

Symbol	Meaning
$\rho \in \mathbb{R}$	Density
$k \in \mathbb{R}$	Stiffness
$V \in \mathbb{R}$	Volume
$r \in \mathbb{R}$	Radius
$\vec{p} \in \mathbb{R}^3$	Position
$\vec{v} \in \mathbb{R}^3$	Translational velocity
$\vec{\omega} \in \mathbb{R}^3$	Angular velocity
$\vec{n} \in \mathbb{R}^3$	Normal (the direction perpendicular to a plane that locally describes a surface geometry)
$\vec{r} \in \mathbb{R}^3$	Lever
$\vec{f} \in \mathbb{R}^3$	Force vector
$\mathbf{I}_B \in \mathbb{R}^{3 \times 3}$	Inertia tensor (in body frame)
$\mathbf{I} \in \mathbb{R}^{3 \times 3}$	Inertia tensor (in world frame)
$\vec{\tau} \in \mathbb{R}^3$	Torque vector
$\theta \in \mathbb{R}$	Angle of a rotation
$P_{\text{name}}$	Global simulation parameter <sup>3</sup>
$c_i$	Contact point $i$
$s_i$	Sphere at index $i$ in a sphere packing

Table B.3: Common symbols. Showing a list of symbols which, generally speaking, have a specific recurring meaning. This list is not exhaustive.

<sup>3</sup>I implemented global parameters to be any value type, though they are mostly in  $\mathbb{R}^x$ ,  $1 \leq x \leq 3$ .

## LIST OF FIGURES

---

1.1	Orthopedic procedures over time . . . . .	3
1.2	Procedure frequencies, grouped by kind of surgery . . . . .	4
1.3	Top 10 most common surgical procedures . . . . .	4
3.1	System algorithmic overview . . . . .	17
3.2	System class diagram . . . . .	20
4.1	Integration of rigid-body simulation . . . . .	25
5.1	The role of Collision Detection. . . . .	31
5.2	Contact clustering . . . . .	40
5.4	Contact resolution methods . . . . .	43
5.3	Reamer in contact . . . . .	43
5.5	Translational dop motivation . . . . .	45
5.6	Interpenetrating spherepackings . . . . .	45
5.7	Depth-of-penetration measures . . . . .	45
5.8	Computation of adaptive stiffness . . . . .	48
6.1	Integration of Haptic Rendering . . . . .	53
6.2	Coordinate spaces to integrate a haptic device in a game-engine . . . . .	54
6.3	Haptic device setup . . . . .	58
6.4	VR Registration . . . . .	59
6.5	Small, complex tool features are also rendered . . . . .	65
6.6	Smooth sliding . . . . .	65
6.7	Arbitrary contacts . . . . .	65
6.8	Reaming contact . . . . .	65
7.1	Integration of material removal & visuals . . . . .	67
7.2	Passive contacts in root canal . . . . .	68
7.3	Passive contacts in caries removal . . . . .	68
7.4	Passive contacts in femur head sawing . . . . .	68
7.5	Passive contacts during acetabular reaming . . . . .	68
7.6	Acetabular reamer . . . . .	69
7.7	Femur bone saw . . . . .	69
7.8	Conical drill directions . . . . .	70
7.9	Human bone model . . . . .	71
7.11	Tooth anatomy . . . . .	72
7.12	Anatomical tooth model . . . . .	72
7.10	Unique teeth in dentition . . . . .	72
7.13	Dynamic features . . . . .	73
7.14	Material removal sketch . . . . .	75
7.15	Capsule material removal sketch . . . . .	79
7.16	Dynamic mesh pipeline . . . . .	81
7.17	Visualization of sphere packing . . . . .	83
7.18	SDF offsets. . . . .	83
7.19	Effect of SDF offset . . . . .	84
7.20	Metaball visualization . . . . .	85
7.21	Effect of distance smoothing . . . . .	86
7.22	Effect of overly optimized smoothing . . . . .	86

7.24	Color smoothing . . . . .	87
7.23	Effect of normal smoothing . . . . .	87
7.25	Femur with unified parameters . . . . .	88
7.26	Pelvis with unified parameters . . . . .	88
8.1	Carious tooth . . . . .	93
8.2	Root-canal procedure . . . . .	93
8.3	Initial dental simulator patient . . . . .	93
8.4	Initial dental VR simulator . . . . .	94
8.5	Initial dental simulator registration . . . . .	94
8.6	VR Simulator . . . . .	94
8.7	Sound sample frequency . . . . .	96
8.8	Volume ramp up . . . . .	96
8.9	Spinning sound synthesis blueprint . . . . .	97
8.10	Simulated sounds . . . . .	97
8.11	Drilling sound synthesis blueprint . . . . .	98
8.12	Eyetracking in dental simulator . . . . .	101
8.13	Optical magnification example . . . . .	102
8.14	Accuracy test pattern . . . . .	103
8.15	HIPS on KUKA robot . . . . .	104
8.16	HIPS on Haption Virtuose . . . . .	104
8.17	THA task #1 . . . . .	104
8.18	THA task #2 . . . . .	104
8.19	Hammering device . . . . .	106
8.20	Biomedical experimental setup . . . . .	106
8.21	Biomedical marker setup . . . . .	106
8.22	Biomedical hammering data, raw data . . . . .	107
8.23	Biomedical hammering data, custom feed metric . . . . .	108
8.24	Biomedical hammering data, normalized hit count . . . . .	108
8.25	Biomedical hammering data, normalized insertion depth . . . . .	108
8.26	Biomedical hammering data, fully normalized . . . . .	108
8.27	Feed rate during hammering . . . . .	109
8.28	Hammering device braking model . . . . .	109
9.1	Registration error . . . . .	113
9.2	HMD study flowchart . . . . .	114
9.3	Guidance system . . . . .	115
9.4	Same strategy . . . . .	115
9.5	Opposite strategy . . . . .	115
9.6	Delta strategy . . . . .	115
9.7	Guidance study flowchart . . . . .	116
9.8	Guidance study result . . . . .	116
9.9	Plastic tooth assessment . . . . .	117
9.10	Mirror study flowchart . . . . .	118
9.11	Abstracted THA task . . . . .	119
9.12	THA study design . . . . .	120
9.13	THA study feedback . . . . .	120
9.14	THA study tuning results . . . . .	122

## LIST OF TABLES

---

6.1	Length scaling of haptics APIs . . . . .	55
8.1	Spin sound parameters . . . . .	96
8.2	Drill sound parameters . . . . .	98
9.1	THA study demographics . . . . .	119
9.2	THA study questions . . . . .	121
9.3	THA study tuning directions . . . . .	121
9.4	THA study tuning . . . . .	121
B.1	Value types . . . . .	139
B.2	Common operations . . . . .	139
B.3	Common symbols . . . . .	141

## LIST OF ALGORITHMS

---

5.1	Collision Detection Meta . . . . .	32
5.2	Sample-Based CCD . . . . .	34
5.3	Translational CCD (CPU) . . . . .	35
5.4	Translational CCD Parallel (GPU) . . . . .	36
5.5	Contact Detection (CPU) . . . . .	38
5.6	Contact Detection Parallel (GPU) . . . . .	38
5.7	Estimation of Local Surface . . . . .	40
5.8	Contact Clustering . . . . .	42
5.9	Contact Resolution Meta . . . . .	43
5.10	Penalty Contact Resolution . . . . .	44
5.11	Adaptive Stiffness . . . . .	47
5.12	Impulse Contact Resolution . . . . .	49
5.13	Impulse Calculation . . . . .	50
5.14	Impulse Application . . . . .	50
5.15	Relative Point Velocities Calculation . . . . .	50
5.16	Hybrid Contact Resolution . . . . .	51
7.1	Material Removal Setup (CPU) . . . . .	75
7.2	Material Removal in Parallel (GPU) . . . . .	76
7.3	Sphere Removal – BVH Update . . . . .	77
7.4	Capsule Material Removal in Parallel (GPU) . . . . .	78
7.5	Dynamic Mesh Update . . . . .	80
7.6	Mesh Update Threading Handler . . . . .	81
7.7	Mesh Update Meta Routine . . . . .	82
7.8	SDF Generation . . . . .	84
7.9	Raw SDF Generation . . . . .	85
7.10	Parallel Bilateral 3D SDF Smoothing . . . . .	86
7.11	Parallel Marching Cubes with Normals and Colors . . . . .	89
7.12	Vertex Pooling . . . . .	90
A.1	Linear Contact Clustering . . . . .	137

## LIST OF CODE

---

6.1	Camera tool clutch . . . . .	58
7.1	Tool sphere packing augmentation . . . . .	69
7.2	Human bone sphere packing augmentation . . . . .	71
7.3	Human tooth augmentation . . . . .	72
7.4	Compacted mesh transfer GPU to CPU . . . . .	81
8.1	SteamVR's eye pose calculation . . . . .	100

ACRONYMS

---

CD	collision detection
CCD	continuous collision detection
VC	virtual coupling
DOF	degree-of-freedom
VPS	voxmap pointshell
SDF	signed distance field
VR	virtual reality
HMD	head-mounted display
THA	total hip arthroplasty
RCT	root canal treatment
RCAO	root canal access opening
CR	caries removal
BVH	bounding volume hierarchy
IST	inner sphere tree
UE	Unreal Engine
GE	game engine
MC	marching cubes
FOV	field of view
PMC	ProceduralMeshComponent
SDK	software development kit
COR	coefficient of restitution
API	application programming interface
GPU	graphics processing unit



**Part IV**

**Publications**

---

A VOLUMETRIC PENETRATION MEASURE FOR 6-DOF HAPTIC RENDERING  
OF STREAMING POINT CLOUDS
*Outline*


---

I.	INTRODUCTION . . . . .	151
II.	RELATED WORK . . . . .	152
III.	OUR VOLUMETRIC PENETRATION MEASURE . . . . .	152
IV.	OUR COLLISION DETECTION ALGORITHM . . . . .	153
	A. Boundary Spheres . . . . .	153
	B. Inside Spheres . . . . .	153
	C. Forces and Torques . . . . .	154
V.	POINT CLOUD NORMALS . . . . .	154
VI.	RESULTS . . . . .	154
	A. Haptic Rendering . . . . .	154
	B. Performance . . . . .	155
	C. Quality . . . . .	155
VII.	CONCLUSION AND FUTURE WORK . . . . .	155
	REFERENCES . . . . .	156

---

*Bibliography Entry*

Kaluschke, M., Weller, R., and Zachmann, G. (2017). **A volumetric penetration measure for 6-DOF haptic rendering of streaming point clouds.** In *2017 IEEE World Haptics Conference (WHC)*, pages 511–516

*Author's Contribution*

The author designed and programmed all of the presented algorithms, generated the pictures and diagrams and wrote the majority of the manuscript.

*Copyright*

Copyright © 2017 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

*This paper has been omitted due to copyright considerations.*

---

IMMERSIVE ANATOMY ATLAS — EMPIRICAL STUDY INVESTIGATING THE USABILITY OF A VIRTUAL REALITY ENVIRONMENT AS A LEARNING TOOL FOR ANATOMY

*Outline*

---

1.	INTRODUCTION . . . . .	160
2.	MATERIALS AND METHODS . . . . .	160
2.1	Immersive Anatomy Atlas . . . . .	160
2.2	Learning Environment in the Two High Schools . . . . .	160
2.3	Study Design . . . . .	161
2.4	Statistics . . . . .	162
3.	RESULTS . . . . .	163
3.1	Error Rates . . . . .	163
3.2	Processing Time . . . . .	163
3.3	Subjective Assessment of the Teaching Unit by Students of High School B . . . . .	163
4.	DISCUSSION . . . . .	164
5.	CONCLUSIONS . . . . .	165
	AUTHOR CONTRIBUTIONS . . . . .	165
	ACKNOWLEDGMENTS . . . . .	165
	SUPPLEMENTARY MATERIAL . . . . .	165
	REFERENCES . . . . .	165

---

*Bibliography Entry*

Weyhe, D., Uslar, V., Weyhe, F., Kaluschke, M., and Zachmann, G. (2018). **Immersive Anatomy Atlas — Empirical Study Investigating the Usability of a Virtual Reality Environment as a Learning Tool for Anatomy.** *Frontiers in surgery*, 5(73):73

*Author's Contribution*

The author designed and programmed the presented anatomy atlas and wrote the technical portions of the anatomy atlas.

*Copyright*

Copyright © 2018 Weyhe, Uslar, Weyhe, Kaluschke and Zachmann. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



# Immersive Anatomy Atlas—Empirical Study Investigating the Usability of a Virtual Reality Environment as a Learning Tool for Anatomy

Dirk Weyhe<sup>1\*</sup>, Verena Uslar<sup>1</sup>, Felix Weyhe<sup>2</sup>, Maximilian Kaluschke<sup>3</sup> and Gabriel Zachmann<sup>3</sup>

<sup>1</sup> Department for Human Medicine, Pius-Hospital, Medical Campus University of Oldenburg, University Hospital for Visceral Surgery, Oldenburg, Germany, <sup>2</sup> Friedrich – Harkort Schule, Städtisches Gymnasium Herdecke, Dortmund, Germany, <sup>3</sup> Department of Computer Science, Computer Graphics and Virtual Reality, University of Bremen, Bremen, Germany

## OPEN ACCESS

### Edited by:

René H. Fortelny,  
Wilhelminenspital, Austria

### Reviewed by:

Gabriel Sandblom,  
Karolinska Institutet (KI), Sweden  
Marko Korschake,  
Innsbruck Medical University, Austria

### \*Correspondence:

Dirk Weyhe  
dirk.weyhe@pius-hospital.de

### Specialty section:

This article was submitted to  
Visceral Surgery,  
a section of the journal  
Frontiers in Surgery

**Received:** 22 October 2018

**Accepted:** 15 November 2018

**Published:** 30 November 2018

### Citation:

Weyhe D, Uslar V, Weyhe F,  
Kaluschke M and Zachmann G (2018)  
Immersive Anatomy Atlas—Empirical  
Study Investigating the Usability of a  
Virtual Reality Environment as a  
Learning Tool for Anatomy.  
*Front. Surg.* 5:73.  
doi: 10.3389/fsurg.2018.00073

We developed a prototype of a virtual, immersive, and interactive anatomy atlas for surgical anatomical training. The aim of this study was to test the usability of the VR anatomy atlas and to measure differences in knowledge acquirement between an immersive content delivery medium and conventional learning (OB). Twenty-eight students of the 11th grade of two German high schools randomly divided into two groups. One group used conventional anatomy books and charts whereas the other group used the VR Anatomy Atlas to answer nine anatomy questions. Error rate, duration for answering the individual questions, satisfaction with the teaching unit, and existence of a medical career wish were evaluated as a function of the learning method. The error rate was the same for both schools and between both teaching aids (VR: 34.2%; OB: 34.1%). The answering speed for correctly answered questions in the OB group was approx. twice as high as for the VR group (mean value OB: 98 s, range: 2–410 s; VR: 50 s, 1–290 s). There was a significant difference between the students of the two schools based on a longer processing time in the OB condition in School B (mean OB in School A: 158 s; OB in School B: 77 s). The subjective survey on the learning methods showed a significantly better satisfaction for VR ( $p = 0.012$ ). Medical career aspirations have been strengthened with VR, while interest of the OB group in such a career tended to decline. The immersive anatomy atlas helped to actively and intuitively perform targeted actions that led to correct answers in a shorter amount of time, even without prior knowledge of VR and anatomy. With the OB method, orientation difficulties and/or the technical effort in the handling of the topographical anatomy atlas seem to lead to a significantly longer response time, especially if the students are not specially trained in literature research in books or texts. This seems to indicate that the VR environment in the sense of constructivist learning might be a more intuitive and effective way to acquire knowledge than from books.

**Keywords:** constructivist learning, virtual reality, immersive and interactive anatomy atlas, medical curriculum, virtual dissection

## INTRODUCTION

Using digital media for learning purposes is a much more discussed field of research than one might suspect. There were already theoretical considerations in the 80s to use computer games in class (1–3). The development is driven by the hope to learn more easily and effectively. The growing interest of researchers, educators, parents, players, and game developers has led to the development of so-called “serious games” and thus prepared the ground for digital game-based learning. The serious game research began in the early 90s. The number of publications has increased exponentially since then and is currently in a consolidation phase (4). The fields of application of serious games are manifold and they are already used in the military or for further training in companies (5). The discussion of the use of serious games in school has become increasingly intense in recent years because a playful environment is assumed to have a higher motivation potential for learning (6–11). Nowadays, especially the use of Virtual Reality (VR) serious games in school and higher education is discussed intensively, especially with regard to the intrinsic motivation potential (12, 13). In addition, a meta-analysis shows a high learning efficiency with VR in higher education (14).

This could for instance be explained by the constructivist view of learning, which claims that human experience and learning are subject to certain construction processes (15–18). Learning is influenced by sensory, neuronal, cognitive, and social processes. Neubert et al.’s approach claims that every learner learns on the basis of his own “experience,” and additionally adds his own values, beliefs, patterns, and previous experiences to the new information. On the basis of the learning pyramid, which is established in the community advocating constructivist learning, it is therefore assumed that only an average of 10% of what is read is remembered, since reading is a passive learning process. Practical actions, on the other hand, are active learning processes and already lead to a correct reproduction of what has been learned in 75% of all cases (Figure 1).

Against the background of practical actions, virtual worlds open up new possibilities to support learning processes more strongly through active interactions such as moving things, acting and being able to more strongly involve in the subject matter. Through this visual exploration and the virtual touching of objects, and the associated high immersion, learning content seems to be conveyed more intensively. Based on the constructivist learning theory, a higher learning efficiency is conceivable through these mechanisms.

So far, little is known about the application of VR technology in a medical learning environment (19–21). For instance, positional relationships in anatomy are difficult to convey by means of books (22). Here, VR might probably unfold the existing potential of the three-dimensional representation. In addition, a large number of examinations have shown that surgical Skills-Lab training improves individual performance and reduces the error rate (23–26). However, it is also known that typical carcass training and/or boxing trainers are either not sufficiently available or are perceived by the residents as unattractive courses (27–29). Therefore, one motivation of our group to

develop the immersive and interactive anatomy atlas (which in the future will be developed into an immersive surgical simulator) was the intention to create a learning tool which raises motivation.

We developed a prototype of the immersive anatomy atlas, featuring a virtual operating theater, where anatomical structures and arrangements of the human body can be explored through an immersive dissection. With this pilot project, we examined the feasibility and usability of the immersive anatomy atlas in comparison to the open book method (OB) under exam conditions in 11th grade students from two different high schools randomized into two groups (VR vs. OB). As a measure for the usability and ease of handling of both learning tools, we determined the error rate for 10 questions posed to each student, in addition to the duration for answering correctly.

## MATERIALS AND METHODS

### Immersive Anatomy Atlas

By wearing a headset with integrated screens for each eye, special lenses and software to bend the image, the all-round view of a virtual reality is simulated. The user is placed in a virtual operating room with realistic lighting and medical equipment. A virtual dummy with precise human anatomy is placed on the operating table, ready to be inspected (Figure 2). Individual organs can be manipulated via bi-manual controllers. The virtual hand is closed by pressing the action button on the back of the right controller with the index finger. For anatomical structures that are currently held in the hand, further information can be called up. The left controller can be used to hide nearby anatomical structures. Each organ will snap back to its original pose when it is within a translational and rotational threshold of said pose. Additionally, a context menu allows switching the controller-assigned actions, for left-handed people, as well as resetting the whole scene, including organ arrangement.

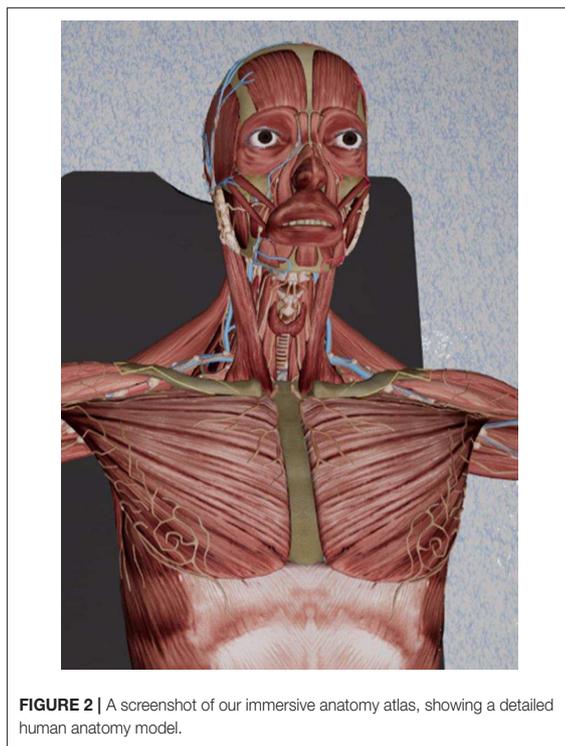
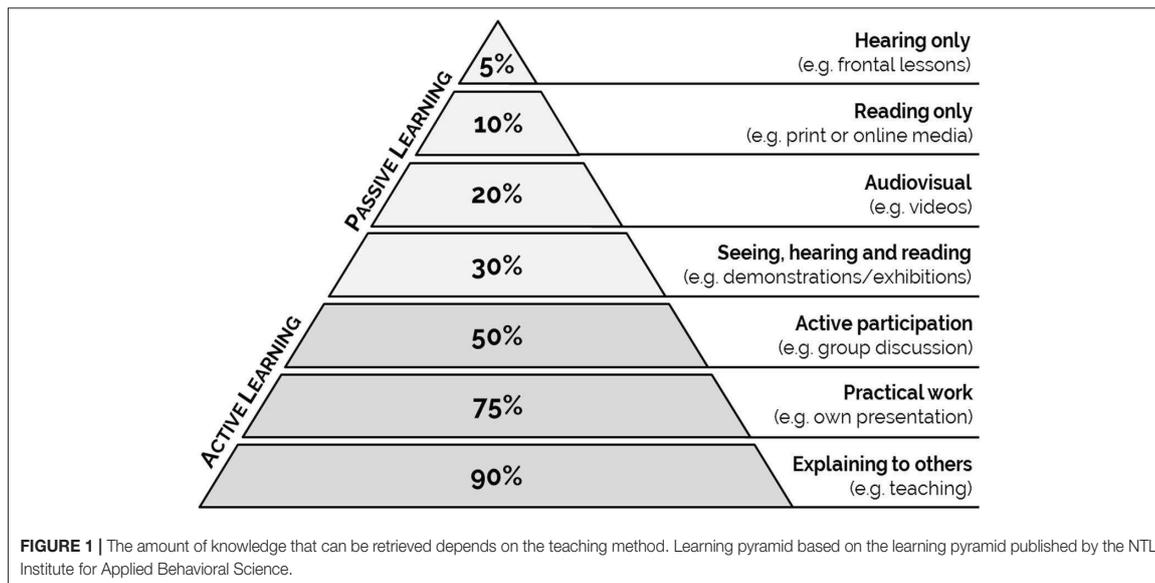
The grabbing action uses a specially designed collision detection algorithm that only allows grabbing of structures that are reachable from the outside. This increases the intuitiveness when interacting with the virtual environment. Collision detection is performed on the raw mesh structure, instead of being approximated by bounding volumes.

A video of the version of the anatomy atlas used in this study can be viewed here (Supplementary Video). There is also a video of the latest version, which has some additional features for manipulating the anatomical dummy (<https://youtu.be/JY50Wjh-olw>).

### Learning Environment in the Two High Schools

We conducted our study in collaboration with two high schools (“school A” and “school B”). We chose those two schools in order to draw a larger number of participants. Also, these two schools follow different teaching approaches, which allowed us to consider the effect of an immersive teaching method within different teaching contexts.

The eleventh-graders of the two high schools are normally introduced to new respective topics by different learning methods. In high school A, students typically receive what is



known as “smart” teaching geared toward utilization of digital media. Lectures by students are regularly delivered as power-point lectures, which are then shared at school via smartphone and projector. The communication during the lessons is also carried out via the school’s own internet-based communication

platform. In addition, modern “whiteboards” and tablet PCs are permanently used as interactive teaching media in the classroom. In high school B, students are explicitly trained in research in books, texts, and online media. Visual media are used more cautiously in high school B, but experimental investigations and a dialogue-based transfer of knowledge are promoted.

Medical students show very differing levels of anatomical knowledge, whereas the high school students are more comparable since they had no specific knowledge of anatomy. Therefore, we conducted the experiment with high school students to avoid bias due to heterogenous knowledge.

### Study Design

After the development of the immersive anatomy atlas, the ease of use of the atlas was tested via an exam with 10 questions at two different high schools in comparison to an exam in open book format. To test the usability, the questions were formulated from the perspective of a high school student of the same age from high school A in the context of a school research project, to ensure understandability and appropriateness of the questions. The questionnaire consisted of three multiple choice questions and six questions with freely formulated answers (see **Table 1**). In addition, a sketching task had to be completed (question 10). Since comparable tests are not available, this non-validated test was used.

Before the students were included in the study, the parents and participating students were informed in writing. Furthermore, a written declaration of consent was obtained for participation in the study, publication and potential photography.

### High School A

The test was carried out by a high school student of the same age. All participating students were randomized into one group using the immersive anatomy atlas (VR group;  $n = 5$ ) and

**TABLE 1** | Translated question catalog and correct answers.

Question	Correct answer
How many lobes does the right lung have? 2, 5, 6, 3, 4?	3
What is the structure between stomach and lungs look like?	Diaphragm
Name the annular muscle that surrounds the eye	Musculus orbicularis(oculi)
Name the Latin term of the kneecap	Patella
Name the nerve structure connecting the brain to the spinal cord	Medulla or brain stem
How many muscles are in direct contact with the femur? 13, 9, 6, 21, 27	13
How many parts does the calf muscle consist of?	3
Where is the thyroid gland? In front of or behind the windpipe?	In front of
What is the right temporal muscle (in German: "Schläfenmuskel") called in Latin?	Musculus temporalis
Sketch the Achilles tendon in proportion to the leg	Complete a schematic drawing

into another group provided with a topographic anatomy atlas and separate anatomy tables (OB group;  $n = 5$ ) (22, 30). The average age of the participants was 17 years (range: 16–17), the gender distribution was equal. The questions were answered on 2 consecutive days (day 1: VR Group, day 2: OB Group) There was no time limitation. At the beginning of the investigation, the analysis of response times and error rates was pointed out. The participating test persons were not allowed to discuss the contents of the examination.

At the beginning of the test, the participants of both groups were given an orientation time of 5 min. Within this time frame, the operating instructions of the immersive anatomy atlas and the familiarization in virtual space as well as the review of conventional teaching aids for orientation in the open book test (**Figures 3A,B**) were given for each group as a whole. The participants in both groups were under constant supervision. The questions were put to both groups orally. During the test, the time required to answer each question was measured and documented by the test manager.

### High School B

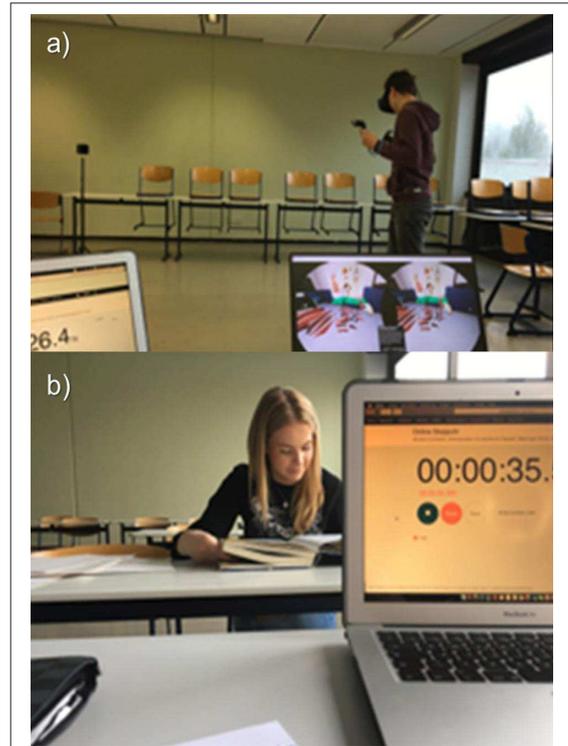
In high school B, the same questions were asked in a group of  $n = 18$  students specially trained to understand texts. Students were randomized to each group (OB:  $n = 10$ ; VR:  $n = 8$ ). The teaching aids in the "open book" to compensate for a potential methodological advantage in the VR group (31). In addition, two questions were added to the questionnaire for subjective appraisal of both teaching methods:

#### **Translation of the questions regarding the subjective appraisal of the teaching methods:**

*Assign a school grade for the teaching unit: (German school grade system: 1–6, 1 being the best grade)*

*Has this teaching unit given you the idea of taking up a medical profession (doctor, physiotherapist, paramedic, nursing, etc.)?*

Yes      Maybe      No      I had this idea before

**FIGURE 3** | (A,B) Photos of the conduction of the study at school A.

The trial was supervised by scientific staff of the University Clinic for Visceral Surgery at the Pius Hospital Oldenburg. The test was carried out according to the specifics described above for high school A. The participants were on average 17 years old (range: 16–19 years). The gender distribution was not equal.  $N = 13$  girls (OB = 7, VR = 6) and  $n = 5$  boys (OB = 3, VR = 2) from high school B took part in the study. An exchange about the content of the questions and the examination situation was impossible both between the pupils and between the schools.

### Statistics

The error rates and the processing time were analyzed. The data was tested for normal distribution using Shapiro-Wilke's test. For the normally distributed error rates, a three-way ANOVA was used to calculate statistical differences. The independent variables used were school affiliation (Gymnasium A and B), teaching conditions (VR and OB) and question number (Q1 to Q10). The processing time data was not distributed normally. Here a Kruskal-Wallis one-way ANOVA was calculated with the four groups VR in school A, VR in school B, OB in school A, and OB in school B. The Mann-Whitney *U*-Test was used to compare the subjective appraisal of the teaching methods. All statistical tests were performed with Sigma-Plot 12.0, the graphics were created with Origin 2016.

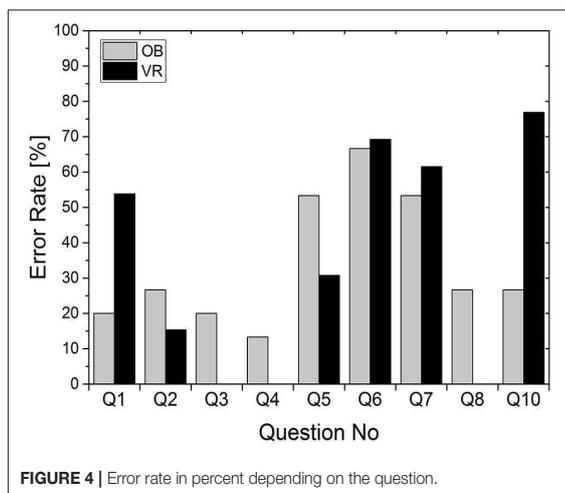


FIGURE 4 | Error rate in percent depending on the question.

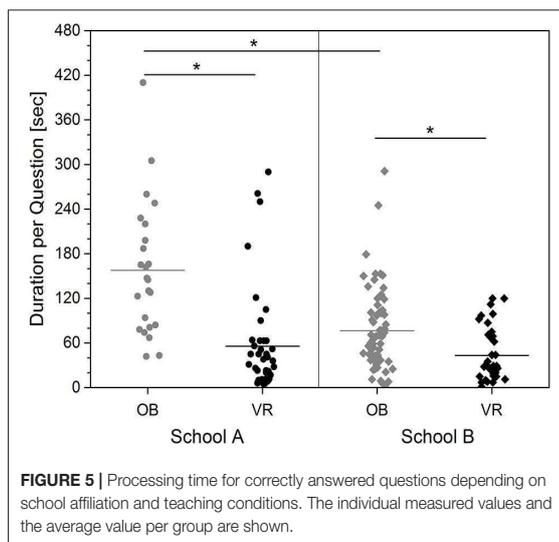


FIGURE 5 | Processing time for correctly answered questions depending on school affiliation and teaching conditions. The individual measured values and the average value per group are shown.

## RESULTS

The 28 participants of both high schools were motivated and concentrated. All students conducted the test in a very disciplined manner. All questions were dealt with and in the case of unclear solutions, especially in the OB group, the answers were commented on by the test persons. The experiment showed a content error in question 9, which could not be answered correctly due to a programming error in the immersive anatomy atlas. The question was therefore not evaluated. The following calculations therefore refer to  $n = 9$  questions.

### Error Rates

The three-way ANOVA showed no difference in the error rate in relation to the respective high school or the respective learning condition. The average error rate in the VR group was 34.2%, in the OB group 34.1% ( $p > 0.05$ ). Clear but not significant differences were found between the error rates for the respective questions [ $F_{(1,35)} = 2,913$ ;  $p = 0.076$ ; see Figure 4]. The VR error rate was at least equal to or better than the error rate of the OB group, except for questions 1 and 10.

### Processing Time

The average processing time for all correctly processed questions over all participants was 76 s per question (range: 1–410 s; see Figure 5). The average processing time in the VR Group was 50 s per question (range: 1–290 s). The average processing time in the OB group was 98 s (range: 2–410 s) per question. Thus, the processing time for the OB group is on average twice as high as for the VR group.

The Kruskal-Wallis One Way ANOVA showed a significant difference between the four groups School A in OB and VR, and School B in OB and VR [ $H_{(3)} = 44.324$ ;  $p < 0.001$ ]. The average processing time in the OB group in school A was 158 s (range: 42–410 s) and in the VR group 56 s (range: 5–290; Figure 5). In school B the average processing time in the OB group was 77 s (range: 2–291 s.) and in the VR group 42 s (range: 1–120 s).

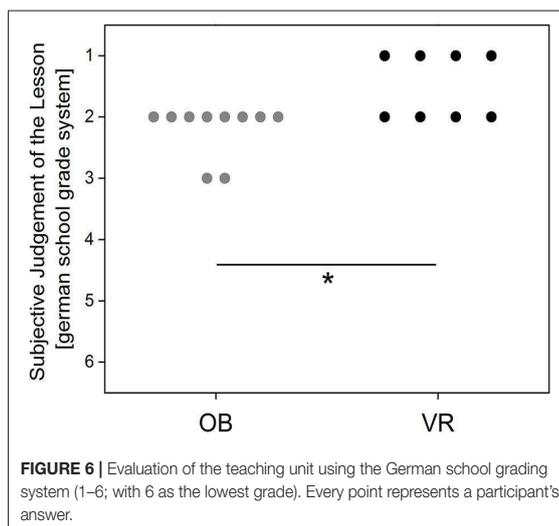


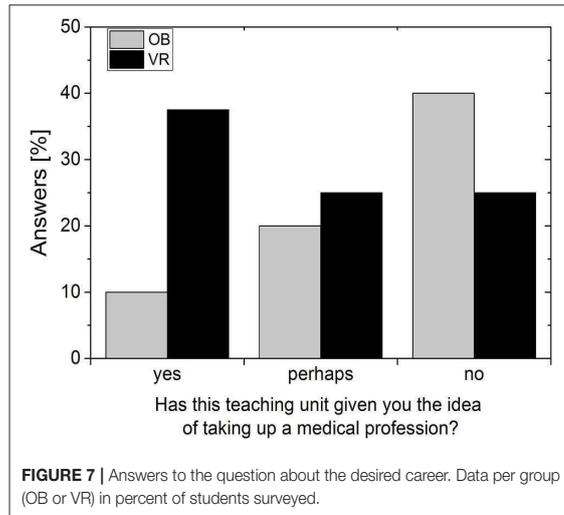
FIGURE 6 | Evaluation of the teaching unit using the German school grading system (1–6; with 6 as the lowest grade). Every point represents a participant's answer.

The group differences between schools and teaching methods are significant, except the difference between VR in both schools (Dunn's All Pairwise Multiple Comparison: Q always  $> 3.1$ ;  $p$  always  $<< 0.05$ ). Specifically, the pupils in school A seem to benefit more from the immersive learning method.

### Subjective Assessment of the Teaching Unit by Students of High School B

The subjective survey on the learning methods showed a significantly better school grade for the VR learning method (Figure 6; Mann-Whitney  $U$ -Statistic = 16.0;  $T = 52$ ;  $p = 0.012$ ).

The immersive teaching unit seems to have additionally aroused the desire and interest in the medical field (Figure 7). In the OB Group, on the other hand, interest seems to



decline. However, these results should be interpreted cautiously, because of the small number of students answering this question.

## DISCUSSION

The present study examines the acceptance and ease of use of a virtual anatomy atlas in a group of young people without specific prior anatomical knowledge in order to avoid bias. The fact that the results are not influenced by previous knowledge is shown by the almost equal error rates in VR and OB groups of 34.2 and 34.1% respectively. However, it seems as if most questions were easier to answer in the VR environment, since VR showed higher error rates only in two questions. A reason for the higher error rate for VR regarding question 1 might be a simple left/right orientation problem, whereas higher error rates for question 10 indicate a lower level of detail in VR, which will be corrected in future versions. Overall, one may conclude that the amount of knowledge to be gained is comparable between both conditions, and depends on the type of question.

However, our results also seem to reflect the learning environment of the students. In the immersive condition, students of both schools took approximately the same time to complete the tasks. With the OB method, however, the students who were less trained in text comprehension needed significantly longer to find a solution than the more trained group. This indicates that the intuitive interaction and the playful approach in the VR condition is more accessible to everyone than the more traditional learning method, for which one must acquire at least some knowledge about text interpretation. This more traditional method of knowledge acquisition and processing seems more complex and it seems as if it has to be specifically trained. In addition, our observations during the study support our considerations of a potentially increased

intrinsic motivation through playful learning using the VR approach.

OB and VR students in high school B, who are all very well-trained in text comprehension, show little difference in answering duration. However, the conventional teaching medium is not only rated worse, but beyond that, it reduces interest in the medical field. In comparison, the VR group showed twice as much potential interest in taking up a medical profession. These results are comparable with the findings of Fairén et al. (21), who showed that satisfaction of students' expectations was high in a VR anatomic course. In the randomized groups, our pilot study not only shows higher satisfaction with the VR teaching method, it also increased the interest to take up a medical profession.

In any case, the constructivist type of learning enabled by the VR anatomy atlas seems to lead to a faster solution, since the participants of the VR group found the right solution in a 50% shorter time. It is conceivable that the interactive and thus constructivist learning methodology of the immersive anatomy atlas has made it possible to better understand the information sought through active actions and thus to solve it more quickly. Constructivist learning by definition means that through the interaction of cognitive performance and simultaneous physical activity new and unknown topics can be grasped and classified more quickly.

This increase in learning efficiency and the fun of learning when using immersive digital media has led to the rapid further development of VR and AR technology in recent years as well as the development of various tools in the medical field, for example for learning anatomy or various (surgical) procedures (32–38). However, this development is now also leading to increased discussions about whether cadaver training is still up-to-date (39–43). With the current state of the art, from our point of view cadaver training is still irreplaceable insofar as it offers haptic feedback that cannot currently be produced with VR simulations. In addition, VR and AR systems are currently not designed for several simultaneous users, an important prerequisite for an educational exchange between pupils or between teacher and pupils. Thus, integration of haptic feedback as well as possibilities for several users in one OP simulator are current important research topics.

However, our study also shows that compared to the open-book method, the immersive anatomy atlas can currently already improve the learning effect for anatomical structures. Thus, with the help of the immersive anatomy atlas it was obviously easier for the participants to actively perform a targeted action according to the question, which then quickly led to a correct answer in over 60% of the questions, even without prior anatomical knowledge. With the conventional book method, orientation difficulties and/or the manual effort in using the topographical anatomy atlas in general seem to lead to a significantly longer response time. As shown by the fact that in the OB condition students from school B trained in text analysis were significantly faster than the untrained students from school A, but they still needed twice as long as the VR group from the same school to find the right solution.

Unclear in our study is the retention rate of the acquired knowledge. Further limitations of this study are the use of a non-scientifically validated questionnaire and not using a standardized intelligence test. For medical students, the benefit of the VR atlas could be, for example, that they are supported by the interactivity in memorizing names of and positional relationships between anatomical structures. Further studies with medical students should therefore be developed and carried out together with specialized learning theorists, psychologists, and university didactics to develop informative tasks more geared toward retention and spatial relations between anatomical structures.

## CONCLUSIONS

The comparative study of the usability of a VR anatomy atlas in high school students without previous anatomical knowledge shows not only that correct answers might be found 50% faster with the help of the digital medium. It also shows a higher acceptance of the learning unit. The effect is particularly clear for students learning in a “smart” learning environment. Students specially trained in text analysis are comparatively good in using a more traditional way to access knowledge, but even they profit significantly from the digital teaching medium. Further scientific interdisciplinary studies should follow this pilot study to formulate and validate the basis of a digital-based constructivist learning theory in medical studies.

## REFERENCES

- Malone TW, editor. What makes things fun to learn? Heuristics for designing instructional computer games. In: *Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems* (Palo Alto, CA: ACM) 1980.
- Malone TW. Toward a theory of intrinsically motivating instruction. *Cogn Sci.* (1981) 5:333–69.
- Lepper, M. “Intrinsic motivation and instructional effectiveness in computer-based education” in: *Aptitude, Learning, and Instruction: III. Conative and Affective Process Analyses*. Erlbaum: Hillsdale (1987). p. 255–96.
- Hoblitz A. *Spielend Lernen im Flow: Die Motivationale Wirkung von Serious Games im Schulunterricht*. Heidelberg; Berlin; Germany: Springer-Verlag (2015).
- David M, Sande C. *Serious Games: Games That Educate, Train, and Inform*. Boston, MA: Course technology, Cengage Learning (2006).
- Egenfeldt-Nielsen S. Overview of research on the educational use of video games. *Nord J Digit Liter.* (2006) 1:184–214. Available online at: [https://www.idunn.no/dk/2006/03/overview\\_of\\_research\\_on\\_the\\_educationaluseof\\_video\\_games?mode=pr](https://www.idunn.no/dk/2006/03/overview_of_research_on_the_educationaluseof_video_games?mode=pr)
- Egenfeldt-Nielsen S. *Beyond Edutainment: Exploring the Educational Potential of Computer Games*. Copenhagen: Lulu; Game-Research.com (2011).
- Kirriemuir J, McFarlane A. Literature review in games and learning. In: NESTA Futurlab Series. Bristol: NESTA Futurelab (2004). Available online at: [www.nestafuturelab.org/research/reviews/08\\_01.htm](http://www.nestafuturelab.org/research/reviews/08_01.htm) (Accessed November 22, 2018)
- McFarlane A, Sparrowhawk A, Heald Y. *Report on the Educational Use of Games*. TEEM (Teachers evaluating educational multimedia), Cambridge (2002).
- Mitchell A, Savill-Smith C. *The Use of Computer and Video Games for Learning: A Review of the Literature*. London: Learning and Skills Development Agency (2004). Available online at: [http://dera.ioe.ac.uk/5270/7/041529\\_Redacted.pdf](http://dera.ioe.ac.uk/5270/7/041529_Redacted.pdf) (Last Accessed November 22, 2018).

## AUTHOR CONTRIBUTIONS

DW and FW developed the study design. FW carried out the measurements at school A. VU carried out the measurement at school B together with two colleagues and was mainly responsible for the analyses of the data and the creation of the figures. DW and VU wrote the manuscript. MK developed the anatomy atlas with the help of several University of Bremen students and described the technical aspects of the anatomy atlas in the manuscript. GZ supervised the Bremen students and MK. All authors have revised the manuscript.

## ACKNOWLEDGMENTS

We would like to thank the students of the two participating high schools, and the two research assistants Sonja Janssen and Heike Nerenz for their help in carrying out the experiments.

Students at the University of Bremen helped develop the anatomy atlas, for which we are very grateful as well. We also would like to thank Lea Huang for script revision.

## SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://youtu.be/gGsLkDfxqLc> and <https://youtu.be/JY50Wjh-olw>

- Williamson B, Sandford R. (editors). Playful pedagogies: Cultural and curricular approaches to game-based learning in the school classroom. In: Williamson B and Sandford R. editors. *Handbook of Research on Improving Learning and Motivation Through Educational Games: Multidisciplinary Approaches*. IGI Global: Hershey (2011). p. 846–59.
- Boyle EA, Hainey T, Connolly TM, Gray G, Earp J, Ott M, et al. An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Comput Educ.* (2016) 94:178–92. doi: 10.1016/j.compedu.2015.11.003
- Hainey T, Connolly TM, Boyle EA, Wilson A, Razak A. A systematic literature review of games-based learning empirical evidence in primary education. *Comput Educ.* (2016) 102:202–23. doi: 10.1016/j.compedu.2016.09.001
- Merchant Z, Goetz ET, Cifuentes L, Keeney-Kennicutt W, Davis TJ. Effectiveness of virtual reality-based instruction on students’ learning outcomes in K-12 and higher education: a meta-analysis. *Comput Educ.* (2014) 70:29–40. doi: 10.1016/j.compedu.2013.07.033
- The Practice of Constructivism in Science Education*. Tobin K, editor. New York, NY: Routledge (1995).
- Siemens G. Connectivism: A learning theory for the digital age. *Int J Instr Tech Distance Learn.* (2005) 2. Available online at: [http://www.itdl.org/Journal/Jan\\_05/article01.htm](http://www.itdl.org/Journal/Jan_05/article01.htm) (Last Accessed November 22, 2018).
- Fosnot CT, Perry RS. Constructivism: A psychological theory of learning. In: *Eurographics 2017: Education Papers* Lyon: Eurographics (2017).
- Neubert S, Reich K, Voß R. Lernen als konstruktiver Prozess. In: Hug T, editor. *Einführung in Das Wissenschaftliche Arbeiten Wie kommt Wissenschaft zu Wissen?* Vol 1. Baltmannsweiler: Schneider Verlag Hohengehren (2001). p. 253–65.
- Seymour NE. VR to OR: a review of the evidence that virtual reality simulation improves operating room performance. *World J Surg.* (2008) 32:182–8. doi: 10.1007/s00268-007-9307-9
- McGaghie WC, Issenberg SB, Petrusa ER, Scalese RJ. A critical review of simulation-based medical education research: 2003–2009. *Med Educ.* (2010) 44:50–63. doi: 10.1111/j.1365-2923.2009.03547.x

21. Fairén González M, Farrés M, Moyes Ardiaca J, Insa E, editors. Virtual Reality to teach anatomy. In: *Eurographics 2017: Education Papers*. Lyon: European Association for Computer Graphics (Eurographics) (2017).
22. Rauber A, Kopsch F. *Anatomie des Menschen Bd. 1-4*. Stuttgart; New York, NY: Thieme (1987).
23. Kurashima Y, Hirano S. Systematic review of the implementation of simulation training in surgical residency curriculum. *Surg Today* (2017) 47:777–82. doi: 10.1007/s00595-016-1455-9
24. Bosse HM, Mohr J, Buss B, Krautter M, Weyrich P, Herzog W, et al. The benefit of repetitive skills training and frequency of expert feedback in the early acquisition of procedural skills. *BMC Med Educ*. (2015)15:22. doi: 10.1186/s12909-015-0286-5
25. Carlsen CG, Lindorff-Larsen K, Funch-Jensen P, Lund L, Konge L, Charles P. Module based training improves and sustains surgical skills: a randomised controlled trial. *Hernia* (2015) 19:755–63. doi: 10.1007/s10029-015-1357-6
26. Sonnadora RR, Van Vliet A, Safir O, Alman B, Ferguson P, Kraemer W, et al. Orthopedic boot camp: examining the effectiveness of an intensive surgical skills course. *Surgery* (2011) 149:745–9. doi: 10.1016/j.surg.2010.11.011
27. Palter VN, Orzech N, Aggarwal R, Okrainec A, Grantcharov TP. Resident perceptions of advanced laparoscopic skills training. *Surg Endosc*. (2010) 24:2830–4. doi: 10.1007/s00464-010-1058-2
28. Chang L, Petros J, Hess DT, Rotondi C, Babineau TJ. Integrating simulation into a surgical residency program. *Surg Endosc*. (2007) 21:418–21. doi: 10.1007/s00464-006-9051-5
29. Stefanidis D, Heniford BT. The formula for a successful laparoscopic skills curriculum. *Arch Surg*. (2009) 144:77–82. doi: 10.1001/archsurg.2008.528
30. Weber EMW. *Schemata der Leitungsbahnen des Menschen*. Heidelberg; Berlin: Springer (1999).
31. Paulsen F, Waschke J, Sobotta, *Atlas der Anatomie des Menschen Bd 1-3*. München; Jena: Urban & Fischer Verlag; Elsevier Health Sciences (2011).
32. Alaker M, Wynn GR, Arulampalam T. Virtual reality training in laparoscopic surgery: A systematic review & meta-analysis. *Int J Surg*. (2016) 29:85–94. doi: 10.1016/j.ijsu.2016.03.034
33. Yiannakopoulou E, Nikiteas N, Perrea D, Tsigris C. Virtual reality simulators and training in laparoscopic surgery. *Int J Surg*. (2015) 13:60–4. doi: 10.1016/j.ijsu.2014.11.014
34. Huber T, Paschold M, Hansen C, Wunderling T, Lang H, Kneist W. New dimensions in surgical training: immersive virtual reality laparoscopic simulation exhilarates surgical staff. *Surg Endosc*. (2017) 31:4472–7. doi: 10.1007/s00464-017-5500-6
35. Jain N, Youngblood P, Hasel M, Srivastava S. An augmented reality tool for learning spatial anatomy on mobile devices. *Clin Anat*. (2017) 30:736–41. doi: 10.1002/ca.22943
36. Bernardo A. Virtual reality and simulation in neurosurgical training. *World Neurosurg*. (2017) 106:1015–29. doi: 10.1016/j.wneu.2017.06.140
37. Ellington DR, Shum PC, Dennis EA, Willis HL, Szychowski JM, Richter HE. Female pelvic floor immersive simulation: a randomized trial to test the effectiveness of a virtual reality anatomic model on resident knowledge of female pelvic anatomy. *J Minim Invas Gynecol*. (2018). doi: 10.1016/j.jmig.2018.09.003. [Epub ahead of print].
38. Pulijala Y, Ma M, Pears M, Peebles D, Ayoub A. Effectiveness of immersive virtual reality in surgical training—A randomized control trial. *J Oral Maxillofac Surg*. (2018) 76:1065–72. doi: 10.1016/j.joms.2017.10.002
39. Korschake M, Brenner E. “Mors auxilium vitae”—Causes of death of body donors in an Austrian anatomical department. *Ann Anat*. (2014) 196:387–93. doi: 10.1016/j.aanat.2014.07.002
40. McMenamin P, McLachlan J, Wilson A, McBride J, Pickering J, Evans D, et al. Do we really need cadavers anymore to learn anatomy in undergraduate medicine? *Med Teach*. (2018). doi: 10.1080/0142159X.2018.1485884. [Epub ahead of print].
41. Lippert H. Medizinstudium: Sind Präparierübungen an der Leiche noch zeitgemäß? *Dtsch Arztebl*. (2012) 109:A-1758/B-422/C-140. Available online at: <https://www.aerzteblatt.de/archiv/128775/Medizinstudium-Sind-Präparieruebungen-an-der-Leiche-noch-zeitgemaess>
42. Ferrada P, Anand RJ, Amendola M, Kaplan B. Cadaver laboratory as a useful tool for resident training. *Am Surg*. (2014) 80:408.
43. Sharma M, Macafee D, Horgan AF. Basic laparoscopic skills training using fresh frozen cadaver: a randomized controlled trial. *Am J Surg*. (2013) 206:23–31. doi: 10.1016/j.amjsurg.2012.10.037

**Conflict of Interest Statement:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2018 Weyhe, Uslar, Weyhe, Kaluschke and Zachmann. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



### REALISTIC HAPTIC FEEDBACK FOR MATERIAL REMOVAL IN MEDICAL SIMULATIONS

#### *Outline*

---

I.	INTRODUCTION . . . . .	169
II.	RELATED WORK . . . . .	169
III.	OUR APPROACH . . . . .	170
	A. <i>Collision Detection and Material Removal</i> . . . . .	170
	B. <i>Force and Torque Computation</i> . . . . .	171
IV.	USE CASES . . . . .	172
	A. <i>Use Case: Dentistry</i> . . . . .	172
	B. <i>Use Case: Hip Replacement</i> . . . . .	173
V.	RESULTS . . . . .	174
VI.	CONCLUSION . . . . .	174
	ACKNOWLEDGEMENTS . . . . .	175
	REFERENCES . . . . .	175

---

#### *Bibliography Entry*

Kaluschke, M., Weller, R., Hammer, N., Pelliccia, L., Lorenz, M., and Zachmann, G. (2020). **Realistic Haptic Feedback for Material Removal in Medical Simulations**. In *2020 IEEE Haptics Symposium (HAPTICS)*, pages 920–926

#### *Author's Contribution*

The author designed and programmed all of the presented algorithms, generated the pictures and diagrams and wrote the majority of the manuscript.

#### *Copyright*

Copyright © 2020 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

*This paper has been omitted due to copyright considerations.*

---

### REVIEW OF HAPTIC RENDERING TECHNIQUES FOR HIP SURGERY TRAINING

#### *Outline*

---

1	<b>Introduction</b>	177
2	<b>Objectives</b>	179
3	<b>Haptic Rendering Techniques</b>	179
3.1	<i>Penalty Method</i>	179
3.2	<i>Impulse Method</i>	181
3.3	<i>Event-Driven Method</i>	182
3.4	<i>Proxy Method</i>	183
3.5	<i>Geometric Placement of the Proxy</i>	183
3.6	<i>Rigid-body dynamics simulation</i>	185
4	<b>Discussion</b>	185
5	<b>Results</b>	182
6	<b>Conclusion</b>	187
	<b>Acknowledgements</b>	187

---

#### *Bibliography Entry*

Ziadeh, T., Perret, J., Kaluschke, M., Knopp, S., and Lorenz, M. (2020). **Review of Haptic Rendering Techniques for Hip Surgery Training**. In *EuroVR 2020 Application, Exhibition & Demo Track: Proceedings of the Virtual EuroVR Conference*, number 381 in VTT Technology, pages 29–39. VTT Technical Research Centre of Finland

#### *Author’s Contribution*

The author wrote the chapter “Objectives”, helped in the literature research and in writing the rest of the manuscript. The diagram was also generated by the author.

#### *Copyright*

Copyright © VTT 2020

*This paper has been omitted due to copyright considerations.*

---

### A CADAVER-BASED BIOMECHANICAL MODEL OF ACETABULUM REAMING FOR SURGICAL VIRTUAL REALITY TRAINING SIMULATORS

#### *Outline*

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>189</b>
<b>2</b>	<b>Materials and methods</b> . . . . .	<b>190</b>
2.1	Sample preparation . . . . .	190
2.2	Experimental setup . . . . .	191
<b>3</b>	<b>Calculation</b> . . . . .	<b>192</b>
3.1	Data filtering, normalization and interpolation . . . . .	192
<b>4</b>	<b>Results</b> . . . . .	<b>193</b>
4.1	Forces and torques . . . . .	193
4.2	Real-time computation of the haptic feedback using spline interpolation . . . . .	194
4.3	Implementation approach . . . . .	196
<b>5</b>	<b>Discussion</b> . . . . .	<b>197</b>
5.1	Limitations . . . . .	199
<b>6</b>	<b>Conclusion</b> . . . . .	<b>199</b>
	<b>References</b> . . . . .	<b>200</b>
	<b>Author contributions</b> . . . . .	<b>200</b>
	<b>Competing interests</b> . . . . .	<b>200</b>

---

#### *Bibliography Entry*

Pelliccia, L., Lorenz, M., Heyde, C.-E., Kaluschke, M., Klimant, P., Knopp, S., Schleifenbaum, S., Rotsch, C., Weller, R., Werner, M., Zachmann, G., Zajonz, D., and Hammer, N. (2020). A Cadaver-Based Biomechanical Model of Acetabulum Reaming for Surgical Virtual Reality Training Simulators. *Scientific Reports*, 10(1):14545

#### *Author's Contribution*

The author implemented the approach into the haptic rendering application, produced the image on sphere packing, and helped writing the manuscript.

#### *Copyright*

Copyright © The Author(s) 2020



OPEN

# A cadaver-based biomechanical model of acetabulum reaming for surgical virtual reality training simulators

Luigi Pelliccia<sup>1,6</sup>, Mario Lorenz<sup>1,2,5,6</sup>, Christoph-E. Heyde<sup>2</sup>, Maximilian Kaluschke<sup>3</sup>, Philipp Klimant<sup>1,4</sup>, Sebastian Knopp<sup>1</sup>, Stefan Schleifenbaum<sup>2</sup>, Christian Rotsch<sup>2,4</sup>, René Weller<sup>3</sup>, Michael Werner<sup>4</sup>, Gabriel Zachmann<sup>3</sup>, Dirk Zajonz<sup>2</sup> & Niels Hammer<sup>2,4,5</sup>✉

Total hip arthroplasty (THA) is a highly successful surgical procedure, but complications remain, including aseptic loosening, early dislocation and misalignment. These may partly be related to lacking training opportunities for novices or those performing THA less frequently. A standardized training setting with realistic haptic feedback for THA does not exist to date. Virtual Reality (VR) may help establish THA training scenarios under standardized settings, morphology and material properties. This work summarizes the development and acquisition of mechanical properties on hip reaming, resulting in a tissue-based material model of the acetabulum for force feedback VR hip reaming simulators. With the given forces and torques occurring during the reaming, Cubic Hermite Spline interpolation seemed the most suitable approach to represent the nonlinear force–displacement behavior of the acetabular tissues over Cubic Splines. Further, Cubic Hermite Splines allowed for a rapid force feedback computation below the 1 ms hallmark. The Cubic Hermite Spline material model was implemented using a three-dimensional-sphere packing model. The resulting forces were delivered via a human–machine-interaction certified KUKA iiwa robotic arm used as a force feedback device. Consequently, this novel approach presents a concept to obtain mechanical data from high-force surgical interventions as baseline data for material models and biomechanical considerations; this will allow THA surgeons to train with a variety of machining hardness levels of acetabula for haptic VR acetabulum reaming.

Total hip arthroplasty (THA) is considered one of the most successful procedures in orthopedic surgery<sup>1</sup> with more than 200,000 cases *per annum* performed in Germany<sup>2</sup> and a predicted number of 500,000 per annum in 2020 in the United States alone<sup>3</sup>. In spite of THA being a highly successful procedure, a number of complications related to this surgery exist, including aseptic loosening, early dislocation following THA, and misalignment. Some of these complications may be related to the surgical techniques deployed for THA.

To date, there is a lack of training opportunities for the various steps involved in THA in a standardized setting and without putting the patient at risk by the less experienced trainee. Human cadaveric tissues may here serve as an ideal model for trainees to gain experience, especially when supported by experienced senior colleagues. However, these scenarios are missing a standardized setting with predictable structure and haptics of the hip, which would be helpful to get a general appreciation for basic techniques and forces applied to the human system while at the same time having the opportunity to train the same intervention with exactly the same conditions.

<sup>1</sup>Professorship Machine Tool Design and Forming Technology, Professorship Factory Planning and Factory Operation, Chemnitz University of Technology, Reichenhainer Straße 70, 09126 Chemnitz, Germany. <sup>2</sup>Department of Orthopedics, Trauma and Plastic Surgery, University Hospital Leipzig, Liebigstraße 20, 04103 Leipzig, Germany. <sup>3</sup>Department of Computer Graphics and Virtual Reality, University of Bremen, Bibliothekstraße 5, 28359 Bremen, Germany. <sup>4</sup>Fraunhofer Institute for Machine Tools and Forming Technology IWU, Nöthnitzer Straße 44, 01187 Dresden, Germany. <sup>5</sup>Department of Macroscopic and Clinical Anatomy, Medical University of Graz, Harrachgasse 21, 8010 Graz, Austria. <sup>6</sup>These authors contributed equally: Luigi Pelliccia and Mario Lorenz. ✉email: niels.hammer@medunigraz.at

Further to this, the approach using cadavers has limitations regarding the availability, and potential changes induced by the post mortem interval or the chemicals used for anatomical fixation<sup>4–6</sup>. Virtual Reality (VR) provides new opportunities to establish highly standardized training scenarios with realistic haptic behavior for THA surgery. Interventions simulated with VR can be trained potentially for an indefinite number of times. VR has already been successfully established in arthroscopy or laparoscopic surgery<sup>7–9</sup>. However, to date, no VR-based simulation, which includes realistic haptic feedback, has been developed for THA. Existing commercial VR simulators for THA are not providing a realistic haptic feedback due to the lack of haptic devices, which are capable of delivering the necessary high forces. In case of commercial VR-based simulators, it is also often unclear if the underlying material models used for the haptic rendering are basing on empirical biomechanical data or the feeling of surgeons involved in their development process. Whilst a lot can be learned from solely visual training simulation, especially for the acquisition of surgical skills, a realistic haptic simulation is at least as important; surgeons rely a lot on their haptic feeling during surgeries. Therefore, one of the main challenges for the development of a haptic VR-based THA simulation environment, is to provide a realistic force-feedback (haptic-feedback) to the user. A number of steps in THA need to be modeled, including the (1) surgical approach, (2) incision of the joint and removal of the femoral head, (3) reaming of the acetabulum and (4) femur, as well as (5) implanting the prosthetic components and (6) wound closure. Both haptic and visual properties are of fundamental interest to develop THA simulators in VR. A feasible approach to provide a realistic haptic feedback is to develop a material model to be used for predicting the interactional forces between the surgical tools and the different tissues.

In recent years, a plethora of research has been conducted aiming to identify the material properties of the acetabulum by using both cadaveric experiments<sup>10,11</sup> and Finite Element Analysis (FEA)<sup>12</sup>. These studies' results could potentially be used to build a simulation model of the acetabulum, e.g., based on FEA, and simulate its biomechanical interaction with the reamer. However, the simulator is required to deliver a real-time force feedback to the user updated with a high frequency. Whilst there is no absolute threshold for the update rate to deliver a reliable and stiff haptic behavior it is shown that faster update rates are leading to stiffer behavior<sup>13,14</sup>. This represents one of the main requirements for realistic surgical VR simulators. As a rule-of-thumb-criteria, an update rate of 1 kHz is considered to provide reliable and stiff haptic behavior<sup>15</sup>, although even at lower update rates, comparable results could be achievable<sup>16</sup>. We decided to set 1 kHz as the requirement for the force feedback calculation speed. Such performance would usually not be feasible using the FEA due to the computational load it requires.

Addressing these issues, this given study measured the direct interactional forces between the reamer and the acetabulum, determined an analytical representation of force-torque as a function of reamer displacement, and implemented this material model in the simulation environment. This study presents the very first step towards developing a surgery simulator capable of delivering biomechanical-based force feedback. Based on the mechanical data, the objective was to develop a VR-based surgery simulator capable of computing the haptic feedback in 1 ms. The resulting data was implemented into a human-machine-interaction certified robotic arm as a force feedback device, equally capable of reacting in the 1 ms timeframe as part of the surgical VR training and as a safety feature for the user.

To achieve this goal, the measured data were interpolated to provide a mathematical description of force-torque as a function of reamer displacement. Two interpolation techniques were implemented to provide an effective mathematical approximation of the data, optimizing computational time. The workup will be described and discussed in detail, including the mechanical testing setup and a self-developed fixture for holding the reamer and the acetabulum in a standardized manner, the reaming itself, as well as the material data and related material models.

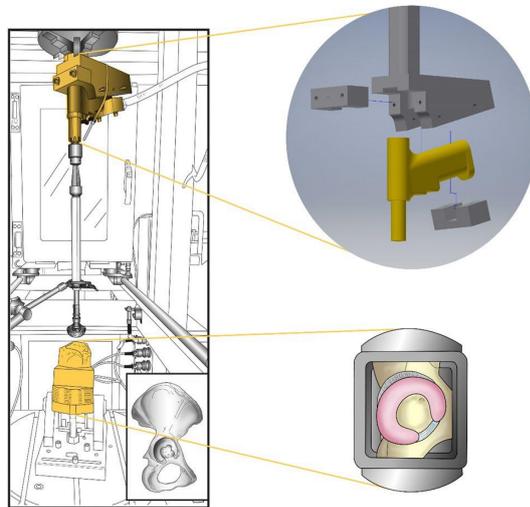
## Materials and methods

Mechanical tests were conducted on cadaveric acetabular specimens to assess their biomechanical behavior under conditions similar to hip surgery. The tissues were obtained in accordance with the Saxonian Death and Funeral Act (version 2014) and University of Otago ethics approval (H17/20), and the experimental protocols were approved by these given institutions. While alive, the body donors gave their informed consent to the donation of their tissues for research purposes. Twenty-four cadaveric acetabula were retrieved from human cadavers (16 females, 8 males; mean age  $87 \pm 8$  years, age range 74–102 years).

**Sample preparation.** Surrounding soft tissues, including the hip muscles, were grossly removed from the hips before the ilio-, ischio- and pubofemoral ligaments were exposed<sup>17</sup>. Care was taken to transect these ligaments as distally as possible before dislocating the femoral heads. The capsular ligaments and the teres ligament of the hip joint were then removed carefully at the level of the acetabular labrum. Following this, the innominate bone with the intact acetabulum was trimmed for all specimens with dimensions fitting into the customized support device, while assuring that the cartilage, the entire subchondral bone lamella and sufficient bone stock measuring at least 15-mm in all directions was still adjacent to the specimen. It was then embedded in ceramic-reinforced polyurethane resin (RenShape solutions, Huntsman International LLC, Salt Lake City, USA) in the support device, with the resin having just enough viscosity to allow thorough anchoring of the specimen with no excessive resin penetration. The labral rim of the acetabulum was positioned using a custom-developed three-dimensional (3D) positioning device the support was mounted on. Care was taken to align the labral rim perpendicular to the axis of rotation of the reamer. Prior to the mechanical tests, the tissues were rinsed in isotonic 0.9 mass% saline and kept moist throughout the mechanical experiments.



**Figure 1.** Reaming of the acetabulum during the hip surgery. The operating surgeon (on the left) uses a reaming tool to remove cartilage from the acetabulum of a left sided hip joint. Retractors hold open the surgical site, facilitated by two assisting surgeons (hands on the right).



**Figure 2.** Experimental setup (left) showing the mounted specimen in a bi-axial testing machine and the reamer mounted above. Reaming tools of matching sizes were used for each of the specimens. The insert shows a right sided innominate bone. The explosion view on the right top shows the hand piece of the reaming tool and a customized fixture to attach it to the bi-axial testing machine. Bottom right shows the top view of the acetabular region of a right sided innominate bone mounted in a mold prior to the reaming experiments. The cartilage is depicted in pink.

**Experimental setup.** Both the experimental setup and protocol were developed to replicate the real surgery conditions as closely as possible. Starting from observation of real surgery (see Fig. 1), an experimental setup was built (see Fig. 2). The reamer was connected to a biaxial material testing machine (DYNA-MESS, Aachen, Germany, 10 kN and 200 Nm certified measure range) to determine forces and torques along the direction of the reaming (see Fig. 2). For the connection of the reamer with the test machine and the embedding of the acetabula, dedicated fixators were CAD designed and manufactured from aluminum (see Fig. 2).

Forces and torques were measured along the direction of the reaming tool axis between the reamer and the acetabulum under both static (no reaming of acetabulum cartilage) and dynamic (reaming of acetabulum cartilage) conditions. The torques were caused by the friction that arose between the reaming tool and the acetabulum when the cartilage was removed. Thus, during the static tests, no torque arose. Three test series were conducted on each sample—one static test followed by two dynamic tests. All tests started from a position at which the contact force between the switched off reamer and the sample was 20 N. The test cycle began after that the reamer was switched into operating mode. For the tests, cadaveric tissue samples were numbered according to the test sequence followed in the experimental protocol (e.g., see sample 22 representing the 22nd test conducted). The angle of the reamer rotation was kept perpendicular relative to the equator of the acetabulum for standardization purposes in order to allow for baseline data acquisition.

During the **static tests**, the compression force was measured when the reamer was pushed into the acetabulum with a constant feed rate of 0.07 mm/s. The test stopped when (1) the force reached 800 N or (2) the tool displacement reached 20 mm. When performing **dynamic tests**, the reamer was operating to remove cartilage from the acetabulum in two consecutive trials with two different reaming tool sizes (labeled Dynamic 1 and Dynamic 2, respectively). The reaming tool for the test was chosen to match the diameter of the acetabulum, which was measured using a precision caliper. For the first trial, once the diameter of the acetabulum was measured (e.g. 51.2 mm), the next larger reamer with the rounded size was used (e.g. 52 mm), and for the second trial, the subsequently 2-mm larger reaming tool was chosen (e.g. 54 mm). The samples for the dynamic tests were further split into subgroups using two different feed rates, 0.01 mm/s and 0.03 mm/s. Dynamic tests were stopped if (1) the friction between the reamer and the sample caused the reamer to discontinue rotating, or (2) the tool displacement in the reaming direction exceeded 20 mm. These numbers and conditions assured that the cartilage was removed entirely in all the specimens.

This test setup measured displacement, force and torque (for dynamic data) in the reaming direction, with a sample time of 0.01 s. Thus, the test outcomes are displacement, force and torque (for dynamic data) vectors sampled at 0.01 s. As the measurements were conducted along one axis only, just one of the three vector components is meaningful, while the other two components are zero.

## Calculation

**Data filtering, normalization and interpolation.** Force and torque data coming from dynamic tests were filtered by determining the appropriate cut-off frequencies for each dataset by means of a Fourier analysis, in order to build customized low-pass filters for each test, and to remove unwanted oscillations. Force data coming from the static tests did not require filtering as the reaming tool was turned off.

Force and torque data coming from both static and dynamic tests were then normalized with reference to the maximum vertical displacement for each test set, to compare the different sample responses. For determining an analytical description of forces and torques, piecewise interpolation was implemented using both Cubic Splines and Cubic Hermite Splines to approximate the filtered and normalized data. Using spline interpolation to approximate the measured curves largely reduces the computational and data storage requirements of the material model. The appropriate number of interpolation nodes within the displacement interval was automatically determined by means of a customized algorithm, considering the second order derivative of the force/torque diagram.

Specifically, a threshold for the second derivative was set and the points in which the derivative was above the threshold were selected. The aim of the algorithm was to obtain larger numbers of nodes in the interval in which the slope variations were higher. To determine the appropriate number of nodes to be used for the interpolation, a desired value of the maximum norm of the approximation error (absolute error value) between the measured data and the interpolation curve was set, defined as follows:

$$e_a = \frac{v - v_{approx}}{v} \times 100,$$

where  $v$  and  $v_{approx}$  were the exact and the approximated values, respectively.

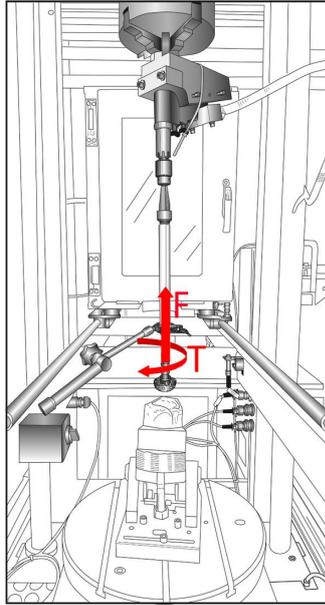
It is worth noticing that the given definition of error only aimed at providing a numerical value to compare the behavior of the interpolation curves, in addition to the visual comparison of the resulting graphs. For a desired value of the maximum error, starting from the maximum value of the second derivative, the threshold of the second derivative is recursively decreased by a defined sample step, and the algorithm recursively determines the number of nodes, calculating the related maximum norm of the approximation error. The calculation stopped when the maximum error was less or equal to the desired error value. The value of displacement at which the desired maximum error can also be determined by the algorithm. Lagrange interpolation schemes represented the easiest solution to the interpolation problem of  $N$  points. It defined an interpolation polynomial of minimum degree ( $N-1$ ) which fitted the  $N$  points to be interpolated. The generic expression for the polynomial  $p$  of degree  $N-1$  was  $p(x) = a_0 + a_1x + a_2x^2 + \dots + a_{N-1}x^{N-1}$  whose values in each node are equal to the desired values in that node, i.e., the data values  $f_i$  at the nodes. This resulted in a set of  $N$  equations.

$$p(x_i) = f_i, i = 0, \dots, N,$$

that allowed the determination of the  $N$  unknown polynomial coefficients. The polynomial degree directly depended on the number of nodes. Specifically, the degree of the polynomial increased with the number of nodes considered. When a large number of nodes was interpolated, the resulting polynomial heavily oscillated in the nodes close to the boundary of the interpolation intervals (so called Runge phenomenon)<sup>18</sup>. To provide a computationally effective model avoiding Runge's phenomenon, the data were processed using a piecewise interpolation, based on a third-order Spline<sup>19,20</sup>. The piecewise interpolation approach using both Cubic Splines and Cubic Hermite Splines allowed working with polynomials of fixed degree, independent from the number of nodes considered, avoiding Runge's phenomenon. Once the appropriate number  $N$  of nodes was determined, the displacement interval was divided in  $N-1$  sub-intervals  $[d_i, d_{i+1}]$ , defined by each pair of nodes. The points  $d_i$  and  $d_{i+1}$  were called knots and the related Spline in each of the interval was defined as follows.

$$s_i(d) = a_i(d - d_i)^3 + b_i(d - d_i)^2 + c_i(d - d_i) + d_i. \quad (1)$$

The coefficients of the polynomial in each interval, for the Cubic Spline, were determined by setting the values of the Spline equal to the desired values  $f_i$  in each node, as well as the continuity of the Spline and its first and second derivative at the nodes in order to achieve a steady connection between the spline segments (piecewise-spline curve is  $C^2$  in knot). The aforementioned conditions lead to the following system of equations:



**Figure 3.** Force (F) and Torque (T) applied to the reaming tool during the tests.

$$\begin{aligned}
 s_i(d_i) &= f_i & i = 0, \dots, n-1 & & s_{n-1}(d_n) &= f_n \\
 s_i(d_{i+1}) &= s_{i+1}(d_{i+1}) & i = 0, \dots, n-1 & & & \\
 s'_i(d_{i+1}) &= s'_{i+1}(d_{i+1}) & i = 0, \dots, n-1 & & & \\
 s''_i(d_{i+1}) &= s''_{i+1}(d_{i+1}) & i = 0, \dots, n-1 & & & 
 \end{aligned} \quad (2)$$

For the Cubic Hermite Splines the coefficients were determined by setting the values of the Spline, and its first derivative, equal to the desired values  $f_i$  and  $f'_i$  in each node, as well as the continuity of its first derivative at the nodes in order to achieve a steady connection between the spline segments (piecewise-spline curve is  $C^2$  in knot). The aforementioned conditions lead to the following system of equations:

$$\begin{aligned}
 s_i(d_i) &= f_i & i = 0, \dots, n-1 & & s_{n-1}(d_n) &= f_n \\
 s_i(d_{i+1}) &= s_{i+1}(d_{i+1}) & i = 0, \dots, n-1 & & & \\
 s'_i(d_{i+1}) &= f'_i & i = 0, \dots, n-1 & & & \\
 s''_i(d_{i+1}) &= s''_{i+1}(d_{i+1}) & i = 0, \dots, n-1 & & & 
 \end{aligned} \quad (3)$$

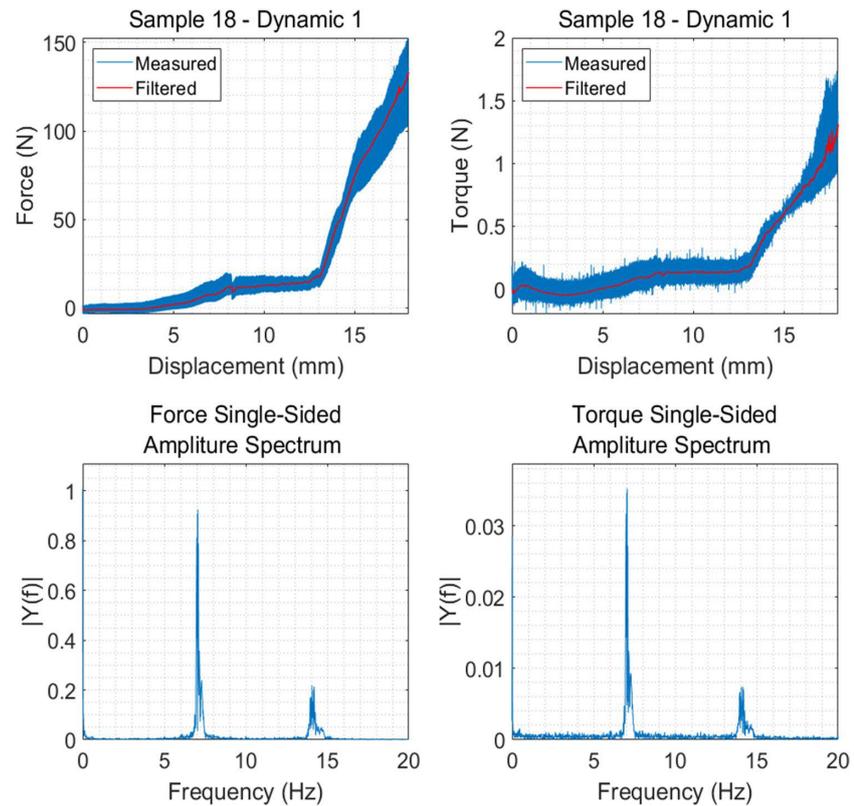
The conditions (2) and (3) allowed determining the expression (1) for each interval, leading to a set of equations that provided the appropriate value of the feedback force according to the reamer's displacement and with the set error threshold. Cubic Hermite polynomials are more complex than the Cubic Splines, but they have no overshoots, if the data to be interpolated is not smooth as in the given case.

## Results

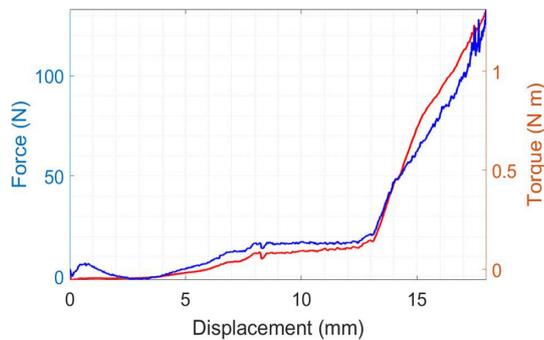
**Forces and torques.** Of the 72 data sets from 24 human acetabula used for the experiments, 62 resulted in valid data. Ten data sets were excluded from further evaluation due to (1) invalid measurements caused by alterations in the feed rates during the experiments (2 data sets), (2) acetabula breaking out of their resin embedding during the reaming (4 data sets) and (3) acetabula which were reamed completely after a 10-mm displacement (4 data sets). Detailed information concerning the tests results is given in the supplement material. Figure 3 shows the force and torques applied to the reaming tool during the execution of the tests.

Figure 4 shows one example of force and torque measurements, before and after the filtering, alongside the frequency spectrum. Figure 5 shows an example of the normalized force and torque measurements. Similar figures for all samples are given in the supplement material.

As shown in Fig. 5, it appears that the trends of forces and torques look similar, reflecting their physical connection being consistent with the definition of torque as a product of a force for distance. The torque on the vertical axis was generated from the friction on the surface of the reamer, which is depended on the vertical force. The reason for the forces and torques remaining close to 0 during the first part of the reaming is likely caused by the slow feed rate giving the reamer sufficient time to completely remove the material so that almost no force and torque is created between the reamer and acetabulum. Further, most of the forces starting from 0 N despite the contact force of 20 N results from the minimal time difference between the start of the reamer



**Figure 4.** Raw and filtered force and torque measurements as well as the frequency spectrum displacement for sample 18, set Dynamic 1, feed rate 0.01 mm/s.

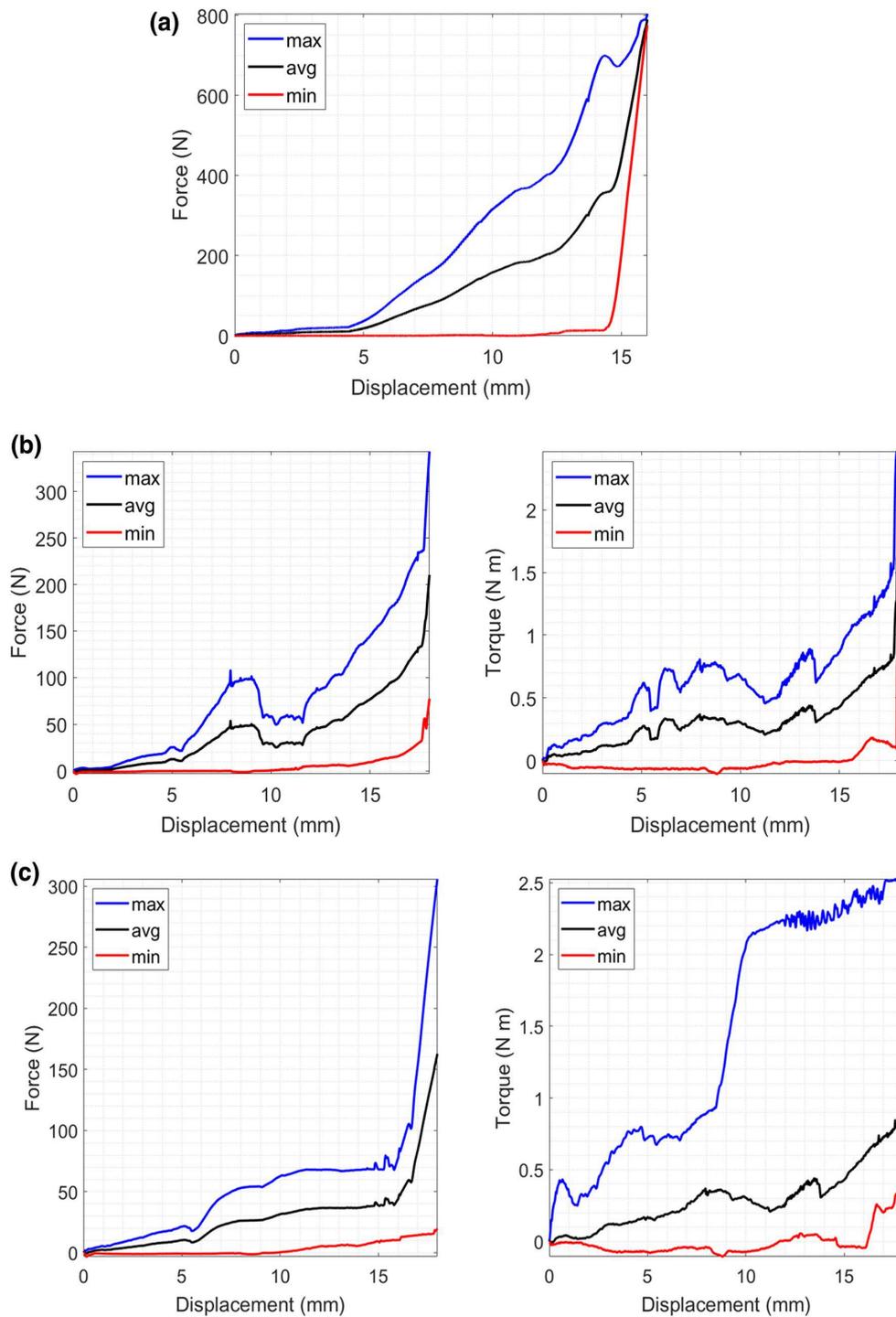


**Figure 5.** Force and torque after the filtering and the normalization with respect to the vertical displacement for sample 18, set Dynamic 1, feed rate 0.01 mm/s.

and the start of the test program. Thus, material was removed leading to a reduction of the contact force to 0 N at the beginning of the test.

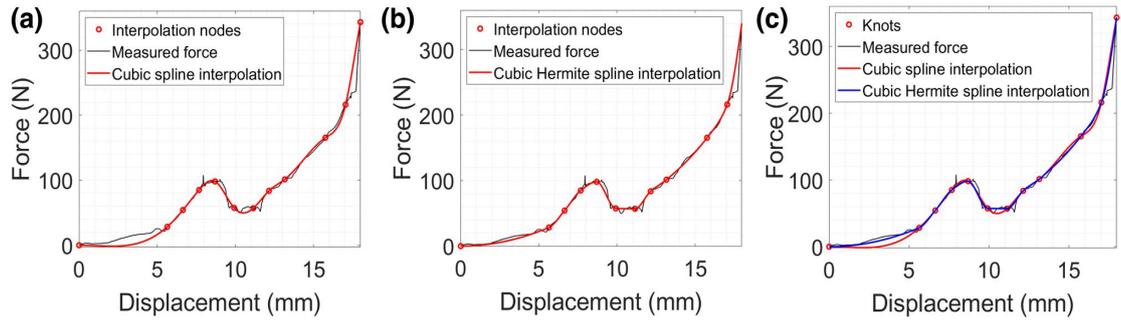
The maximum, average and minimum forces and torques for the test sets Static, Dynamic 1, and Dynamic 2 were calculated to provide a valid spectrum of occurring forces and torques (see Fig. 6). The depicted curves are based on the filtered and normalized data. The maximum, average and minimum curves are composed from different samples, as for each displacement value the maximum, average and minimum values at this point out of all samples were taken.

**Real-time computation of the haptic feedback using spline interpolation.** In order for the VR THA simulator to deliver real-time force feedback, it is necessary that both forces and torques value are updated



**Figure 6.** Maximum, minimum and average forces and torques for the test set Static (a), Dynamic 1 (b) and Dynamic 2 (c).

with a frequency of 1 kHz<sup>15</sup>. Therefore, on basis of the filtered and normalized data, an analytical formulation of the reamer-acetabulum interactional forces/torques, as a function of the reamer displacement, based on



**Figure 7.** Piecewise Cubic interpolation of the maximum force feedback for the test set Dynamic 1 based on Cubic Spline (a) and Cubic Hermite Spline (b), and comparison between the interpolation techniques (c) in the case of 12 interpolation nodes.

interpolation of the data was determined, using the algorithm described in “Data filtering, normalization and interpolation” section.

Figure 7a shows an example of interpolation of the maximum force feedback for the test set Dynamic 1, based on Cubic Splines. Using the algorithm described in “Data filtering, normalization and interpolation” section, 12 interpolation nodes have been determined to achieve a maximum norm of the approximation error of 41%, corresponding at a reamer displacement of 4.7 mm. A more detailed interpolation (higher number of interpolation nodes) can be achieved reducing the norm of the approximation error.

The main advantage of Cubic Spline interpolation approach is that it produced smooth and accurate results if the data set of interpolation nodes had a smooth distribution, otherwise it presented overshooting. Cubic Hermite Spline has been determined for the same 12 interpolation nodes, in order to make a comparison between the two techniques. The smooth and accurate interpolation of the Cubic Hermite Splines resulted in a maximum norm of the approximation error of 14%, at the same reamer displacement, which was much smaller than the 41% obtained with Cubic Splines. Figure 7b shows the interpolation of the maximum force feedback for the test set Dynamic 1, based on Cubic Hermite Splines with 12 nodes.

The interpolation technique based on Cubic Hermite Splines provided a better approximation of the data for the same number of interpolation nodes. Figure 7c shows a comparison between the two interpolation techniques.

**Implementation approach.** To achieve a simulation model that represents the experimental data in 3D space, the hip bone mesh was converted into a volumetric representation, called sphere packing<sup>21</sup>. This involved filling the entire mesh with non-overlapping spheres (see Fig. 8), to optimize volume coverage. Starting from a uniform density distribution, a haptic rendering algorithm was designed, tracking a physically constraint hip reamer tool. The forces that act upon the virtual tool were rendered to the KUKA iiwa robot as force feedback. This simulation took into consideration all spheres with which the virtual reamer was in contact with and calculated a current density by weighting each sphere’s density by the contact surface between the virtual reamer and the individual spheres. The surface contact was also used to compute other surface material properties, such as friction and surface normal. The surface normal can only be reasonably approximated when all spheres within a sphere-packing are non-overlapping. Otherwise, the same geometrical features would be considered multiple times. This would incorrectly skew the resulting normal of that feature.

Based on this simulation model, a virtual experiment was established to mimic the real experiment. A virtual reamer approached a hip bone, which was fixed in space in the same angle as the physical experiment. The simulation resulted in a force curve, which gave a mean error compared to the experimental data, represented by the Cubic Hermite Spline approximation. In this way, one could relate the reamer displacement to the measured force magnitude represented by the Cubic Hermite Spline. During simulation, the density distribution of the spheres was optimized by particle swarm optimization with the goal of minimizing the error between simulated and measured force at each time step. This way an accurate volumetric model of the hip from the measured experimental data is created. The size of the spheres determines the resolution at which this optimization can operate, with smaller spheres resulting in a higher resolution. However, an increased number of spheres negatively affects the performance, therefore both effects have to be considered when choosing the sphere count. In the technical setup the material simulation was used in, the 1 kHz requirement could be achieved with 201,532 spheres<sup>22,23</sup>.

Additionally, the material simulation already supports material removal by splitting up, shrinking and displacing spheres. The density of each sphere is incorporated into the removal behavior, as the contact density is already calculated at runtime for the previously mentioned force feedback. Ultimately, by using the software library developed by Knopp et al.<sup>24</sup>, the simulation model was utilized for giving force feedback to the user with a KUKA iiwa robot.



**Figure 8.** Example of a volumetric model of the acetabulum filled with spheres, which is used to calculate the force feedback given by the Virtual Reality simulator. The usage of spheres allows a fast computation of the force feedback.

## Discussion

The here proposed approach represents a first attempt to develop a cadaver-based real-time material model which can be implemented into a haptic VR-based THA surgery simulator. For this purpose, an industrial KUKA iiwa robotic arm (KUKA AG, Augsburg, Germany) was deployed as a force feedback device. The aim was to deliver a realistic haptic feeling when virtually reaming the acetabulum. During real THA, the forces applied by the surgeon for the reaming of the hip exceed what can be delivered by standard commercial type haptic devices available on the market at present. We here made attempt to resolve all these issues for a more realistic VR reaming simulation. Another challenge in the development of surgical VR simulators lies in the delivery of realistic force feedback, as it would be felt when manipulating, cutting or milling human tissues. To achieve these goals, it was necessary to measure the biomechanical behavior of the acetabulum when reamed in human cadavers. Industrial robotic arms are capable of delivering much higher forces, as is the case in THA or similar musculo-skeletal surgery. The approach comes with alienating the industrial robot not to just perform standard handling operations but to actively provide a counter force for the force applied by the surgeon via a reaming tool upon it, simulating what is experienced during THA surgery. A simulation model, which implemented the approach described in “[Implementation approach](#)” section is depicted in Fig. 8. Here, spheres have been integrated into the acetabular region to effectively minimize computational time, given that intersections between spheres are the fastest computational method.

Different from other fields in engineering, the test procedures and the underlying test setup for determining the material properties of the reaming were neither defined nor standardized. Therefore, we developed a new standardized testing procedure to assess the biomechanical response of the acetabula when reamed with a real surgical instrument. Instead of multiple angles as it would be the case during surgery, we only used one direction for standardization purposes. This resulted in highly accurate and consistent data of the interactional forces and torques between the surgical instrument and the acetabula. Further, our here presented standardized test procedure could also be applied to other drilling or reaming scenarios where interactional forces and torques between a surgical instrument and tissue need to be measured, e.g. the drilling of bone channels for inserting bone screws. The specific nature of the load and torque data being non-uniform and involving one characteristic local peak at this stage is not well understood and is lacking the morphological link to the here presented mechanical data. Potentially, the diameters of the acetabulum and reaming device, inhomogeneities in cartilage thickness as well as the hip not being a perfect sphere may be causes for the here observed phenomena. This explanation, however, remains unsubstantiated and warrants further study in future projects.

A common approach for developing a material model simulating interactional forces and torques between the reamer and the acetabulum would use FEA. However, to develop such an FEA model, measurements with far more than the here available 24 human acetabula would be necessary to determine the influence of parameters



**Figure 9.** The Virtual Reality acetabulum reaming simulator, where the developed real-time material model is used to simulate the interactional forces and torques between the reamer and the acetabulum.

like age, sex, bone density and thicknesses of the different tissue layers. Additionally, in mechanical engineering, tests aimed at determining material properties to be implemented in FEA are starting from a known and standardized testing procedure and sample geometry. In the given case, the testing procedure had to be developed first and the geometry of each acetabulum is well known to be different inter-individually. However, even if an FEA model simulating the interactional forces and torques between the reamer and the acetabulum existed, it could not be used in a surgical VR training simulator: Haptic feedback needs an update rate of 1 ms, which is usually out of reach of any contemporary FEA simulation, because of the computational load.

For this reason, the here given approach has been to measure directly the interactional forces and torques between the reamer and the acetabulum and deriving an analytical representation as a function of the reamer displacement. However simply using polynomials for interpolating the measured curves would have led to polynomials of high orders consuming considerable computational power. Therefore, the here presented approach uses Cubic Splines to interpolate piecewise the measured forces/torques with an appropriate number of interpolation nodes. Further, piecewise interpolation based on Cubic Splines avoids problems associated with simpler solutions like the Lagrange approach because they avoid Runge's phenomenon.

However, by using Cubic Splines, one can run into computational overload depending on the number of nodes needed for the interpolation. Here an optimization conflict arises between the fitting of the interpolated curve to the measured data (many nodes) and the computational resources it requires (few nodes). This work presented an approach to determine the appropriate number of nodes by analyzing the second derivative of the measured quantities together with the maximum tolerable error between the measured data and the interpolated curves.

Additionally, we compared Cubic Hermite Splines with Cubic Splines to increase the fitting quality with lesser node numbers. The results showed that Cubic Hermite Splines in general provide lower error between the interpolated curve and the measured data than Cubic Splines using the same number of nodes. By using Cubic Hermite Splines, one could further avoid the problem of the deviation of the interpolated curve from the data set as the number of nodes decreases, leading to more accurate results. Even with a decreasing number of interpolation nodes, a better approximation is achieved using Cubic Hermite Splines. This indicates, in order to achieve a defined approximation using the Cubic Hermite Splines, it is possible to use a reduced number of interpolation nodes, requiring lower computational power.

It could further be shown that using the given approach, a valid spectrum of occurring forces and torques can be obtained, which can be used for the simulation of forces and torques during reaming in VR simulation. As the here developed material model bases on Cubic Hermite Splines, the level of machining hardness of the virtually-reamed acetabulum can be freely defined within a given range. These combined efforts will allow the training surgeons to experience different acetabula in the VR acetabulum reaming simulator developed by our group<sup>25,26</sup> (see Fig. 9). However, conclusions on the realism of the haptic behavior cannot be drawn and need to be evaluated in conjunction with the entire VR training experience in a user study with orthopedic surgeons. Such future user study should target advanced students in medical schools or allied disciplines, residents and experienced surgeons, to get a holistic overview of the expectations of all relevant stakeholders. Medical students may provide their anticipated expectations on the training simulator. Residents may express their needs based on their current training experiences with state-of-the-art training methods. Experienced surgeon may best assess the quality of the training simulator and can evaluate the training concept. The user study should measure user experience, usability, and the cognitive work load of the simulator using established post-test questionnaires. Further presence should be measured with short post-test questionnaire as well. As a last quantitative measure, a self-developed post-test questionnaires about the quality and the expected capacity of the simulator for each participant groups should be used. This to be developed questionnaire should base on and adapt questionnaire from literature used to assess other VR surgery trainings simulator, e.g. for minimally-invasive surgeries. An interview with each participant about their training experience should conclude the assessment.

Further, the here obtained data sets could be used as a part of a training data set to develop a deep neural network (DNN) based material model, which was not the aim of the study but presents interesting future work. However, generating a good DNN based material model would require much more data sets than we could acquire. Given that such a DNN based material model would exist, it would be interesting to compare the felt realism of the generated haptic force feedback in a user study.

**Limitations.** A number of limitations exist related to the given work: First, the number of acetabula available for the tests was limited to 24 samples, with a narrow geriatric age range and only one angle of reamer rotation. Though the age range of the cadavers assessed here may not exactly fit with the age range of patients undergoing THA, these first baseline data may well serve as a reference for ranges of material properties of the acetabulum when reamed. With the limited number of acetabula, it was only possible to obtain data on the general spectra of forces and torques occurring during the reaming of human acetabula. For training purposes, it would be desirable to train patient group specific reaming behavior depending on age, sex and bone quality. To achieve this, more data sets are needed. Second, the interactional forces and torques were measured along one (reaming tool) direction. Future work should repeat the given experiments, obtaining mechanical data at different reaming angles similar to the acquisition of a bidirectional reflectance distribution function for the rendering of graphics models to improve the material model. Third, the acetabula were casted in a rigid support, thus, the influence of soft tissues surrounding the acetabulum was not reflected in the measured biomechanical data. A fourth possible limitation is that the Spline interpolation leads to a smooth material behavior. A solution for this would be to reduce the maximum error leading to higher numbers of control point with a better representation of the measured jitter. Alternatively, the presented Cubic Hermite Spline material model could be extended with a noise factor introducing the measured jittering behavior to the material model. A user study determining the felt realism of the presented model is necessary to evaluate if the smoothed behavior of the presented model poses a problem.

### Conclusion

We here presented for the first time a fast computing material model of the human acetabulum, which is applicable for haptic reaming simulators, as is required for VR THA surgery training. A standardized biomechanical testing setup was established to measure the interactional forces and torques during the reaming of acetabula, as is performed during THA. Based on tests with human acetabula under static and dynamic settings, a fast computing Cubic Hermite Splines based material model was developed. It was shown that Cubic Hermite Splines provided a superior approximation of the real forces and torques compared to Cubic Splines. Finally, the implementation of a Cubic Hermite Splines material model combined with the here measured spectra of interactional forces and torques into a simulator will allow surgeons to train with a variety of machining hardness levels of acetabula of haptic VR acetabulum reaming (Supplementary Information 1).

Received: 16 September 2019; Accepted: 3 August 2020

Published online: 03 September 2020

### References

1. Learmonth, I. D., Young, C. & Rorabeck, C. The operation of the century. Total hip replacement. *Lancet* **370**, 1508–1519. [https://doi.org/10.1016/S0140-6736\(07\)60457-7](https://doi.org/10.1016/S0140-6736(07)60457-7) (2007).
2. Grimberg, A., Jansson, V., Liebs, T., Melsheimer, O. & Steinbrück, A. Endoprothesenregister Deutschland (EPRD) - Jahresbericht 2017. Mit Sicherheit mehr Qualität. EPRD Deutsche Endoprothesenregister gGmbH (2018).
3. Singh, J. A., Yu, S., Chen, L. & Cleveland, J. D. Rates of total joint replacement in the united states future projections to 2020–2040 using the national inpatient sample. *J. Rheumatol.* <https://doi.org/10.3899/jrheum.170990> (2019).
4. Neugebauer, R. et al. Experimental modal analysis on fresh-frozen human hemipelvic bones employing a 3D laser vibrometer for the purpose of modal parameter identification. *J. Biomech.* **44**, 1610–1613. <https://doi.org/10.1016/j.jbiomech.2011.03.005> (2011).
5. Hammer, N. et al. Ethanol and formaldehyde fixation irreversibly alter bones' organic matrix. *J. Mech. Behav. Biomed. Mater.* **29**, 252–258. <https://doi.org/10.1016/j.jmbbm.2013.09.008> (2014).
6. Steinke, H. et al. Deformation behavior of the iliotibial tract under different states of fixation. *Med. Eng. Phys.* **34**, 1221–1227. <https://doi.org/10.1016/j.medengphy.2011.12.009> (2012).
7. Kneebone, R. Simulation in surgical training. Educational issues and practical implications. *Med. Educ.* **37**, 267–277. <https://doi.org/10.1046/j.1365-2923.2003.01440.x> (2003).
8. Escobar-Castillejos, D., Noguez, J., Neri, L., Magana, A. & Benes, B. A review of simulators with haptic devices for medical training. *J. Med. Syst.* **40**, 104. <https://doi.org/10.1007/s10916-016-0459-8> (2016).
9. Dawe, S. R. et al. A systematic review of surgical skills transfer after simulation-based training. Laparoscopic cholecystectomy and endoscopy. *Ann. Surg.* **259**, 236–248. <https://doi.org/10.1097/SLA.0000000000000245> (2014).
10. Athanasiou, K. A., Agarwal, A. & Dzida, E. J. Comparative study of the intrinsic mechanical properties of the human acetabular and femoral head cartilage. *J. Orthop. Res.* **12**, 340–349. <https://doi.org/10.1002/jor.1100120306> (1994).
11. Ishiko, T., Naito, M. & Moriyama, S. Tensile properties of the human acetabular labrum—the first report. *J. Orthop. Res.* **23**, 1448–1453. <https://doi.org/10.1016/j.orthres.2004.08.025.1100230630> (2005).
12. Vasu, R., Carter, D. R. & Harris, W. H. Stress distributions in the acetabular region—I. Before and after total joint replacement. *J. Biomech.* **15**, 155–164. [https://doi.org/10.1016/0021-9290\(82\)90247-0](https://doi.org/10.1016/0021-9290(82)90247-0) (1982).
13. Colgate, J. E. & Brown, J. M. Factors affecting the Z-Width of a haptic display. In *Robotics and Automation, '94 International Conference*, 3205–3210 (IEEE Computer Society Press, Los Alamitos, 1994).
14. Love, L. J. & Book, W. J. Contact stability analysis of virtual walls. In *International Mechanical Engineering Congress and Exposition*, edited by Georgia Institute of Technology, 698–694 (1995).
15. Coles, T. R., Meglan, D. & John, N. W. The role of haptics in medical training simulators. A survey of the state of the art. *IEEE Trans. Haptics* **4**, 51–66. <https://doi.org/10.1109/TOH.2010.19> (2011).
16. Colonnese, N. & Okamura, A. Stability and quantization-error analysis of haptic rendering of virtual stiffness and damping. *Int. J. Robot. Res.* **35**, 1103–1120. <https://doi.org/10.1177/0278364915596234> (2015).
17. Möbius, R. et al. A pilot trial comparing the tear-out behavior in screw-sockets and cemented polyethylene acetabular components—A cadaveric study. *Orthop. Traumatol. Surg. Res.* **102**, 723–728. <https://doi.org/10.1016/j.otsr.2016.06.007> (2016).
18. Epperson, J. F. On the Runge example. *Am. Math. Mon.* **94**, 329–341. <https://doi.org/10.1080/00029890.1987.12000642> (2018).
19. Erdogan, K. Spline interpolation techniques. *J. Tech. Sci. Technol.* **2**, 47–52 (2014).
20. de Boor, C. A *practical guide to splines* 5th edn. (Springer, New York, 1991).

21. Weller, R. & Zachmann, G. A unified approach for physically-based simulations and haptic rendering. In *Sandbox 2009. Proceedings; 4th ACM SIGGRAPH Symposium on Video Games; New Orleans, Louisiana, August 4–6, 2009*, (eds Davidson, D.) 151 (ACM, New York, 2009).
22. Kaluschke, M. *et al.* Realistic haptic feedback for material removal in medical simulations. In *2020 IEEE Haptics Symposium (HAPTICS)* 920–926 (IEEE, New York, 2020).
23. Kaluschke, M., Weller, R., Zachmann, G. & Lorenz, M. A continuous material cutting model with haptic feedback for medical simulations. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* 1002–1003 (IEEE, New York, 2019).
24. Knopp, S., Lorenz, M., Pelliccia, L. & Klimant, P. Using industrial robots as haptic devices for VR-training. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* 607–608 (IEEE, New York, 2018).
25. Kaluschke, M. *et al.* HIPS—A virtual reality hip prosthesis implantation simulator. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* 591–592 (IEEE, New York, 2018).
26. Kaluschke, M. *et al.* A virtual hip replacement surgery simulator with realistic haptic feedback. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* 759–760 (IEEE2018).

### Acknowledgements

This work was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) under the grant KF2039721AK2. We also wish to thank the body donors who, while alive, consented to donate their bodies for research purposes and their relatives for supporting this decision. Robbie McPhee provided the illustrations. Aqeeda Singh proofread the article as a native speaker. Niels Hammer was granted a visiting scientist stipend by the Chemnitz University of Technology, Germany. We acknowledge support from the German Research Foundation (DFG) and Universität Leipzig within the program of Open Access Publishing.

### Author contributions

L.P., M.L., M.W., S.S. and N.H. created the experimental design and conducted the experiments. M.L. and L.P. developed the algorithm for automatically determining the appropriate number of spline interpolation nodes. P.K., C.R. and G.Z. critically reviewed the manuscript prior to submission. S.K., R.W. and M.K. developed the implementation approach. C.H. and D.Z. contributed to the clinical perspective of the paper. All authors approved the final version of the paper.

### Funding

Open Access funding provided by Projekt DEAL.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s41598-020-71499-5>.

**Correspondence** and requests for materials should be addressed to N.H.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020



---

### IMMERSIVE ANATOMY ATLAS: LEARNING FACTUAL MEDICAL KNOWLEDGE IN A VIRTUAL REALITY ENVIRONMENT

#### *Outline*

---

1	<b>INTRODUCTION</b> . . . . .	<b>203</b>
2	<b>MATERIALS AND METHODS</b> . . . . .	<b>204</b>
	2.1 Immersive ANatomy Atlas . . . . .	204
	2.2 Study Design . . . . .	205
	2.3 Participants . . . . .	206
	2.4 Statistical Analysis . . . . .	207
3	<b>RESULTS</b> . . . . .	<b>208</b>
4	<b>DISCUSSION</b> . . . . .	<b>208</b>
	4.1 Limitations of the Study . . . . .	210
5	<b>CONCLUSIONS</b> . . . . .	<b>210</b>
	<b>ACKNOWLEDGMENTS</b> . . . . .	<b>210</b>
	<b>NOTES ON CONTRIBUTORS</b> . . . . .	<b>210</b>

---

#### *Bibliography Entry*

Gloy, K., Weyhe, P., Nerenz, E., Kaluschke, M., Uslar, V., Zachmann, G., and Weyhe, D. (2022). **Immersive Anatomy Atlas: Learning Factual Medical Knowledge in a Virtual Reality Environment.** *Anatomical Sciences Education*, 15(2):360–368

#### *Author's Contribution*

The author wrote the Objectives, helped in the literature research and in writing the rest of the manuscript. The diagram was also generated by the author.

#### *Copyright*

Copyright © 2021 The Authors. *Anatomical Sciences Education* published by Wiley Periodicals LLC on behalf of American Association for Anatomy.

# Immersive Anatomy Atlas: Learning Factual Medical Knowledge in a Virtual Reality Environment

Kilian Gloy,<sup>1</sup> Paul Weyhe,<sup>1</sup> Eric Nerenz,<sup>1</sup> Maximilian Kaluschke,<sup>2</sup> Verena Uslar,<sup>1</sup> Gabriel Zachmann,<sup>2</sup> Dirk Weyhe<sup>1</sup>

<sup>1</sup>Department for Visceral Surgery, Carl von Ossietzky University of Oldenburg, Oldenburg, Germany

<sup>2</sup>Institute for Computer Graphics and Virtual Reality, University of Bremen, Bremen, Germany

In order to improve learning efficiency and memory retention in medical teaching, furthering active learning seems to be an effective alternative to classical teaching. One option to make active exploration of the subject matter possible is the use of virtual reality (VR) technology. The authors developed an immersive anatomy atlas which allows users to explore human anatomical structures interactively through virtual dissection. Thirty-two senior-class students from two German high schools with no prior formal medical training were separated into two groups and tasked with answering an anatomical questionnaire. One group used traditional anatomical textbooks and the other used the immersive virtual reality atlas. The time needed to answer the questions was measured. Several weeks later, the participants answered a similar questionnaire with different anatomical questions in order to test memory retention. The VR group took significantly less time to answer the questionnaire, and participants from the VR group had significantly better results over both tests. Based on the results of this study, VR learning seems to be more efficient and to have better long-term effects for the study of anatomy. The reason for that could lie in the VR environment's high immersion, and the possibility to freely and interactively explore a realistic representation of human anatomy. Immersive VR technology offers many possibilities for medical teaching and training, especially as a support for cadaver dissection courses. *Anat Sci Educ* 15: 360–368. © 2021 The Authors. Anatomical Sciences Education published by Wiley Periodicals LLC on behalf of American Association for Anatomy.

**Key words:** gross anatomy education; anatomical atlas; virtual reality; immersive VR; active learning; knowledge retention

## INTRODUCTION

Medical education not only requires the trainees to acquire a number of practical skills, but also to learn large amounts of basic factual information. This circumstance makes efficient

learning and accurate retention imperative (Yeh and Park, 2015). Many different methods have been explored in order to improve learning efficiency (Yeh and Park, 2015). One type of learning that is consistently connected with improved learning and memory results is active learning (Hazlett, 2009; Kornell et al., 2009; Markant et al., 2016). There are several different definitions of active learning, from different fields of study; Common aspects of those definitions include: “some combination of increased physical activity or interaction, deeper processing, elaboration or explanation of material, planning of learning activities, question asking, metacognitive monitoring, and social collaboration” (Markant et al., 2016). These criteria tie well into the psychological theory of constructivist learning which assumes learning to be an active process and learners to be actively seeking knowledge. In complex interaction with the material, the learners generate knowledge and deeper understanding (Fosnot and Perry, 1996; Siemens, 2005).

In recent years, the use of virtual reality (VR) technology has been considered as a useful tool in education (Markant et al., 2016). In line with the ideas of active and constructivist

\*Correspondence to: Dr. Kilian Gloy, Abteilung für Viszeralchirurgie, Carl von Ossietzky University of Oldenburg, Grüne Str. 6, 26121 Oldenburg, Germany. E-mail: kilian.gloy@uni-oldenburg.de

Additional supporting information can be viewed in the online version of this article.

Received 13 July 2020; Revised 16 April 2021; Accepted 19 April 2021.

Published online 30 June 2021 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/ase.2095

© 2021 The Authors. *Anatomical Sciences Education* published by Wiley Periodicals LLC on behalf of American Association for Anatomy

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

learning, VR could be used to increase the physical interaction with the subject matter and allow learning in an explorative context more similar to real-life conditions. The additional control, which learners would have over the experience, can be expected to improve learning (Gureckis and Markant, 2012), and the digital environment could also allow for gamification in order to increase learner motivation (Koivisto and Hamari, 2019). It has already been shown that VR has a positive impact on learning compared to conveying information via desktop personal computer (Selzer et al., 2019), likely due to the increased immersion which aids in information recall (Krokos et al., 2019).

The immersion VR provides is highest for the so-called “immersive VR”, in which a user can interact with a computer-generated three-dimensional (3D) environment as if they were physically present in that environment (Freina and Ott, 2015; Zackoff et al., 2019). Immersive VR is associated with higher ratings for interest and motivation in students (Parong and Mayer, 2018). Despite the advantages of immersive VR, there are two potentially negative aspects: cybersickness and high cognitive load.

Cybersickness is a phenomenon common in interaction with virtual environments, especially VR, and consists of a multitude of physiological symptoms similar to car- or seasickness (Brewer-Deluce et al., 2021). The cause of this unpleasant experience is unclear, but likely related to sensory mismatch (Yildirim, 2019). However, studies which used immersive VR in surgical training have reported little to no problems with cybersickness in their participants (Huber et al., 2017; Frederiksen et al., 2020). Frederiksen and colleagues argued that this may be because of the limited head movements in this setting compared to average VR games (Frederiksen et al., 2020).

The cognitive load of a learning task is commonly divided into at least two types: the intrinsic load, inherent to the task or information that must be learned, and the extraneous load, generated by external processes or information that distract from the learning material (Wong et al., 2012). There is evidence that, in laparoscopic surgical training, immersive VR has increased cognitive load and an associated worsened task performance (Frederiksen et al., 2020). The complete implications of these circumstances have yet to be discussed; For example, it was argued that training under increased cognitive load may actually be beneficial since it improves the transfer of training into a real situation with a strong cognitive load (Sankaranarayanan et al., 2020).

Within the medical field, VR technology has been shown to be a helpful tool for teaching procedural skills (Bracq et al., 2019), since the proficiency acquired there can transfer to the real-world clinical setting (Seymour, 2008). So far, it seems that conventional VR outperforms immersive VR in this context (Frederiksen et al., 2020). Medical trainees have to acquire both procedural skills and factual knowledge. There have been a few research studies into the effectiveness of immersive VR for the teaching of anatomy, but a clear result has yet to emerge. For example, Stepan and colleagues found that VR provided a more enjoyable learning experience than textbooks without actually increasing the learning benefit (Stepan et al., 2017), whereas Kurul and colleagues found a significantly positive learning effect of VR compared to attending a presentation on the material (Kurul et al., 2020). A study by Birbara and colleagues found that the learning preferences differed between different groups of participants (tutors vs. students), although immersive VR was seen as more mentally taxing than a desktop

version of the same program, and more strongly connected to physical discomfort (Birbara et al., 2020). Lastly, Zinchenko and colleagues found immersive VR to be most beneficial for learning previously unknown information when compared to books and a 3D desktop application (Zinchenko et al., 2020).

In summary, the current state of research on immersive VR as a tool for learning human anatomy is ambiguous. Further research in different populations and with different methods is necessary to gather more empirical data and piece together the whole picture. Seeing as anatomy study through cadaver dissection has many advantages which neither textbooks nor VR applications can recreate (Dua et al., 2021), it is unlikely that VR will replace this traditional learning method. However, if immersive VR is shown to be effective in learning and retaining anatomy knowledge, it might become a meaningful support in anatomy courses.

The immersive, interactive 3D anatomy atlas used in this study was developed at the VR laboratory at the University of Bremen. The atlas features a virtual operating theater and allows the user to actively explore anatomical structures and arrangements of the human body through virtual dissection. A previous pilot study using an older version of the same atlas has already shown that information acquisition was faster when novices to the study of anatomy used the VR atlas compared to retrieving the information from books (Weyhe et al., 2018). As a second step, the aim of this study was the examination of long-term knowledge retention in novices by measuring the amount of correct answers they give after working with the immersive anatomy atlas (VR condition) in comparison to using only anatomical books (open book condition; OB) utilizing a randomized study design. The ratio of correct answers to a second questionnaire conducted several weeks later operationalized the information retention rate in both groups.

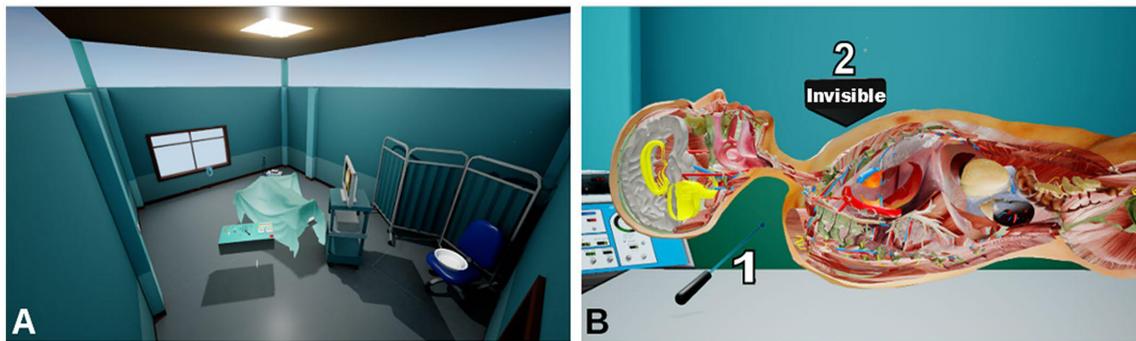
In line with the literature presented above and the aim of this study, the hypotheses were as follows: (1) Acquiring new information with the VR atlas is faster than using standard printed anatomical atlases [replication of effect from (Weyhe et al., 2018)]; and (2) Working with the VR atlas leads to an improved retention of knowledge compared to working with standard anatomical atlases.

## MATERIALS AND METHODS

This study was approved by the medical ethics committee of the Carl von Ossietzky Universität Oldenburg, Germany (ID-number: 2020-065).

### Immersive Anatomy Atlas

The immersive anatomy atlas is an application developed by the authors that uses a head-mounted display (HMD) to immerse the user in a virtual operation room (see Fig. 1). Using head tracking, users have a full 360 view, that is, they can look around and move within the virtual room. Additionally, there are bi-manual controllers which enable the user to interact with virtual reality. They can manipulate individual organs by grabbing them with their virtual hands, which are controlled by the controllers. Several virtual tools are placed on a nearby table. Some of these tools mimic realistic surgery tools, some allow for more “magic” tasks, such as exploring the anatomy by controlling the model’s transparency. Others allow to hide anatomy in spherical areas around a pointing tool (see Fig. 1) or place a cross section to hide all organs in front of it (see Fig. 1).



**Figure 1.**

Immersive Anatomy Atlas. A, Overview of the 3D environment that comprises the virtual operating room. The camera is positioned in one of the rooms upper corners; B, A cross-section of the anatomical model, generated with the use of virtual tools. The pointing baton (1) makes a spherical area around its tip transparent. The cross-sectional tool (2) makes everything on the front side of the imaginary plane which passes through the tool transparent. The combination of these tools enables the user to inspect any part of the model.

During the learning phase, participants can explore and study the immersive anatomy atlas by inspecting it from every point of view, by grabbing organs and other structures and inspecting them from all angles, then replacing them in the original position. When organs are placed back, they snap to their original place, as long as the release pose (position and orientation) is close to their original, correct pose. Thus, the anatomical model is always correct, unless deliberately altered by the user.

While anatomical structures are held in hand, further information about them can be viewed by the user. Furthermore, the complete anatomical model can be reset at once to its original state using a virtual button.

An introductory video for the anatomy atlas used in this study can be viewed online (Pius-Hospital Oldenburg, 2018).

The immersive anatomy atlas system was implemented by the authors on top of the game engine Unreal Engine, version 4.23 (Epic Games, Inc. Cary, NC) using the built-in programming language Blueprint. The 3D geometrical models were created by a 3D artist and purchased by the authors. All the anatomical parts are designed to closely resemble real anatomy. The geometry was further modified by the authors through Blender software, version 2.92.0 (Blender Foundation, Amsterdam, The Netherlands). Coordinate origins were moved, geometry groups were separated, textures were changed, and missing organs were added. Everything is rendered from 3D geometrical models of the anatomical structures at runtime in real time.

The models are loaded at runtime by the immersive anatomy atlas, then rendered stereoscopically by the game engine and displayed in stereo on the head-mounted display (HMD), thus providing stereoscopically correct images to the user. During the study, a head-mounted display HTC VIVE™ (High Tech Computer Corp., New Taipei City, Taiwan) with the resolution of 1080 by 1200 pixel per eye was used to display the immersive anatomy atlas to the users in stereo vision. The frame rate of the immersive anatomy atlas was sustained at 90 frames per second, to allow for the illusion of presence in the virtual reality.

All the virtual tools (both surgery and “magic” ones) were implemented by the authors.

## Study Design

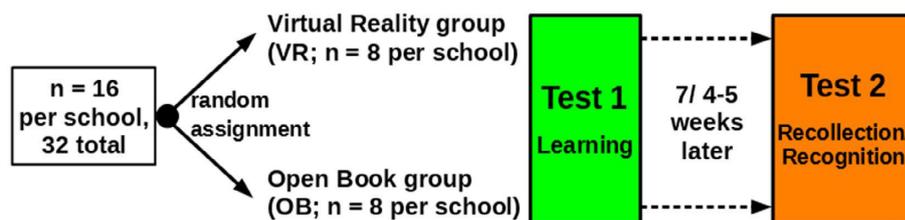
The study was conducted at two separate German high schools, referred to as school A and school B below. In each school, 16 participants were recruited and randomly assigned into equally sized groups for two different learning modalities: open book (OB condition) and virtual reality (VR condition) learning (per school: OB:  $n = 8$ , VR:  $n = 8$ ). High school students were chosen to ensure that the participants would have no prior formal anatomical training and would approach the learning content as novices.

A schematic overview of the experimental design can be found in Figure 2.

The VR-group used the immersive anatomy atlas. They viewed a short introductory video for the atlas before the experiment started. They had the opportunity to familiarize themselves with the VR environment for a maximum of 5 minutes and clear up any questions regarding the handling of the VR interface.

The OB-group used standard anatomy atlases (Paulsen and Waschke, 2017a,b,c). The participants in both groups were presented with the same set of nine single-choice questions on paper, encompassing the topics topography, cardiovascular system, and nervous system (see Supplemental Material File 1 for a list of the questions). They were tasked to answer the questions correctly and as quickly as possible, using only the respective method of learning at their disposal (OB or VR). The time which elapsed between the question and the participants’ answer was recorded (response time), and the percentage of correct questions constituted the test score. Because of this experimental setup, the questions of this test can’t be classified according to Bloom’s revised taxonomy (Krathwohl, 2002). As soon as all questions had been answered, either correctly or incorrectly, Test 1 was over (see Fig. 3 for an impression of the experimental set up in Test 1).

After seven weeks in school A and four to five weeks in school B, the participants were tested for their long-term memory of the topics they learned during Test 1. Each participant answered a second list of nine multiple-choice questions from the same three topics as before (see also Supplemental Material File 1). Participants had to answer the questions without any help, based only on their memory. The



**Figure 2.**

Schematic overview of the experimental design. Students in both schools were randomly assigned to either VR or OB groups. Afterward, they completed the first test (green) with their respective learning method. Following a seven-week waiting period in school A or a four- to five-week period in school B, the students completed the second test (orange) from memory. VR, virtual reality; OB, open book.

response time was not measured; The sole point of interest was whether the recall of the factual knowledge was successful or not, as reflected in correct or incorrect answers. All questions in this test belong to the category “Remember”, according to Bloom’s revised taxonomy, since recognizing and recalling information from long-term memory was necessary (Krathwohl, 2002).

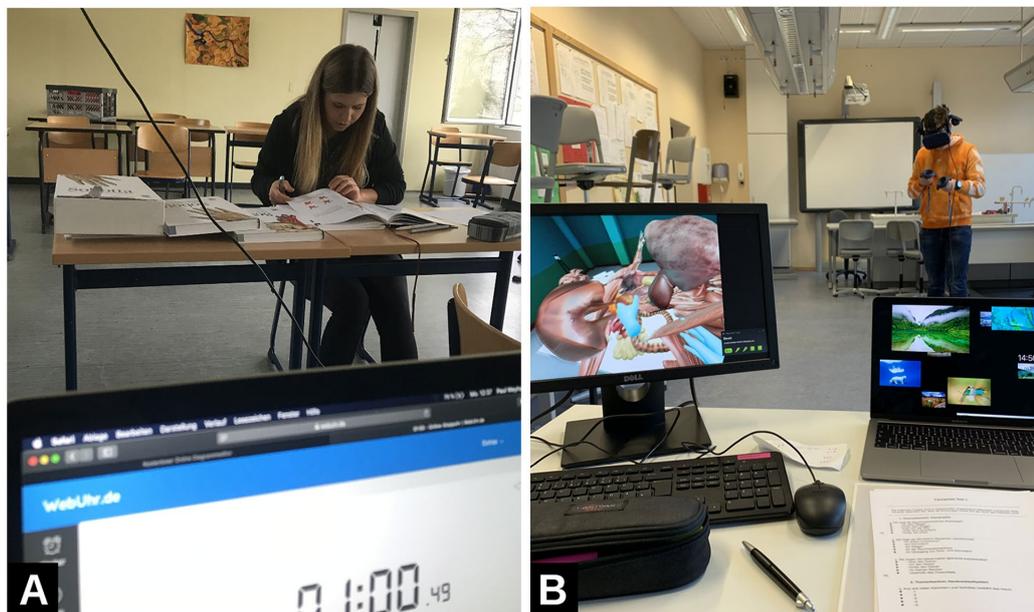
The anatomy tests (Test 1 and Test 2) were developed in-house, by medical experts, to ensure that each question can be answered with the given material and is of appropriate difficulty level for high school students. Since the tests were developed specifically for this study and were not tested elsewhere, no statements can be made about the tests’ reliability or validity.

### Participants

Thirty-two eleventh-grade high school students participated in the experiment on a voluntary basis, 16 from each of the two high schools (overall  $n = 32$ ). Human anatomy was not part of the senior-classes’ biology curriculum, meaning that the courses the participants took during that time had no influence on their preexisting knowledge of anatomy.

In school A, the average age  $\pm$  SD was  $16.5 \pm 0.52$  years (minimum = 16, maximum = 17.). The OB group consisted of three female and five male students, and the VR group of five female and five male students.

In school B, the average age  $\pm$  SD was  $17.6 \pm 0.5$  years (minimum = 17, maximum = 18). The OB group consisted of three



**Figure 3.**

Impressions of the experimental setup for Test 1. A, The open book condition; A student is using the textbooks supplied to them to answer the questionnaire. A timer can be seen at the lower edge of the panel, measuring response times. B, The virtual reality condition; A student wearing the head-mounted display (HMD) is using the immersive anatomy atlas to answer the questionnaire. A space has been cleared around them to allow for free movement in the virtual reality environment. At the left edge of the panel, the monitor shows what the student is currently seeing in their display.

**Table 1.**Results of the *Post Hoc* t-tests for the Mixed ANOVA, for the Interaction Effect of SCHOOL and TIME

School	Test Number	School A		School B	
		Test 1 <i>P</i> -value <sup>a</sup>	Test 2 <i>P</i> -value <sup>a</sup>	Test 3 <i>P</i> -value <sup>a</sup>	Test 4 <i>P</i> -value <sup>a</sup>
School A	Test 1				
	Test 2	<0.001 <sup>b</sup>			
School B	Test 3	0.554	<0.001 <sup>b</sup>		
	Test 4	<0.001 <sup>b</sup>	<0.041 <sup>b</sup>	<0.001 <sup>b</sup>	

<sup>a</sup>Benjamini–Hochberg corrected *P*-values; <sup>b</sup>Indicates statistically significant results. Empty cells were either nonexistent values or repetitions.

female and five male students, the VR group of four female and four male students.

There were no significant differences in the distribution of gender or age between the learning groups, in neither school, as confirmed with Fischer's exact test and t-test, respectively.

No participant was familiar with the head-mounted display or the immersive anatomy atlas before participating in this study. All participants received a letter detailing the contents of the experiment. Only students who handed in the letter signed by their legal guardians were allowed to participate.

### Statistical Analysis

In order to investigate hypothesis 1, a replication of the response time effect found in the preceding study on the immersive anatomy atlas (Weyhe et al., 2018), the response time data from Test 1 in school B were used. School A had to be excluded from this analysis because an error during the experiment led to a loss of the relevant data. The response times were averaged over the questions and then grouped by learning method. The distributions of average response times in both learning groups were tested for normal distribution using the Shapiro–Wilk test ( $\alpha = 0.1$ ), because this test works well for small samples (see Field et al., 2012). Since the assumption of normality was violated, a two-sided Wilcoxon rank sum test had to be used instead of the parametric t-test (Field et al., 2012) to test for an effect of the learning method on the response time.

To test hypothesis 2, the assumption of improved knowledge retention for those working with the immersive anatomy atlas, the test performance was compared between experimental groups, time points, and schools. The performance in the nine individual questions was summarized into one variable that represented the percentage of correctly answered questions per participant and per test. A mixed ANOVA was calculated for the percentage of correct answers with the factors METHOD (VR or OB, between factor), SCHOOL (school A or school B, between factor), and TIME (Test 1 or Test 2, within factor). This statistical test was chosen because three factors had a potential influence on the dependent variable, making an ANOVA necessary in order to make comparisons of means. The assumptions of normality, homoscedasticity, and sphericity were tested beforehand using Shapiro–Wilk tests and Levene tests ( $\alpha = 0.1$ ). The assumption of normality was violated in three of the eight groups, but since the ANOVA is a robust procedure as long as the group sizes are equal (Field et al., 2012) and all other assumptions were fulfilled, the parametric ANOVA was used.

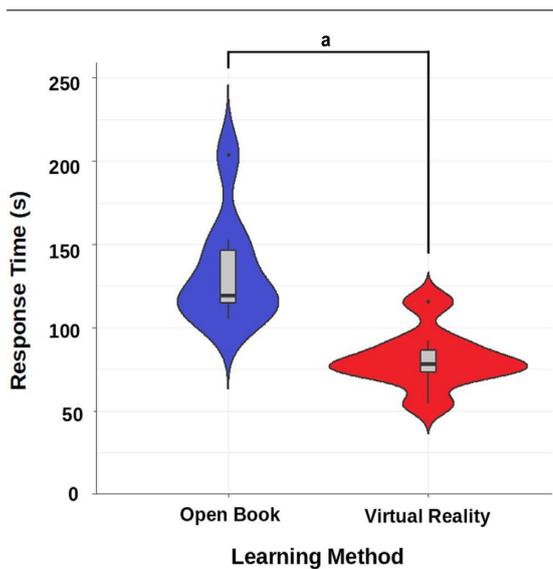
To check the difficulty of the single-choice questions, the difficulty index was calculated for Test 1 and Test 2. Additionally, for an investigation of the individual questions, the performance of all participants was averaged for each of the nine questions, separately for Test 1/Test 2 and VR/OB. The resulting percentage of correct answers per question was then visualized in a bar chart.

All statistical analysis was performed using R statistical software, version 3.6.3 (R Foundation for Statistical Computing, Vienna, Austria).

**Table 2.**Results of the *Post Hoc* t-tests for the Mixed ANOVA, for the Interaction Effect of METHOD (Virtual Reality vs. Open Book) and TIME

Method	Test Number	Virtual Reality		Open Book	
		Test 1 <i>P</i> -value <sup>a</sup>	Test 2 <i>P</i> -value <sup>a</sup>	Test 3 <i>P</i> -value <sup>a</sup>	Test 4 <i>P</i> -value <sup>a</sup>
Virtual Reality	Test 1				
	Test 2	<0.001 <sup>b</sup>			
Open Book	Test 3	0.175	<0.001 <sup>b</sup>		
	Test 4	<0.001 <sup>b</sup>	<0.01 <sup>b</sup>	<0.001 <sup>b</sup>	

<sup>a</sup>Benjamini–Hochberg corrected *P*-values; <sup>b</sup>Indicated significant results. Empty cells were either nonexistent values or repetitions.



**Figure 4.**

Response time (in seconds) of Test 1 in school B, grouped by learning method. The colored outlines depict the distribution of the data; therefore, the width of the outline at any point  $Y$  increases with the number of participants whose response time results equal  $Y$ , or lie close to it. The box plot within shows the quartiles 1, 3, and the median. The black dots denote outliers. <sup>a</sup>The difference between the two methods is significant ( $P < 0.001$ ).

## RESULTS

There was a significant difference between the response times in the OB and the VR learning conditions, as shown by the Wilcoxon rank sum test (median difference = 41.3,  $P < 0.001$ ,  $d = 2.01$ ). It took the participants significantly longer to answer the questions in the OB condition (median = 119.5 seconds) than in the VR condition (median = 78.3 seconds). This difference represents a large effect. The results of the response time analysis are visualized in Figure 4.

There were several significant effects on the performance of the students, operationalized by the percentage of correct answers. The mixed ANOVA showed a main effect of METHOD [ $F_{(1,28)} = 9.15$ ,  $P < 0.01$ , partial  $\eta^2 = 0.353$ ]. Participants using the VR atlas achieved better results (mean  $\pm$  SD of correct answers = 73  $\pm$  22%) than those working with books (mean  $\pm$  SD of correct answers = 60  $\pm$  25%). The associated effect size represents a large effect.

There was also a main effect of TIME [ $F_{(1,28)} = 113.89$ ,  $P < 0.001$ , partial  $\eta^2 = 0.803$ ], and a significant interaction effect SCHOOL and TIME [ $F_{(1,28)} = 6.98$ ,  $P < 0.05$ , partial  $\eta^2 = 0.199$ ]. For a visualization of the mixed ANOVA results, see Figure 5.

In order to investigate the interaction effect of SCHOOL and TIME, the mean percentages of correct answers per school were calculated for Test 1 (school A: 85%  $\pm$  13%, school B: 82%  $\pm$  15%) and Test 2 (school A: 42%  $\pm$  16%, school B: 56%  $\pm$  21%). Additionally, post hoc t-tests were performed for this interaction effect (Benjamini–Hochberg correction, see Table 1) and, for explorative purposes, the interaction of METHOD and TIME (Benjamini–Hochberg correction, see Table 2).

Test 2 was overall more difficult than Test 1. The difficulty index (in %) for Test 1 was 83.68 ( $\pm$ 13.82), with a range of 68.75–100, and for Test 2 it was 48.61 ( $\pm$ 19.71), with a range of 9.38–100.

## DISCUSSION

The response time effect postulated in hypothesis 1, which was also found in the previous study on the immersive anatomy atlas (Weyhe et al., 2018), was replicated in this study. This was indicated by the significant Wilcoxon rank sum test, in combination with the higher median response time in the OB group compared to the VR group. It follows that acquiring previously unknown information was faster in the VR condition; this confirms hypothesis 1. The reason for this could be the interactive way of retrieving information from the immersive anatomy atlas, which leads to an easier access to factual anatomical knowledge.

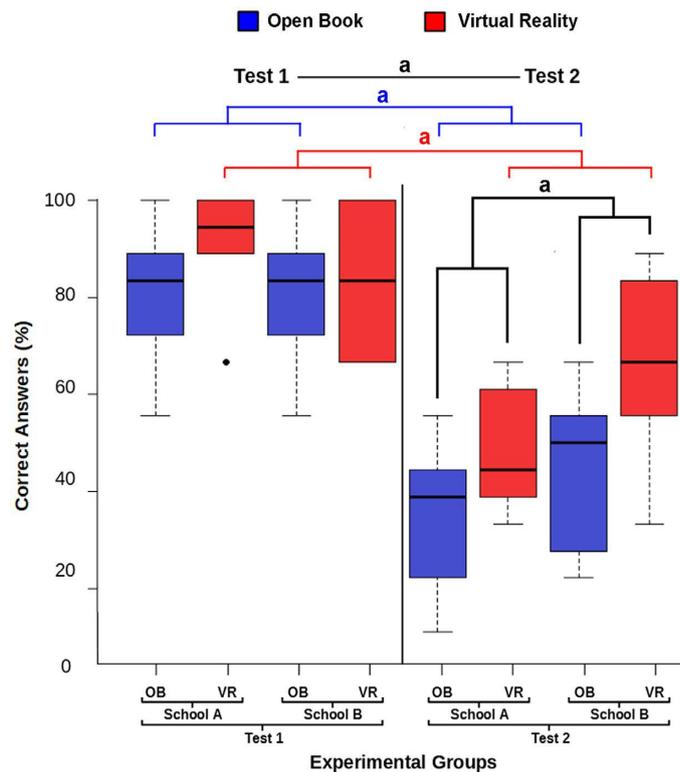
The main goal of this study was the investigation of long-term effects of learning through immersive VR. The short-term benefits were already well documented, while little could be said about retention of knowledge over a longer period of time. Now, this study adds the results of the mixed ANOVA on the knowledge-test performance to the relevant empirical evidence. Two main effects of the factors METHOD and TIME were revealed in the ANOVA.

The effect of TIME simply represents the difference between acquiring the information directly and recalling it several weeks later. It is, therefore, no surprise that the percentage of correct answers was higher during Test 1 (mean  $\pm$  SD = 84  $\pm$  14%) than during Test 2 (mean  $\pm$  SD = 49  $\pm$  20%).

The main effect of METHOD shows that the participants learning with the VR atlas achieved better results than those learning with books. The post hoc t-tests showed that, more specifically, the results of the OB and VR groups were significantly different in the second test, not in the first. The improvement in test results can thus be attributed to better memory retention in the VR group; this confirms hypothesis 2.

Taken together with the response time effect described above, this study has shown that, under the given conditions, the VR atlas both enabled faster information acquisition and facilitated improved memory retention. This makes the immersive anatomy atlas an overall more efficient tool for learning anatomical knowledge than classical learning through books. Combining the active learning and exploration already possible in the VR atlas with additional methods like tests and gamification, which can be added to VR comparatively easily, might enhance the performance of VR learning even further. Additionally, the constructivist learning aspects already present in VR environments could be strengthened with further technological additions. Presently, the virtual atlas allows for self-guided exploration in a relevant and realistic environment, and enables the learners to take ownership of their learning (see for aspects of constructivist learning; Amarin and Ghishan, 2013; Johnson-Glenberg, 2018). Future developments of the VR-atlas may allow multiple people to enter the same simulation, adding social interaction and collaboration (Amarin and Ghishan, 2013), and may add the option for users to construct the virtual environments themselves, which improves learning success especially for low-performance students (Winn et al., 1997).

In summary, the immersive anatomy atlas already seems more efficient than classical learning modalities, and future



**Figure 5.**

Box plots depicting the percentage of correct answers for schools A and B, in Test 1 (first test, learning) and Test 2 (second test, weeks later, recognition/ recall), for both OB and VR. The black dot denotes an outlier. There was a significant main effect of TIME (Test 1 vs. Test 2), a significant main effect of learning METHOD (OB vs. VR), and a significant interaction effect of TIME and SCHOOL (School A, Test 2 vs. School B, Test 2). OB, open book; VR, virtual reality. <sup>a</sup>Significant differences between groups ( $P < 0.01$ ).

developments of VR in general and this software in particular, are expected to increase this advantage.

The significant interaction effect between SCHOOL and TIME is an ordinal effect for TIME; The mean percentage of correct answers is consistently lower in Test 2 than in Test 1. This means that the global main effect of TIME reported above is unaffected by the interaction effect.

The associated post hoc t-tests produced five significant differences. Four of those, however, contained the TIME effect and thus offer no new insight. The last was the significant difference between (school A, Test 2) and (school B, Test 2). Apparently, the retention of anatomical knowledge was overall better in school B (mean  $\pm$  SD = 55  $\pm$  21%) than in school A (mean  $\pm$  SD = 42  $\pm$  16%). However, the reason for that effect could also be the different extent of time between the tests in the two schools.

The anatomy tests employed in this study (see Supplemental Material File) to assess the learned and retained knowledge seemed to have performed well enough, but could be improved in future studies. The questions used in Test 1 were of comparable difficulty; The difficulty index for this test was 83.68 ( $\pm$ 13.82), with a range of 68.75%–100%. Question 1 seemed to suffer from a ceiling effect as participants from both conditions were able to answer it with average correctness of 100%. This test fulfilled its core role to teach the relevant knowledge

and give room to engage with the teaching material, but was sub-optimal in differentiating high- and low-performing students. Test 2 was overall more difficult, which is of course due to the time elapsed between learning and recall. Aside from that, the variability in difficulty between questions was also larger than in Test 1, with a difficulty index of 48.61 ( $\pm$ 19.71) and a range of 9.38–100. Questions 1 and 6 seemed to have been too difficult. Future studies should substitute them with easier alternatives and use question 5, the easiest, at the start of the test.

The participants in this study had no previous knowledge in the field of medicine, which makes the generalization to more experienced medical personnel difficult. Previous literature for a population of experienced medical students had suggested a lack of advantages of immersive VR (Stepan et al., 2017). However, this study has shown the usefulness of VR environments for the initial acquisition of anatomical knowledge, which is in accordance with the findings of Zinchenko and colleagues (Zinchenko et al., 2020).

Virtual reality seems on its way to becoming an integral part of education and training for medical vocations (Rizzetto et al., 2020), and this study supports this direction. Future research should focus on the use of tools like the immersive anatomy atlas for medical trainees or students, especially in the early stages of teaching. If the VR application has a

high physical fidelity, inexperienced students may gain more from its use (Birbara and Pather, 2021), and this learning experience could prepare students for the eventual cadaver dissection courses. A similar statement about the usefulness of VR early in teaching has been made by Andersen and colleagues regarding the acquisition of surgical skills, after they found that for novices cognitive load is higher during cadaver training than VR training (Andersen et al., 2016). In general, supplementing cadaver courses with VR applications for preparation and repetition makes sense, given some of the problems cadaver studies face, like high financial expenses, limited availability, or high student to cadaver ratios (Wainman et al., 2021).

### Limitations of the Study

Some limitations should be considered in regard to this study and its results.

The two schools had different time intervals between the initial learning and the test for knowledge retention. Any difference between the schools could thus be attributed to that discrepancy. Therefore, any further in-depth comparison of the schools and how their curricula may have affected the results became impossible.

The questions used to test the participants' anatomical knowledge were not from a standardized questionnaire. Instead, they were specifically created for this study. This had the advantage of being a perfect fit for the purpose of the experiment, but the disadvantage was that the tests lacked established values for quality criteria such as reliability and validity.

This study compared immersive VR to anatomy textbooks; There are, however, other ways to learn human anatomy. Especially physical models, which have been shown to be superior to VR in some contexts (Wainman et al., 2020), could have been included as a third group to provide a more complete overview in this article.

Lastly, it has to be noted that the sample size was small; with only eight participants per group per school, the results have to be viewed with caution.

### CONCLUSIONS

Based on the results of this study, immersive VR learning seems to be more efficient and to facilitate better long-term retention of knowledge in previously inexperienced students. The reason for that could lie in the VR environment's high immersion and the possibility to freely explore a realistic replication of human anatomy. There are many possibilities for medical teaching and training which VR technology offers, the extent of which might grow with the advancements of the hard- and software. A future challenge for anatomical educational research will be establishing a meaningful standard for a curriculum which combines immersive VR, classic textbooks, and cadaver training.

### ACKNOWLEDGMENTS

The authors thank the students of high schools A and B for participating in the experiment and thereby supporting this research. The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

### NOTES ON CONTRIBUTORS

**KILIAN GLOY**, Dr. Rer. Nat., is a research fellow at the Department for Visceral Surgery at the University Clinic Oldenburg, Carl von Ossietzky University Oldenburg in Oldenburg, Germany. His research interest is in human learning and decision making, as well as the neuropsychological foundations thereof.

**PAUL WEYHE**, is a twelfth-grade high school student who worked as a high-school research assistant in cooperation with the Department for Visceral Surgery at the University Clinic Oldenburg, Carl von Ossietzky University Oldenburg in Oldenburg, Germany. He has a keen interest in science.

**ERIC NERENZ**, is a twelfth-grade high school student who worked as a high-school research assistant in cooperation with the Department for Visceral Surgery at the University Clinic Oldenburg, Carl von Ossietzky University Oldenburg in Oldenburg, Germany. He has a keen interest in science.

**MAXIMILIAN KALUSCHKE**, M.Sc., is a researcher at the faculty for Mathematics and Computer Science at the University of Bremen in Bremen, Germany. His research interests are collision detection, haptic rendering, and physically based simulation.

**VERENA USLAR**, Dr. Rer. Nat., is a senior research scientist in the Department for Visceral Surgery, at University Hospital for Visceral Surgery, the Carl von Ossietzky University Oldenburg in Oldenburg, Germany. She teaches all aspects of good scientific practice to medical students and clinicians and her research interest is in the use of new technologies in the perioperative setting.

**GABRIEL ZACHMANN**, Ph.D., is a professor for computer graphics and virtual reality in the Institute for Computer Graphics and Virtual Reality at the University of Bremen in Bremen, Germany. He teaches virtual reality, computer graphics, and geometric computing to Bachelor's and Master's students. His research interests include geometric algorithms for computer graphics, massively parallel algorithms on the GPU, virtual prototyping, and medical immersive simulators.

**DIRK WEYHE**, M.D., is a professor of visceral surgery and Head of the Department for Visceral Surgery at the University Clinic Oldenburg, Carl von Ossietzky University Oldenburg in Oldenburg, Germany. His research interests are in digital medicine and carcinoma studies.

### LITERATURE CITED

- Amarin NZ, Ghishan RI. 2013. Learning with technology from a constructivist point of view. *Int J Bus Humanit Technol* 3:52–57.
- Andersen SA, Mikkelsen PT, Konge L, Cayé-Thomasen P, Sørensen MS. 2016. Cognitive load in mastoidectomy skills training: Virtual reality simulation and traditional dissection compared. *J Surg Educ* 73:45–50.
- Birbara NS, Pather N. 2021. Real or not real: The impact of the physical fidelity of virtual learning resources on learning anatomy. *Anat Sci Educ* 14:774–787.
- Birbara NS, Sammut C, Pather N. 2020. Virtual reality in anatomy: A pilot study evaluating different delivery modalities. *Anat Sci Educ* 13:445–457.
- Bracq MS, Michinov E, Arnaldi B, Caillaud B, Gibaud B, Gouranton V, Jannin P. 2019. Learning procedural skills with a virtual reality simulator: An acceptability study. *Nurse Educ Today* 79:153–160.
- Brewer-Deluce D, Bak AB, Simms AJ, Sinha S, Mitchell JP, Shin D, Saraco AN, Wainman BC. 2021. Virtual reality bell-ringer: The development and testing of a stereoscopic application for human gross anatomy. *Anat Sci Educ* 14:330–341.
- Dua A, Coppola KM, Mulheron GW, Troupe D, Lebeau R. 2021. Development of a novel peer-sharing application to supplement learning from cadaveric dissection. *Anat Sci Educ* 14:491–504.
- Field A, Miles J, Field Z. 2012. *Discovering Statistics Using R*. 1st Ed. London, UK: SAGE Publications Ltd. 992 p.
- Fosnot CT, Perry RS. 1996. *Constructivism: A psychological theory of learning*. In: Fosnot CT (Editor). *Constructivism: Theory, Perspectives, and Practice*. 1st Ed. New York, NY: Teachers College Press. p 8–33.

- Frederiksen JG, Sørensen SM, Konge L, Svendsen MB, Nobel-Jørgensen M, Bjerrum F, Andersen SA. 2020. Cognitive load and performance in immersive virtual reality versus conventional virtual reality simulation training of laparoscopic surgery: A randomized trial. *Surg Endosc* 34:1244–1252.
- Freina L, Ott M. 2015. A literature review on immersive virtual reality in education: State of the art and perspectives. In: Proceedings of the 11th International Conference eLearning and Software for Education (eLSE 2015): Rethinking Education by Leveraging the eLearning Pillar of the Digital Agenda for Europe; Bucharest, Romania, 2015 April 23–24. p 133–141. Carol I National Defense University, Bucharest, Romania.
- Gureckis TM, Markant DB. 2012. Self-directed learning: A cognitive and computational perspective. *Perspect Psychol Sci* 7:464–481.
- Hazlett CB. 2009. Prerequisite for enhancing student learning outcomes in medical education. *Sultan Qaboos Univ Med J* 9:119–123.
- Huber T, Paschold M, Hansen C, Wunderling T, Lang H, Kneist W. 2017. New dimensions in surgical training: Immersive virtual reality laparoscopic simulation exhilarates surgical staff. *Surg Endosc* 31:4472–4477.
- Johnson-Glenberg MC. 2018. Immersive VR and education: Embodied design principles that include gesture and hand controls. *Front Robot AI* 5:81.
- Koivisto J, Hamari J. 2019. The rise of motivational information systems: A review of gamification research. *Int J Inform Manag* 45:191–210.
- Kornell N, Hays MJ, Bjork RA. 2009. Unsuccessful retrieval attempts enhance subsequent learning. *J Exp Psychol Learn Mem Cognit* 35:989–998.
- Krathwohl DR. 2002. A revision of Bloom's taxonomy: An overview. *Theor Pract* 41:212–218.
- Krokos E, Plaisant C, Varshney A. 2019. Virtual memory palaces: Immersion aids recall. *Virtual Reality* 23:1–15.
- Kurul R, Ögün MN, Neriman Narin A, Avci S, Yazgan B. 2020. An alternative method for anatomy training: Immersive virtual reality. *Anat Sci Educ* 13: 648–656.
- Markant DB, Ruggeri A, Gureckis TM, Xu F. 2016. Enhanced memory as a common effect of active learning. *Mind Brain Educ* 10:142–152.
- Parong J, Mayer RE. 2018. Learning science in immersive virtual reality. *J Educ Psychol* 110:785–797.
- Paulsen F, Waschke J. 2017a. Sobotta, Atlas der Anatomie Band 1: Allgemeine Anatomie und Bewegungsapparat. 24th Ed. Munich, Germany: Elsevier GmbH / Urban & Fischer. 469 p.
- Paulsen F, Waschke J. 2017b. Sobotta, Atlas der Anatomie Band 2: Innere Organe. 24th Ed. Munich, Germany: Elsevier GmbH / Urban & Fischer. 352 p.
- Paulsen F, Waschke J. 2017c. Sobotta, Atlas der Anatomie des Menschen Band 3: Kopf, Hals und Neuroanatomie. 24th Ed. Munich, Germany: Elsevier GmbH / Urban & Fischer. 941 p.
- Pius-Hospital Oldenburg. 2018. Immersive virtual reality anatomy atlas V02. YouTube video. Pius-Hospital Oldenburg, Oldenburg, Germany. 1:24 min. URL: <https://www.youtube.com/watch?v=JY50Wjh-olw> [accessed 23 November 2020].
- Rizzetto F, Bernareggi A, Rantas S, Vanzulli A, Vertemati M. 2020. Immersive virtual reality in surgery and medical education: Diving into the future. *Am J Surg* 220:856–857.
- Sankaranarayanan G, Odlozil CA, Wells KO, Leeds SG, Chauhan S, Fleshman JW, Jones DB, De S. 2020. Training with cognitive load improves performance under similar conditions in a real surgical task. *Am J Surg* 220:620–629.
- Selzer MN, Gazcon NF, Larrea ML. 2019. Effects of virtual presence and learning outcome using low-end virtual reality systems. *Displays* 59:9–15.
- Seymour NE. 2008. VR to OR: A review of the evidence that virtual reality simulation improves operating room performance. *World J Surg* 32:182–188.
- Siemens G. 2005. Connectivism: A learning theory for the digital age. *Int J Instr Technol Dist Learn* 2:1.
- Stepan K, Zeiger J, Hanchuk S, Del Signore A, Shrivastava R, Govindaraj S, Iloreta A. 2017. Immersive virtual reality as a teaching tool for neuroanatomy. *Int Forum Allergy Rhinol* 7:1006–1013.
- Wainman B, Aggarwal A, Birk SK, Gill JS, Hass KS, Fenesi B. 2021. Virtual dissection: An interactive anatomy learning tool. *Anat Sci Educ* 14:788–798.
- Wainman B, Pukas G, Wolak L, Mohanraj S, Lamb J, Norman GR. 2020. The critical role of stereopsis in virtual and mixed reality learning environments. *Anat Sci Educ* 13:401–412.
- Weyhe D, Usler V, Weyhe F, Kaluschke M, Zachmann G. 2018. Immersive anatomy atlas—Empirical study investigating the usability of a virtual reality environment as a learning tool for anatomy. *Front Surg* 5:73.
- Winn W, Hoffman H, Hollander A, Osberg K, Rose H, Char P. 1997. The effect of student construction of virtual environments on the performance of high- and low-ability students. In: Proceedings of the Annual Meeting of the American Educational Research Association (AERA 1997): Talking Together in Educational Research and Practice; Chicago, IL, 1997 March 24–28. p 1–7. American Educational Research Association, Washington, DC.
- Wong A, Leahy W, Marcus N, Sweller J. 2012. Cognitive load theory, the transient information effect and e-learning. *Learn Instr* 22:449–457.
- Yeh DD, Park YS. 2015. Improving learning efficiency of factual knowledge in medical education. *J Surg Educ* 72:882–889.
- Yildirim C. 2019. Cybersickness during VR gaming undermines game enjoyment: A mediation model. *Displays* 59:35–43.
- Zackoff MW, Real FJ, Cruse B, Davis D, Klein M. 2019. Medical student perspectives on the use of immersive virtual reality for clinical assessment training. *Acad Pediatr* 19:849–851.
- Zinchenko YP, Khoroshikh PP, Sergievich AA, Smirnov AS, Tomyalis AV, Kovalev AI, Gutnikov SA, Golokhvast KS. 2020. Virtual reality is more efficient in learning human heart anatomy especially for subjects with low baseline knowledge. *New Ideas Psychol* 59:100786.

---

### A SHARED HAPTIC VIRTUAL ENVIRONMENT FOR DENTAL SURGICAL SKILL TRAINING

#### *Outline*

---

	<b>ABSTRACT</b> . . . . .	213
<b>1</b>	<b>INTRODUCTION</b> . . . . .	213
<b>2</b>	<b>RELATED WORK</b> . . . . .	213
<b>3</b>	<b>DESIGN</b> . . . . .	214
	3.1 Collaborative Setup . . . . .	214
	3.2 Teleteaching Strategies . . . . .	215
<b>4</b>	<b>METHODS</b> . . . . .	215
	4.1 Simulation . . . . .	215
	4.2 Haptic Rendering . . . . .	215
	4.3 Visualization . . . . .	216
	4.4 Scoring . . . . .	217
<b>5</b>	<b>EVALUATION PLAN</b> . . . . .	217
<b>6</b>	<b>CONCLUSIONS AND FUTURE WORK</b> . . . . .	218
	<b>ACKNOWLEDGEMENTS</b> . . . . .	218
	<b>REFERENCES</b> . . . . .	218

---

#### *Bibliography Entry*

Kaluschke, M., Yin, M. S., Haddawy, P., Srimaneeekarn, N., Saikaew, P., and Zachmann, G. (2021). **A Shared Haptic Virtual Environment for Dental Surgical Skill Training**. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 347–352, Lisbon, Portugal. IEEE

#### *Author's Contribution*

The author wrote the Objectives, helped in the literature research and in writing the rest of the manuscript. The diagram was also generated by the author.

#### *Copyright*

Copyright © 2021 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

*This paper has been omitted due to copyright considerations.*

---

THE IMPACT OF 3D STEREOPSIS AND HAND-TOOL ALIGNMENT ON EFFECTIVENESS OF A VR-BASED SIMULATOR FOR DENTAL TRAINING

*Outline*

---

I.	INTRODUCTION . . . . .	221
II.	RELATED WORK . . . . .	222
III.	SIMULATOR . . . . .	223
	A. Stereo Rendering . . . . .	223
	B. Hand-Tool Alignment . . . . .	224
IV.	USER STUDY . . . . .	224
V.	RESULTS . . . . .	224
	A. Groups . . . . .	225
	B. 3D Rendering Modes . . . . .	225
	C. Hand-Tool Alignment . . . . .	226
VI.	DISCUSSION . . . . .	226
VII.	CONCLUSION . . . . .	227
	ACKNOWLEDGEMENT . . . . .	227
	REFERENCES . . . . .	227

---

*Bibliography Entry*

Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2022). **The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training.** In *2022 IEEE 10th International Conference on Healthcare Informatics (ICHI)*, pages 449–455, Rochester, MN, USA. IEEE

*Author's Contribution*

The author wrote the Objectives, helped in the literature research and in writing the rest of the manuscript. The diagram was also generated by the author.

*Copyright*

Copyright © 2022 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

*This paper has been omitted due to copyright considerations.*

PERCEIVED REALISM OF HAPTIC RENDERING METHODS FOR BIMANUAL HIGH FORCE TASKS: ORIGINAL AND REPLICATION STUDY

*Outline*

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>229</b>
1.1	Contribution . . . . .	230
<b>2</b>	<b>State of the art</b> . . . . .	<b>231</b>
2.1	Penalty-based methods . . . . .	232
2.3	Impulse-based methods . . . . .	233
2.4	Rigid-body-based methods . . . . .	233
<b>3</b>	<b>Methods</b> . . . . .	<b>233</b>
3.1	Experimental setup . . . . .	233
3.2	Implementation of haptic rendering methods . . . . .	234
3.3	Study design . . . . .	235
3.4	Statistical evaluation . . . . .	237
3.5	Participants demographics . . . . .	237
<b>4</b>	<b>Results</b> . . . . .	<b>237</b>
4.1	Performance of haptic rendering methods in original study . . . . .	237
4.2	Performance of haptic rendering methods in replication study . . . . .	239
4.3	Performance of haptic rendering methods: study comparison . . . . .	240
4.4	Subjective feedback . . . . .	241
4.5	Summary . . . . .	242
<b>5</b>	<b>Discussion</b> . . . . .	<b>242</b>
5.1	Original study vs. replication study . . . . .	242
5.2	Limitations . . . . .	243
<b>6</b>	<b>Conclusion</b> . . . . .	<b>243</b>
	<b>References</b> . . . . .	<b>243</b>

---

*Bibliography Entry*

Lorenz, M., Hoffmann, A., Kaluschke, M., Ziadeh, T., Pillen, N., Kusserow, M., Perret, J., Knopp, S., Dettmann, A., Klimant, P., Zachmann, G., and Bullinger, A. C. (2023). **Perceived Realism of Haptic Rendering Methods for Bimanual High Force Tasks: Original and Replication Study**. *Scientific Reports*, 13(1):11230

*Author's Contribution*

The author designed and planned the user-study, helped run the study, wrote part of “Introduction”, all of “State of the art” and “Implementation of haptic rendering methods” and helped in writing the rest of the manuscript.

*Copyright*

Copyright © The Author(s) 2023



OPEN

# Perceived realism of haptic rendering methods for bimanual high force tasks: original and replication study

Mario Lorenz<sup>1,2,3,8</sup>✉, Andrea Hoffmann<sup>4,8</sup>, Maximilian Kaluschke<sup>5,8</sup>, Taha Ziadeh<sup>6</sup>, Nina Pillen<sup>7</sup>, Magdalena Kusserow<sup>7</sup>, Jérôme Perret<sup>6</sup>, Sebastian Knopp<sup>1,3</sup>, André Dettmann<sup>4</sup>, Philipp Klimant<sup>1</sup>, Gabriel Zachmann<sup>5</sup> & Angelika C. Bullinger<sup>4</sup>

Realistic haptic feedback is a key for virtual reality applications in order to transition from solely procedural training to motor-skill training. Currently, haptic feedback is mostly used in low-force medical procedures in dentistry, laparoscopy, arthroscopy and alike. However, joint replacement procedures at hip, knee or shoulder, require the simulation of high-forces in order to enable motor-skill training. In this work a prototype of a haptic device capable of delivering double the force (35 N to 70 N) of state-of-the-art devices is used to examine the four most common haptic rendering methods (penalty-, impulse-, constraint-, rigid body-based haptic rendering) in three bimanual tasks (contact, rotation, uniaxial transition with increasing forces from 30 to 60 N) regarding their capabilities to provide a realistic haptic feedback. In order to provide baseline data, a worst-case scenario of a steel/steel interaction was chosen. The participants needed to compare a real steel/steel interaction with a simulated one. In order to substantiate our results, we replicated the study using the same study protocol and experimental setup at another laboratory. The results of the original study and the replication study deliver almost identical results. We found that certain investigated haptic rendering method are likely able to deliver a realistic sensation for bone-cartilage/steel contact but not for steel/steel contact. Whilst no clear best haptic rendering method emerged, penalty-based haptic rendering performed worst. For simulating high force bimanual tasks, we recommend a mixed implementation approach of using impulse-based haptic rendering for simulating contacts and combine it with constraint or rigid body-based haptic rendering for rotational and translational movements.

Whilst the visual rendering quality for Virtual Reality (VR) applications has dramatically increased in the last two decades, the quality of haptic feedback is severely lagging behind in simulation realism. This is especially crucial for VR applications for training motor-skills. Obviously, procedural knowledge about the correct execution of a task is trainable with visuals/audio-only VR, but the most important, and often most difficult part, the motor-skill training, still has to be trained in reality since proper haptic simulations often do not exist or are not good enough.

A prominent application area where this can be observed is the training of surgical procedures using VR<sup>1</sup>. For procedures requiring the simulation of small forces, e.g. laparoscopic and arthroscopic procedures<sup>2-6</sup>, dentistry tasks<sup>7,8</sup> or endoscopic surgeries<sup>9</sup>, there are suitable VR training simulators incorporating haptic feedback already available. Some residency curricula even require to train at such simulators<sup>10,11</sup>. In contrast, surgical procedures requiring the application of large forces, i.e. hip, knee and shoulder arthroplasty, lack VR training simulators with realistic haptic feedback. Here, solely visuals-only VR applications exist<sup>12</sup>, or they rely on pen-like haptic

<sup>1</sup>Professorship for Production Systems and Processes, Chemnitz University of Technology, Reichenhainer Straße 70, 09126 Chemnitz, Germany. <sup>2</sup>Department of Orthopedics, Trauma and Plastic Surgery, University Hospital Leipzig, Liebigstraße 20, 04103 Leipzig, Germany. <sup>3</sup>Division of Macroscopic and Clinical Anatomy, Gottfried Schatz Research Center, Medical University of Graz, Auenbruggerplatz 25, 8036 Graz, Austria. <sup>4</sup>Chair for Ergonomics and Innovation, Chemnitz University of Technology, Erfenschlager Straße 73, 09125 Chemnitz, Germany. <sup>5</sup>Chair of Computer Graphics and Virtual Reality, University of Bremen, Bibliothekstraße 5, 28359 Bremen, Germany. <sup>6</sup>Haption GmbH, Dennewartstraße 25, 52068 Aachen, Germany. <sup>7</sup>YOUSE GmbH, Florastraße 47, 13187 Berlin, Germany. <sup>8</sup>These authors contributed equally: Mario Lorenz, Andrea Hoffmann and Maximilian Kaluschke. ✉email: mario.lorenz@mb.tu-chemnitz.de

devices capable of providing only a fraction of the forces required for a realistic haptic simulation, e.g. over 300 N for Acetabula reaming during hips arthroplasty<sup>13</sup>. Devices like the Touch from 3D Systems, Inc., or the Omega series from Force Dimension are limited to forces up to 12 N<sup>14,15</sup>. Even the currently most advanced force feedback devices from 3D Systems, Inc., Force Dimension or Haption are currently limited to 20 N respectively to 35 N<sup>16–18</sup>. In order to overcome this shortcoming Sagardia et al.<sup>19</sup>, Kaluschke et al.<sup>20,21</sup> or Knopp et al.<sup>22</sup> were utilizing industrial robots like the KUKA LBR iiwa or its predecessors for providing force feedback up to 140 N (KUKA LBR iiwa 14).

Aside from the mechatronic capabilities of the haptic device, the utilized haptic rendering methods are crucial for the perceived haptic realism. A large body of work is available regarding these methods<sup>23</sup>; however due to the lack of high-force haptic devices these methods were never evaluated for forces above 35 N. In this study we aim to close this gap, by utilizing a Virtuoso 6D prototype from Haption, capable of delivering 70 N force feedback. We focus on investigating the four most common haptic rendering methods:

1. Penalty-based haptic rendering (*penalty*)
2. Constraint-based haptic rendering (*constraint*)
3. Impulse-based haptic rendering (*impulse*)
4. Rigid-body-based haptic rendering (*rigid body*)

Each has different advantages and disadvantages and might perform relatively better than another rendering method only in a specific task. For example, *penalty* is advantageous if constraints should be contradictable, such as inserting a peg into a hole that is smaller than its diameter, but it cannot prevent interpenetration of objects. Constraint-based haptic rendering guarantees that constraints are not violated, and especially avoids pop-through issues of penalty, but is more complex to implement. The impulse-based method gives a sharper feeling of collision, but it cannot handle stable contact on its own. Finally, rigid-body-based rendering is closer to the real laws of physics, but it is much more computationally expensive. Those four rendering methods cover the vast majority of existing methods from the literature.

As forces of above 35 N are difficult to control one-handed, they are often performed two-handed. This, and the fact that two-handed interactions are rather less explored, are the reasons why we evaluated the four haptic rendering methods in a two-handed scenario.

Our long-term goal is to provide realistic haptic feedback for hip arthroplasty where forces above 35 N are required for certain steps<sup>13</sup>. One example for this is the reaming of an Acetabulum (hip socket) during hip arthroplasty, which is done to prepare the Acetabulum geometry for the hip socket implant. The surgical tool used for this task resembles a hand-driller and is operated as such. To ream the Acetabulum, a rod with a half-spherical reamer is clamped by the tool holder of the surgical hand-driller. There is further one part around the rod which allows to grab it whilst rotating. Inspired by this task we designed and built an abstract evaluation scenario to compare the four haptic rendering methods in terms of perceived realism to a real scenario. By inserting a steel half-sphere attached at a rod, which is held by a hand-drill, into a steel cavity we can investigate the four haptic rendering methods (see Sects. “Experimental setup” and “Study design” as well as Figs. 1 and 2 for details).

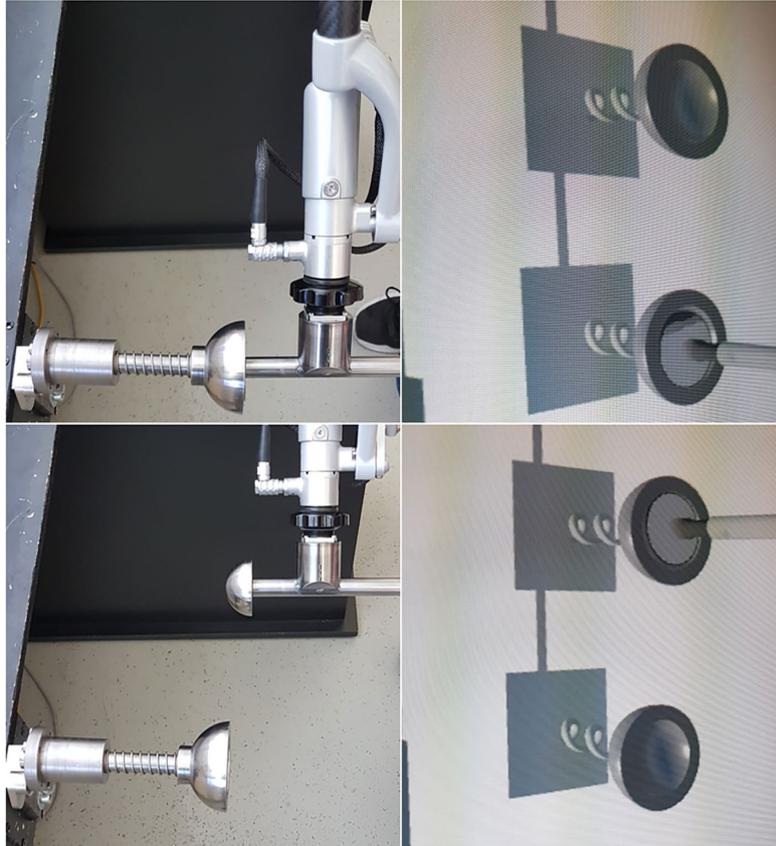
The 2015 article by the Open Science Collaboration<sup>24</sup> reported that the results of a large quantity of psychological studies published in high ranking journals could not be reproduced. This article fostered a debate about the so-called ‘replication crisis’, which also concerns the reproducibility of human subject research in Virtual, Augmented and Mixed Reality<sup>25</sup>. Unfortunately, replicating studies in Virtual, Augmented and Mixed Reality research is very rare, but highly needed to strengthen the trust in the reliability of the study results. This situation motivated us to conduct a replication study in addition to our original study using the same materials and experimental design but performed by a different research lab. Using this approach, we aim to strengthen the reliability of our results and would like to set a motivating example for other researchers in the field.

Our motivation to evaluate a steel/steel contact instead of a bone-cartilage/steel contact is based on three reasons. Firstly, bone and cartilage are biomaterials and as such their material properties vary largely between specimens unlike non-biomaterial like steel which material properties only have miniscule differences. Secondly, in a real surgical situation, the bone-cartilage part, e.g. the Acetabulum, is not rigid. It is surrounded by soft tissue and kinematically coupled via tendons and muscles to other bones, therefore, being a highly damping and moveable system. Here, the inter-specimen variety is even greater than the bone-cartilage material properties. Thoroughly evaluating haptic rendering methods in such a setting would lead to an unmanageable amount of studies to explore their performance. By abstracting this problem to a rigid steel/steel contact, we investigated the haptic rendering methods in a most challenging condition in terms of stiffness. We argue that our results present baseline data from which reliable estimations on the likely performance of the four investigated haptic rendering methods in high-force surgical scenarios can be drawn. As a third point, using a rigid steel/steel contact scenario allows us to provide conclusions for the simulation of other, non-surgical tasks, e.g. drilling into steel, concrete, wood, or screwing with an electrical screwdriver.

The research questions we answer in this paper, in respect to the abstracted physical model used in the experimental tasks, are:

- RQ1. Is any investigated haptic rendering method capable of delivering realistic haptic feedback?
- RQ2. Which investigated haptic rendering methods delivers the highest degree of perceived realism across all tasks?
- RQ3. In which way do the investigated haptic rendering methods differ in perceived realism for different tasks?

**Contribution.** Our study’s main contributions are:



**Figure 1.** The tool in contact with the real haptic object (top left) and the corresponding virtual view in VR (top right); The tool in contact with the virtual haptic object (bottom left) and the corresponding virtual view in VR (bottom right).

1. Investigation of the four most common haptic rendering methods for forces above 35 N in terms of perceived realism.
2. Comparison of the four most common haptic rendering methods in a steel/steel contact scenario to reality.
3. Validation of the original study results by a replication study.

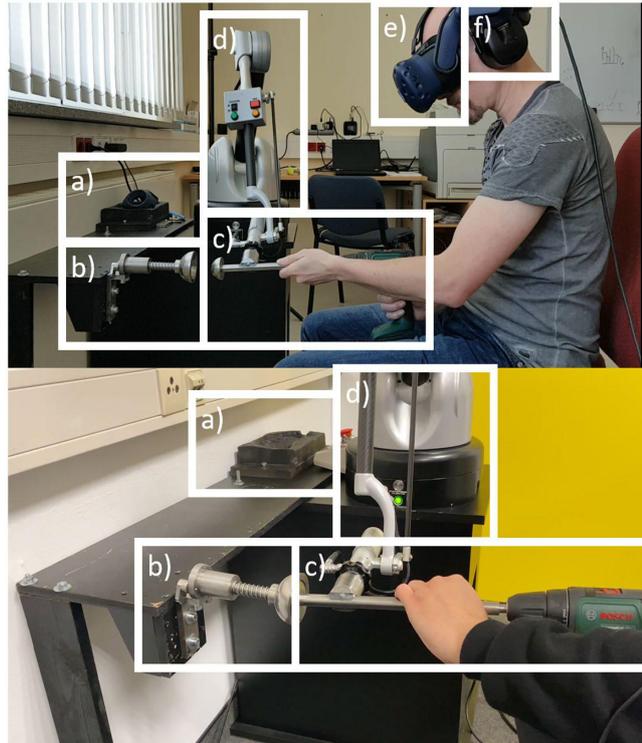
### State of the art

There have been numerous unique haptic rendering solutions presented in the last three decades. We do not intend to list each one of them. Rather, we will present four types of haptic rendering approaches which are most commonly employed throughout the literature, either rigidly or in various forms of hybrid combination. As such, this document provides a coarse overview of haptic rendering methods. For a more comprehensive list of approaches, we refer to recent literature reviews<sup>23,26,27</sup>.

Haptic rendering techniques can firstly be differentiated into two types of application methods, *direct* and *indirect* force rendering<sup>28</sup>. Direct force rendering renders forces which are exerted on a virtual object, which is simultaneously directly attached to the force rendering device. We will call this object the *haptic tool*<sup>29</sup> if it is directly attached to the haptic device. In contrast, indirect force rendering methods employ a second instance of the same geometrical object, here called the *graphic tool*<sup>29</sup>, which position and rotation is calculated in a simulation loop and the difference between the poses of *graphic tool* and *haptic tool* are rendered to the haptic device, often as a dampened spring. The exact details of how the simulation is implemented varies greatly.

When looking at both categories of application methods, we can differentiate between four major haptic rendering techniques:

1. *Penalty* (Sect. “[Penalty-based methods](#)”)
2. *Constraint* (Sect. “[Constraint-based methods](#)”)
3. *Impulse* (Sect. “[Impulse-based methods](#)”)
4. *Rigid body* (Sect. “[Rigid-body-based methods](#)”)



**Figure 2.** The experimental setup in the original study (top) and replication study (bottom). (a) Form for HTC VIVE controller (b) Real haptic object; (c) Hand-drill with attached rod and half-sphere; (d) Virtuose 6D haptic device; (e) HTC VIVE PRO; (f) Passive-noise cancelling headphones.

Where *penalty* is commonly implemented as direct force rendering techniques, and *constraint*, *impulse* and *rigid body* are commonly implemented as indirect techniques.

The terms *penalty*-, *impulse*- and *constraint*-based methods also carry definitions in multi-body dynamics (MBD) simulations and should not be confused with the similarly named haptic rendering methods. In MBD, *penalty*-, *impulse*- and *constraint*-based describe methods that solve the dynamics of physical phenomenon, such as contacts, in a physically inspired manner. In that sense, they are variations of the *rigid body*-based method, from the view of haptic rendering. Here, the impulse-based method for example does not simulate rigid bodies, but essentially overlays a damping force during a collision, on top of the force that is rendered based on the tool's configuration in space. In the following we will explain each method in more detail.

**Penalty-based methods.** Penalty-based methods treat constraint violations of the tool, such as interpenetrations with the virtual environment, by measuring the amount of violation and applying a penalty force proportional to the violation. Less abstractly, if half of a sphere is overlapping with the environment, then a penalty force with a magnitude proportional to the overlapping will be applied to the sphere. The measure that is used to quantify the constraint violation can be implemented in different ways, most commonly is the depth of penetration (dop), either translation dop or generalized dop<sup>30</sup>, or volumetric measures<sup>31,32</sup> or adaptive stiffness<sup>28</sup>. This rendering method is most commonly implemented as direct application. However, there are techniques to modify the method to be indirect application<sup>33</sup>, which can improve system stability. Penalty methods are easy to implement, as only discrete collisions and a penetration measure are needed. However, the disadvantages are plenty: constraints can be violated, such as overlap and discrete collision detection can miss a fast moving collision. The penetration measure may define the inside based on the current position, which can lead to pop-through events, when penetrating too far. Based on the fact that penalty violates real non-overlapping behavior, we expect this method to perform worse than all other methods in all tasks, especially when the normal force is high, such as during pushing.

**Constraint-based methods.** Constraint-based rendering methods collect and solve non-penetration constraints that result from the virtual environment's geometry contacts with the *graphic tool*, in order for it to remain on the surface without interpenetration. This interpenetration is of high importance in some industries, i.e. in construction in order to check if a car or plane can be assembled. Xu et al.<sup>38</sup> showed clearly the benefit of constraint-based methods regarding overlap-free behavior. The computation time is usually proportional to the number of contacts and might dip significantly below the desired 1 kHz update-rate, as constraint solving

is usually computation heavy. This is commonly mitigated by parallelizing the indirect rendering scheme and constraint solving<sup>34</sup>. However, some methods are even fast enough to allow for highly dynamic environments, such as streaming point clouds<sup>35</sup> or material removal<sup>36</sup>. When the constraints are approximated very coarsely or linearized, these methods are commonly referred to as proxy-based methods. Constraint methods are usually the most complicated to implement, as the computational load needs to be processed at sufficiently fast speed in order to allow for interaction. Most time is spent calculating contact points and solving for the new movement of the tool under the current constraints. Usually, constraint methods exhibit very little intersection between tool and environment, such that it is not easily visible and in theory they could also guarantee intersection-free tool movement, if constraints are solved in parallel. Based on the fact that these methods are stable and don't allow for visible overlap, we suspect they have overall good performance.

**Impulse-based methods.** Constantinescu et al.<sup>37</sup> introduced a combination of a penalty method with the addition of impulsive forces when a colliding impact is detected. That means the tool collides with a velocity magnitude that exceeds a certain threshold, as to not trigger on resting contact. The force impulses are calculated based on Newton's restitution law, inspired by impulse-based rigid body simulation<sup>38</sup>. However, when it is employed in haptic rendering there is an increase in perceived hardness when coming into contact with virtual objects. Kuchenbecker et al.<sup>39</sup> showed that adding event-driven transients significantly increases the perceived hardness of a constraint-based force computation. Similarly, to these transients, many short impulses arise when using impulse-based rendering, which suggests that these high-frequency responses are the reason for the increase in perceived hardness. The implementation can be done on top of any kind of underlying haptic rendering method, giving it a lot of flexibility. This method most likely performs especially well in tasks where collisions happen often, such as coming into contact.

**Rigid-body-based methods.** A broad group of methods are built on top of a rigid-body simulation. Consequently, this opens up a vast and completely separate area of research. Popular commercial physics engines are PhysX<sup>40</sup>, Bullet<sup>41</sup>, and Havok<sup>42</sup>, as well as open source engines like ODE<sup>43</sup> and Box2D<sup>44</sup> for 2D cases. We refer to Bender et al.<sup>45</sup> for a comprehensive and educational overview of rigid-body simulation approaches, as there are numerous ways to implement the underlying simulation. However, the integration with the haptic device is usually achieved in same way in all rigid-body simulation, with a dampened spring between *haptic* and *graphic tool*, which enables the haptic device to drive the simulation. The force that this spring exerts on the *haptic tool* is displayed to the user, as such it is a classic indirect application method. We suspect rigid-body-based methods will perform similarly well to constraint-based methods, since both are stable and don't allow for visible overlap. However, hard contacts can likely not be rendered as well as with impulse-based methods.

## Methods

The four haptic rendering methods were investigated using a with-in-subject study design. Every participant compared the real object on the left with the virtual one on the right in every rendering condition and in order of three tasks. The study design for the original and the replication study were identical. The experimental setup is described in Sect. "Experimental setup". The only differences were the location of the study and the principle investigator guiding the study. Sections "Implementation of haptic rendering methods" and "Study design" are providing the implementation of the haptic rendering methods respectively the study design. The statistical methods used are described in Sect. "Statistical evaluation" and the participants demographics in Sect. "Participants demographics".

Ethical approval was obtained from Chemnitz University of Technology ethics committee (number: #101534678). All methods were performed in accordance with the relevant guidelines and regulations. All participants provided written and informed consent. A COVID-19 protection protocol, approved by the Chemnitz University of Technology and University of Bremen, was prepared and followed.

**Experimental setup.** In order to ensure a similar body posture to acetabulum reaming, a seated setup was chosen where the participants should hold a hand-drill with an attached rod (steel, 15 mm diameter, 300 mm length) that had a half-sphere (steel, 50 mm diameter) at its end (see Fig. 2c). This hand-drill was connected to a novel Virtuose 6D prototype from Haption (see Fig. 2d). The prototype is an evolution of the standard Virtuose 6D, with higher torques on all motors, giving a maximum force of 70 N in translation (5 Nm in rotation) at the wrist in the whole workspace. The higher motor torques are achieved by larger reduction factors, so that the domain of stability increased and allows for a control stiffness up to 12 kN/m in translation (40 Nm/rad in rotation) with an update rate of 1 kHz. The connector to the Virtuose 6D was welded to the rod 60 mm behind the tip of the half-sphere. As an interaction counterpart a haptic object (see Fig. 2b) was chosen, with a cavity (steel, 50 mm inner diameter, 70 mm outer diameter) of the same size as the half-sphere at the hand-drill. This cavity was screwed to a steel rod (steel, 15 mm diameter, 128 mm length) which was running inside a bush (steel, 15 mm diameter, 60 mm length) allowing a guided uniaxial movement of the cavity. Between the cavity and the bush, a spring (spring constant = 1061 N/m) was mounted with a preload of 30 N, in order to enable a contact between both parts without the cavity moving. Further, 30 N is close to the upper boundaries of forces that commercially available haptic devices are able to deliver. Towards the end of the rod leaving the bush, a notch was milled. At the backside of the bush a lose lever was screwed, gliding on the rod until the notch appeared. The lever would fall into the notch due to gravity therefore blocking the cavity in this position. By rising the lever, the cavity is released returning to the starting position. This mechanism allowed to push the cavity 27 mm in, starting from 30 to 60 N resisting force. This interval was chosen as it represents the upgraded force capabilities of the novel Virtuose 6D prototype. We decided to stay 10 N below its maximum capacity in order to avoid the

participants over pushing the device. In order to attach this haptic object to a table a steel plate was screwed to the bush with two M8 drill holes.

The Virtuose 6D prototype with the attached hand-drill and the haptic object were mounted on a table in front of which the participants can be seated on a height adjustable chair. In order to block the auditory channel and any bias coming from here, the participants were wearing a passive-noise cancelling headset. Via this headset an additional white noise signal was conveyed to the participants (see Fig. 2f). The choice to use passive noise cancellation was due to the high frequency and transient nature of the noise, which is better blocked by passive techniques<sup>46</sup>.

A visual cue was essential to ensure that the participants hit the cavity with the hand-drill. Via an HTC VIVE PRO Head-mounted display (HMD) the participants saw a sparse virtual environment of the table with the hand-drill and two haptic objects next to each other, without a virtual body (see Figs. 1, 2e). The left haptic object was co-located with the position of the real haptic object. The right haptic object was placed 160 mm next to the left one and solely virtual (see Fig. 1). It was designed to deliver the haptic force feedback of the four haptic rendering methods. We ensured that the participants could not collide with the table whilst interacting with the right haptic object. The position of both haptic objects right next to each other was chosen carefully. This arrangement assures that the participant only had to move minimally to compare both haptic objects, while also keeping the kinematic chain of the Virtuose 6D almost unchanged. Thus, bias from the participants posture and the Virtuose' kinematic chain was minimized. A visualization of this evaluation scenario was required so that the participants were able to make contact of the half-sphere with the cavity. This was implemented with the Unity 3D engine. In a blind-folded scenario, some sort of haptic guiding mechanism would have had to be designed, which would properly had led to a bias in the perceived haptics. The position of the hand-drill in VR was tracked via the Virtuose 6D. In order to have a smooth and stable visual movement when the cavity was pushed in, a collision of the half-sphere and the cavity was detected with Unity. The virtual movement of both cavities in VR was then done using the tracking information from the Virtuose 6D.

The alignment of the VR scenario with the real table and the haptic object was achieved with a form where an HTC VIVE controller could be placed (see Fig. 2a). This way the location and orientation of the HTC VIVE Controller in the VR frame was registered. The relative position of this form to the base of the Virtuose 6D was measured. The Virtuose 6D itself was screwed to the table at a fixed position. Based on this information the offset between the Virtuose 6D and the HTC VIVE Controller was computed, and the virtual camera in VR positioned accordingly. Due to tracking errors of the Lighthouse Tracking System a minimal, but still perceivable offset of the virtual and the real haptic object was sometimes present. Thus, the position of the virtual haptic object had to be slightly adjusted manually by the principle investigator in order to achieve a sufficient alignment.

Via a web-based interface the principal investigator switched the haptic rendering methods. Further, a Unity window enabled the principal investigator to follow the participants' actions in VR and provided input fields for the initial alignment of the real and the virtual environment.

The entire software of the Virtuose 6D, the VR scene and the web-interface ran on a single PC with an Intel® Core™ i7-3770 CPU 3.40 GHz, 16.0 GB of RAM, a NVIDIA Quadro M6000 graphic card, and Windows 10.

**Implementation of haptic rendering methods.** The following sections describe in detail the implementation of the four haptic rendering methods as used in the experiment. The choice of the methods reflects the taxonomy used in the review paper Zendejas et al.<sup>4</sup>. The forces and torques we describe in the following sections are calculated in relation to the virtual object, the hemisphere tool. However, before applying them to the haptic device, they need to be translated to the device in order for the contact to be perceived at the sphere's location. The device force and torque are calculated in the following way:

$$F_{device} = F,$$

$$\tau_{device} = \tau + ((t_h - p_{device}) \times F),$$

where  $t_h$  is the *haptic tool's* sphere center, which coincides with the physical steel sphere mounted to the haptic device.  $p_{device}$  is the control point of the haptic device. The Virtuose 6D device can display a maximum torque of 5 Nm at its control point, and the tool was designed in such a way that the offset to the center of the steel sphere does not exceed 7 cm in any direction. Therefore, forces up to 70 N can be rendered without saturating the torques of the haptic device.

All methods have been implemented from scratch, without the use of third-party libraries. Our implementation guarantees a very short execution time, and all methods achieve a stable update rate of 1 kHz.

Because the spring-damper parameters can significantly affect the perception of the haptic rendering, we made sure that their effective values were identical throughout all four haptic rendering methods. The limit of stability of the Virtuose 6D device at the sphere center was determined experimentally and set to  $k_1 = 6000$  N/m and  $b_1 = 100$  N/m/s. We used those values to simulate unyielding contact, i.e. for the end of travel of the rod. For the compression phase, we used  $k = 1481$  N/m and  $b = 20$  N/m/s, as these values best resemble the real spring. The spring preload of 30 N is simulated quite literally, i.e. the cavity does not move unless/before a force above 30 N is applied by the test subject."

**Penalty-based methods.** The first haptic rendering method uses penalization of the penetration distance. The virtual environment consists only of a single hemisphere cavity, which means we can assume outside is always pointing towards the center of the hemisphere. Now wherever the *haptic tool* is, we only need to determine if it is in contact with discrete collision detection. In case it is in contact we calculate the force from the *haptic tool's*

center  $t_h$  towards the hemisphere's center  $c_h$  as a dampened spring:  $F = k(c_h - t_h) + bv_h$  with  $v_h$  being the *haptic tool's* velocity,  $k$  and  $b$  being the spring-damper parameters discussed above. Our method does not exhibit any pop-through.

**Constraint-based methods.** The second haptic rendering method uses a proxy constrained outside of contact<sup>47</sup>. As our scenario consists only of a single sphere (the geometry of the tool) constrained in a half-sphere cavity, our constraint is solved by translational projection of the *graphic tool* sphere  $t_g$  towards the *haptic tool*  $t_h$ , with the obstacle being the cavity. This computation is very fast and geometrically exact, and therefore free of such artifacts as can be experienced with surface meshes or assemblies of geometric primitives. In our implementation, the reaction force is very similar to our penalty-based force, with the notable difference, that the tool does not visually penetrate the cavity. The force is then calculated by  $F = k(t_g - t_h) + bv_h$ .

**Impulse-based methods.** We combined the implementation of our constraint-based method (Sect. "Constraint-based methods") with added impulsive forces according to Constantinescu et al.<sup>37</sup> to represent impulse-based force rendering in our user study through a four channel teleoperation controller. The force rendering overall runs the same routine as the constraint-based method, except that there is an additional force overlay which comes into play when collisions occur. A collision is a new contact that has a non-zero, non-separating relative velocity. In that case the force overlay will display an impulse during this time window, based on the tool friction state, as well as relative normal and tangential velocity to the obstacle (for details see Constantinescu et al.<sup>37</sup>). In case of a sphere along a single movement axis, the formula 26 of the Constantinescu et al.<sup>37</sup> simplifies to:  $F_{env} = -\frac{(1+e)}{\Delta t} \dot{q}$  which is equivalent to an additional damping. In practice, we used the formula  $F_{impulse} = b_{impulse} v_h$  with  $b_{impulse} = 300\text{N/m/s}$  which acts as an additional force overlay to the force calculation given in Sect. "Constraint-based methods".

**Rigid-body-based methods.** In the last haptic rendering method, the proxy is managed by a rigid-body simulation using the approach of Ortega et al.<sup>34</sup>. The *graphic tool's* trajectory is interactively simulated based on the *haptic tool's* position  $t_h$  and *graphic tool's* position  $t_g$ . A difference between them results in an acceleration of the *graphic tool*  $a = \frac{k}{m}(t_h - t_g)$  with  $m = 9.87\text{kg}$  (mass of the real tool). We project the acceleration, such that it does not violate the non-penetration constraint. The acceleration is then integrated to yield velocity (which is projected onto the constraint as well) and thus updates the position of the *graphic tool*. The force display is calculated similar to the other methods  $F = k(t_g - t_h) + bv_h$ .

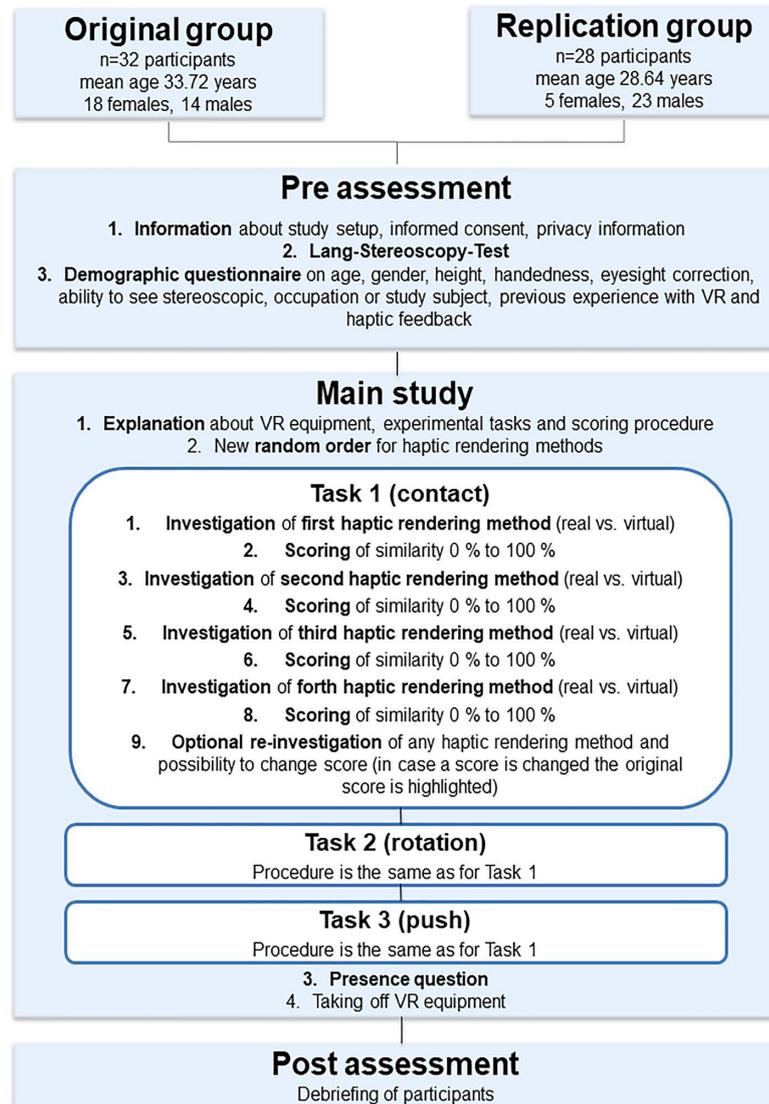
**Study design.** A written study protocol including all instructions given to the participants from welcoming them to their debriefing was prepared and followed during the experiment. It consisted of three parts: (1) pre-assessment, (2) main study and (3) post-assessment (see Fig. 3).

After welcoming the participants, they were informed about the study setup and data privacy verbally and written. After giving their written informed consent the participants filled out the demographic's questionnaire providing information about age, gender, height, handedness, eyesight correction, ability to see stereoscopic, occupation or study subject, previous experience with VR and haptic feedback. The stereoscopic vision of each participant was additionally checked by the principal investigator using the Lang-Stereoscopy-Test<sup>48</sup> Lang Stereotest II from LANG-STEREOTEST AG, Switzerland.

The main-study-part began with the participants adjusting the height of the chair where they sat in front of the experimental setup (see Fig. 2). The principal investigator explained the VR and haptic devices as well as the experimental tasks and how the participants should score the realism of the haptic rendering methods. Next, they were presented with three consecutive tasks. In each task the participants investigated all four haptic rendering methods which order was randomized between participants. We made sure to uniformly sample the permutations of the possible order of methods to minimize the effect of the order on the results. The order of tasks was content-related and could not be randomized.

In Task 1 the participants should only make contact between the half-sphere attached at the hand drill and the cavity, which corresponds to placing the reamer on the acetabulum during real surgery (see Fig. 4, left). Task 2 consisted of rotating the hand-drill inside the cavity, which corresponds to finding the correct angle for reaming during real surgery (see Fig. 4, center). Here the participants were instructed to only pay attention to the feeling of rotation and not to push the hand-drill. In Task 3 the participants were asked to push the haptic object with the hand-drill until it either blocked (real) or was highlighted green (virtual; see Fig. 4, right). They were instructed only to include the sensation of uniaxial movement into their score but not the blocking/highlighting. This corresponds to the actual reaming during real surgery when material is removed. Inspired by Park et al.<sup>49</sup> each haptic rendering method was scored on a scale from 0 to 100%. Therefore, the participants were always asked "Please score now how close the simulation on the right side resembles the real experience on the left side. Please provide a value between 0 and 100%". This question was adapted from literature investigating haptic feedback<sup>39,50-54</sup>.

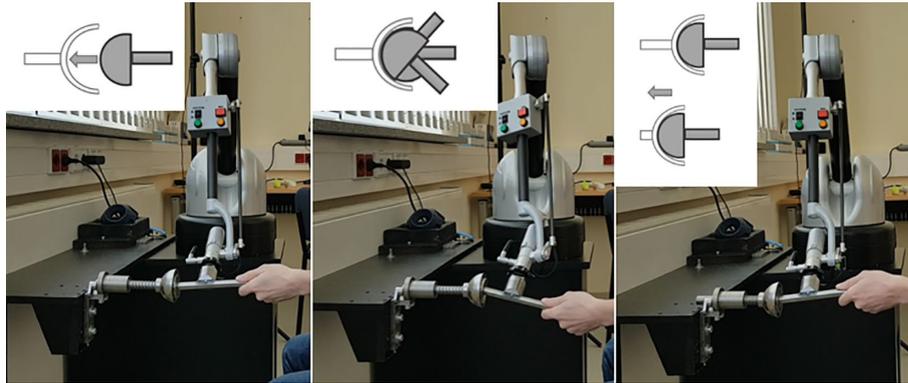
Then the participants put on a passive noise cancelling headphones and the HMD. After the participants were familiar with the virtual environment, the hand drill was given to them and Task 1 was explained again. After the participants said that they were ready, white noise was given via the headphones and the first haptic rendering method was presented. After they stopped investigating the haptic object with the hand drill the principal investigator turned off the white noise, asked for their scoring and noted it. The second rendering method and the white noise were activated so that the participants could investigate the second haptic rendering method. This procedure continued until all four haptic rendering methods in all three tasks were investigated and scored by the participants. The participants were able to retry any rendering method during a task and could also change



**Figure 3.** Graphical overview of the most important steps of the study protocol.

their scoring. The principal investigator reminded them of these options at the end of each task. If the participants changed a score, the first score was noted and highlighted. After the completion of all three tasks the hand drill was taken from the participants but they remained seated, still wearing the HMD. The principal investigator asked the participants for an overall ranking of all for haptic rendering methods from the best to the worst. This question was not disclosed to the participants earlier in order to not influence their subjective rating. Therefore, the principal investigator made sure that the participants always knew if they were currently interacting with the first, second, third or fourth during the investigation of the haptic rendering methods in each task. Next, the participants were asked to rate their subjective presence on a 1 to 10 scale inside the virtual environment using Bouchard et al. single-item measure “To which extend do you feel present in the virtual environment, as if you were really there”<sup>55</sup> following the advice of Skarbez et al.<sup>56</sup> for the application of presence questionnaires. The presence measure was used to control the influence of visual cues in the evaluation of perceived realism, assuming that sufficient presence means no or only little irritations via the VR technology are given. Lastly, the participants took off the HMD and headphones.

During post-assessment the participants were asked about their well-being and any questions regarding the experiment were answered by the principal investigator.



**Figure 4.** Task 1: Making contact between the tool and the haptic object (left); Task 2: Rotating the tool in the cavity of the haptic object (center); Task 3: Pushing-in the haptic object with the tool (right).

The participants of the study were recruited using mailing lists of the Chemnitz University of Technology and University of Bremen and social media. Only participants aged over 18, with stereoscopic vision and normal or corrected eyesight (contact lenses or glasses) were included.

**Statistical evaluation.** Data analysis was carried out with IBM SPSS Statistics 28. To compare the demographics of the original and the replication study samples, *t* tests for independent samples were calculated.

To identify differences of the perceived realism between the haptic rendering methods and between the tasks, a two-way repeated measures ANOVA was performed. The according preconditions were checked using a Shapiro–Wilk-test to examine the residues for normal distribution, followed by analyzing a boxplot for outliers. Mauchly's test of sphericity was calculated to check the variances of differences between all possible pairs of conditions. If sphericity was not fulfilled Greenhouse–Geisser correction was applied. Pair-wise *t*-tests with Bonferroni correction were used as post-hoc tests.

Differences between original and replication study in relation to the rendering conditions and tasks were analyzed with a mixed ANOVA. *P*-values  $\leq 0.05$  were considered as statistically significant. Partial eta squared ( $\eta^2_p$ ) and Cohen's  $d_z$  as effect sizes were interpreted according to conventions of Cohen<sup>57</sup>.

To verify the consistency of the overall ranking of the four haptic rendering methods and their perceived realism in the tasks Spearman's rho ( $\rho$ ) as rank correlation coefficient was calculated. For this the perceived realism assessment of every rendering method was summarized over all tasks.

**Participants demographics.** The original study included 32 subjects – 18 females and 14 males (self-identified). Individuals age ranged from 22 to 57 years ( $M=33.72$ ,  $SD=9.25$ ). Two of the subjects were left-handed, 22 subjects wore glasses or contact lenses during the study. All participants in the original study passed the Lang-Stereoscopy-Test, ensuring their ability to correctly perceive stereoscopic images. For the replication study 29 participants were recruited. One person failed the Lang-Stereoscopy-Test and was excluded from the study. The remaining 28 subjects age ranged from 20 to 55 years ( $M=28.64$ ,  $SD=7.99$ ). Of those, 5 self-identified as female and 23 as male. One of the subjects was left-handed, 16 subjects were wearing glasses or contact lenses during the study (see Table 1). In both studies most of the participants were experienced with VR and some with haptic feedback. The two study samples differ significantly in age and body height. The original study sample is older ( $t(58)=2.26$ ,  $p=0.028$ ) and smaller ( $t(58)=2.48$ ,  $p=0.016$ ).

## Results

Section “Performance of haptic rendering methods in original study” provides the results of the perceived realism of the haptic rendering methods for the original study. In Sect. “Performance of haptic rendering methods in replication study” the results of the replication study are described. In Sect. “Performance of haptic rendering methods: original study vs. replication study” the two studies are compared. Section “Subjective feedback” includes selection of comments from the participants and observation from the principle investigators from both studies. Section “Summary” is briefly summarizing all results.

**Performance of haptic rendering methods in original study.** Mean perceived realism values of all haptic rendering methods vary between 48.28 ( $SD=23.78$ ) and 73.28 ( $SD=15.17$ ; see Table 2). Before calculating the ANOVA, the assumptions were checked and confirmed. The two-way ANOVA shows a significant main effect for differences between the haptic rendering methods (Greenhouse–Geisser  $F(2.13; 65.90)=4.02$ ,  $p=0.021$ ,  $\eta^2_p=0.12$ ). The post-hoc tests for all haptic rendering methods independent of the task are significant for differences between *penalty* and *constraint* ( $p=0.006$ ;  $d_z=-0.65$ ) and between *penalty* and *rigid body* ( $p=0.018$ ;  $d_z=0.57$ ). *Penalty* was perceived less real than all other haptic rendering methods (see Table 2 and Fig. 5). The main effect for the task is also significant and reveals the highest effect size (Greenhouse–Geisser

Variable	Value	Original study N = 32	Replication study N = 28
Gender	Female	18 (56.3%)	5 (17.9%)
	Male	14 (43.8%)	23 (82.1%)
	Diverse	0	0
Age [years]		$M = 33.72$	$M = 28.64$
		$SD = 9.25$	$SD = 7.99$
		Range = 22–57	Range = 20–55
Body height [cm]		$M = 173.13$	$M = 178.79$
		$SD = 9.30$	$SD = 8.28$
		Range = 159–193	Range = 164–196
Handedness	Right	30 (93.8%)	27 (96.4%)
	Left	2 (6.3%)	1 (3.6%)
Visual aid	No visual aid	10 (31.3%)	12 (42.9%)
	Glasses	14 (43.8%)	12 (42.9%)
	Contact lenses	4 (12.5%)	3 (10.7%)
	Glasses & lenses	4 (12.5%)	1 (3.6%)
Occupation	Researchers	18 (56.3%)	8 (28.6%)
	Student	6 (18.8%)	18 (64.3%)
	Other	8 (25.0%)	2 (7.1%)
VR experience	Yes	24 (75.0%)	27 (96.4%)
	No	8 (25.0%)	1 (3.6%)
Haptic experience	Yes	13 (40.6%)	15 (53.6%)
	No	19 (59.4%)	13 (46.4%)

**Table 1.** Demographics of original and replication study.

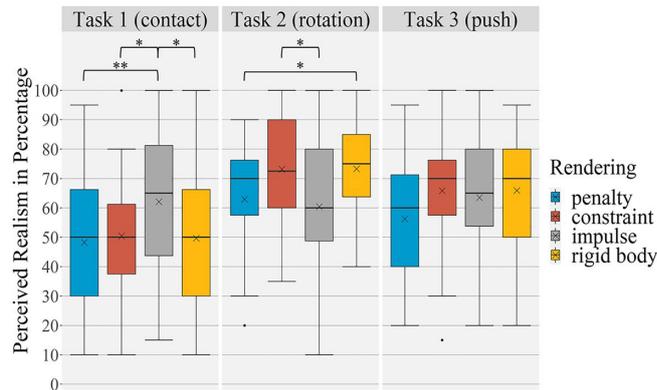
Task	Rendering	M	SD
1 (contact)	Penalty	48.28	23.78
	Constraint	50.43	20.64
	Impulse	62.09	23.88
	Rigid body	49.68	21.99
2 (rotation)	Penalty	62.97	19.46
	Constraint	73.12	17.76
	Impulse	60.43	22.44
	Rigid body	73.28	15.17
3 (push)	Penalty	56.25	19.05
	Constraint	65.86	18.37
	Impulse	63.53	20.43
	Rigid body	65.91	19.96

**Table 2.** Perceived realism of haptic rendering methods in the original study for each task.

$F(1.42; 44.13) = 9.14, p = 0.002, \eta^2_p = 0.23$ ). The post-hoc tests show that Task 1 has a significant lower mean than Task 2 ( $p = 0.001; d_z = -0.70$ ; see Table 2). Further, the interaction between haptic rendering method and task is significant ( $F(6, 186) = 5.83, p < 0.001, \eta^2_p = 0.16$ ). The comparison of the haptic rendering methods in relation to the different tasks shows significant results for Task 1 and Task 2. In Task 1 the perceived realism evaluation for *impulse* was significantly higher than for *penalty* ( $p = 0.001; d_z = 0.73$ ), *constraint* ( $p = 0.016; d_z = 0.58$ ) and *rigid body* ( $p = 0.017; d_z = 0.58$ ). In Task 2 the post-hoc tests shows significant differences between *penalty* and *rigid body* ( $p = 0.024; d_z = -0.55$ ) as well as between *constraint* and *impulse* ( $p = 0.023; d_z = 0.55$ ).

The frequencies of ranking order were similar distributed. Only *penalty* was evaluated mostly as worst (17 of 32). The ranking results for the other haptic rendering methods revealed an inconclusive picture with no clear results. There, was a correlation between perceived realism of *impulse* summarized over all tasks and its rank ( $\rho = -0.51, p = 0.003$ ). All other rankings indicate no correlation with the perceived realism of rendering methods.

Presence was rated with a mean of 7.65 ( $SD = 1.52$ ). In total the participants repeated a haptic rendering method in 55 of 384 cases and corrected it in 29 cases. Corrections ranged from 5 to 20 percentage points with a single exception of 55 percentage points. In 33 cases the testing was repeated without correction. Nine times the testing was corrected without repetition. Task 1 was repeated (25 times) and corrected most often (14 times).

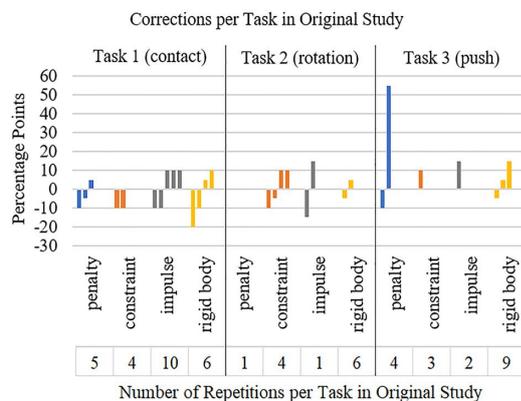


**Figure 5.** Boxplots of perceived realism of haptic rendering methods in the original study. x indicates the mean value, \* indicates a significant difference of  $p < 0.05$ , \*\* indicates a significant difference of  $p < 0.01$ .

Out of the haptic rendering methods *rigid body* with 21 repetitions and 9 corrections was the most repeated and corrected one (see Fig. 6).

**Performance of haptic rendering methods in replication study.** Mean perceived realism values of all rendering methods vary between 51.82 ( $SD = 22.89$ ) and 74.82 ( $SD = 16.60$ ) (see Table 3). The assumption for ANOVA was also checked and confirmed. The two-way ANOVA shows a significant main effect for the differences between the haptic rendering methods ( $F(3; 81) = 4.41, p = 0.006, \eta^2_p = 0.14$ ). The post-hoc tests for all haptic rendering methods independent of the task are significant for the differences between *penalty* and *constraint* ( $p = 0.007; d_z = -0.64$ ) and between *penalty* and *impulse* ( $p = 0.030; d_z = -0.56$ ). *Penalty* was perceived less realistic than all other rendering methods. The main effect for Task is also significant and shows the highest effect size ( $F(2; 54) = 9.60, p < 0.001, \eta^2_p = 0.26$ ). In the post-hoc tests significant differences between Task 1 and Task 2 ( $p < 0.001; d_z = -0.96$ ) and also between Task 2 and Task 3 ( $p = 0.013; d_z = 0.61$ ) are evident. Task 2 shows the highest mean value of perceived realism of all tasks (see Table 3). Further, the interaction between haptic rendering method and task is significant (Greenhouse–Geisser  $F(4.41; 118.97) = 5.02, p < 0.001, \eta^2_p = 0.157$ ). The comparison of the haptic rendering methods in relation to the different tasks show significant results for all tasks. In Task 1 the perceived realism for *impulse* was significantly higher than for *penalty* ( $p = 0.029; d_z = 0.58$ ) and for *rigid body* ( $p = 0.016; d_z = 0.59$ ). In Task 2 the post-hoc tests reveal that *constraint* is rated significantly higher than *penalty* ( $p = 0.001; d_z = 0.79$ ). In Task 3 *penalty* was perceived as significantly less realistic than *constraint* ( $p = 0.006; d_z = -0.68$ ) and *impulse* ( $p = 0.017; d_z = -0.59$ ; see Table 3 or Fig. 7).

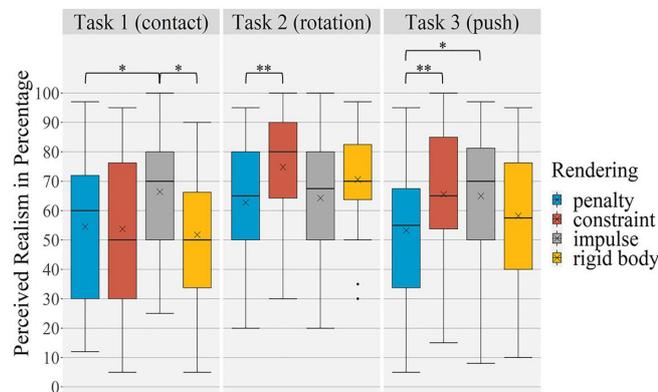
The frequencies of ranking order were similar distributed. Only *penalty* was evaluated mostly as worst (11 of 24). The ranking results for the other haptic rendering methods revealed an inconclusive picture with no clear results. There was a correlation between perceived realism of *rigid body* summarized over all tasks and its



**Figure 6.** Correction values and number of repetitions for each haptic rendering method for each task in the original study.

Task	Rendering	M	SD
1 (contact)	Penalty	54.54	24.62
	Constraint	53.75	26.63
	Impulse	66.43	19.95
	Rigid body	51.82	22.89
2 (rotation)	Penalty	62.75	21.86
	Constraint	74.82	16.60
	Impulse	64.29	22.42
	Rigid body	70.57	16.98
3 (push)	Penalty	53.32	23.77
	Constraint	65.57	24.13
	Impulse	65.00	23.18
	Rigid body	58.25	23.87

**Table 3.** Perceived realism of haptic rendering methods in the replication study for each task.

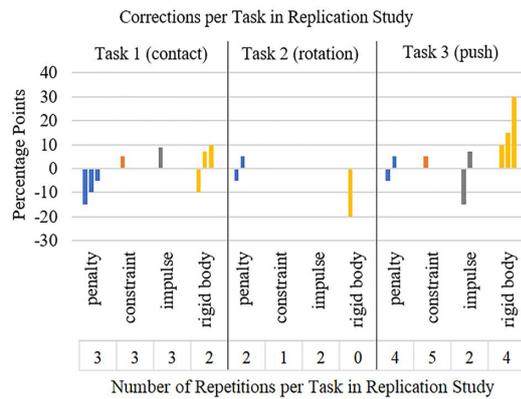


**Figure 7.** Boxplots of perceived realism of haptic rendering methods in the replication study. x indicates the mean value, \* indicates a significant difference of  $p < 0.05$ , \*\* indicates a significant difference of  $p < 0.01$ .

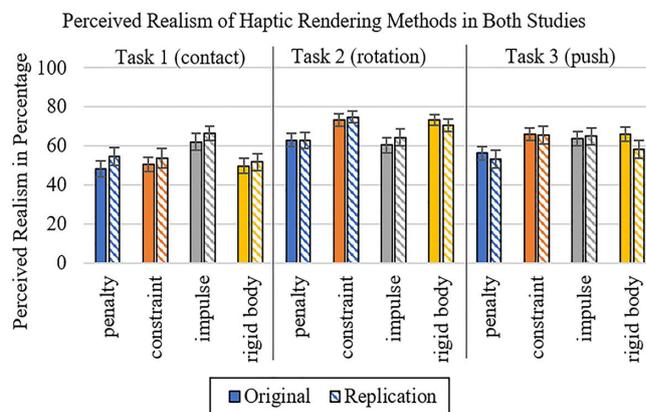
rank ( $\rho = -0.631$ ,  $p = 0.001$ ). All other rankings indicate no correlation with the perceived realism of rendering methods. Three times no ranking was given.

Presence was rated with a mean of 7.50 ( $SD = 1.30$ ). In total the participants repeated a haptic rendering method in 31 of 336 cases and corrected it 20 times. Corrections ranged from 5 to 30 percentage points (see Fig. 8). In 13 cases the testing was repeated without correction. Three times the testing was corrected without repeating. Task 3 was repeated (15 times) and corrected most often (9 times), *penalty* and *constraint* were the most often repeated haptic rendering methods (9 times), *penalty* and *rigid body* were corrected most often (7 times). In the replication study there are also negative correlations of participants height and perceived realism of the haptic rendering methods, showing that taller participants scored lower on the perceived realism. This was found in Task 1 for *penalty* ( $r = -0.514$ ,  $p = 0.006$ ) and *constraint* ( $r = -0.55$ ,  $p = 0.002$ ), in Task 2 for *penalty* ( $r = -0.54$ ,  $p = 0.003$ ) and *impulse* ( $r = -0.54$ ,  $p = 0.003$ ) and in Task 3 for *penalty* ( $r = -0.47$ ,  $p = 0.013$ ) and *rigid body* ( $r = -0.41$ ,  $p = 0.029$ ).

**Performance of haptic rendering methods: original study vs. replication study.** The distribution of means of the descriptive values is very similar for both studies (see Fig. 9). The mixed ANOVA shows no significant effect for the between-subject factor ( $F(1; 58) = 0.05$ ,  $p = 0.829$ ,  $\eta^2_p = 0.00$ ), meaning that there is no significant difference between the original and the replication study at all. The examination of the within-subject factors reveals a significant effect for haptic rendering method (Greenhouse–Geisser  $F(2.43; 140.71) = 7.51$ ,  $p < 0.001$ ,  $\eta^2_p = 0.12$ ) and task (Greenhouse–Geisser  $F(1.55; 89.96) = 16.79$ ,  $p < 0.001$ ,  $\eta^2_p = 0.22$ ). Further, the interaction of haptic rendering method and task is significant ( $F(6; 348) = 10.46$ ,  $p < 0.001$ ,  $\eta^2_p = 0.15$ ). The interactions between haptic rendering method and the two studies (Greenhouse–Geisser  $F(2.43; 140.71) = 0.97$ ,  $p = 0.395$ ,  $\eta^2_p = 0.02$ ), between task and the two studies (Greenhouse–Geisser  $F(1.55; 89.96) = 0.98$ ,  $p = 0.360$ ,  $\eta^2_p = 0.02$ ) and also the interaction between haptic rendering method, task and the two studies together ( $F(6; 348) = 0.31$ ,  $p = 0.934$ ,  $\eta^2_p = 0.01$ ) are not significant.



**Figure 8.** Correction values and number of repetitions for each haptic rendering method for each task in the replication study.



**Figure 9.** Comparison of perceived realism of haptic rendering methods in both studies (error bars = standard error).

**Subjective feedback.** Throughout the study a lot of comments from the subjects were collected to find out what they are thinking about the perceived realism of the haptic rendering method and the study in general. The principal investigator of the original study noticed that about eight subjects operated very careful and slow with the hand-drill. Two other subjects (original study) moved the hand-drill in a rugged way. Eight subjects rotated the tool in huge radiuses (7 in original, 1 in replication study).

The comments about the perceived realism were ordered by task and haptic rendering method to see if participants tended to comment more on the tasks or the haptic rendering methods. Similar comments accumulated for tasks rather than haptic rendering method. In Task 1 of the original study, all rendering methods were described like too soft (“Feels soft”, “much softer than in reality”). Only one person in the original study commented for *impulse*: “feeling that edges are really there”. The rotation performance of haptic rendering methods in Task 2 was partially characterized as “floating”, “swinging” or “elastic” and also “too soft” (original study). The evaluation of *impulse* in Task 2 was described with “scraping a bit” (2 in original study), “feeling like sand in between” (1 in original study) or “the virtual is like vibrating” (3 in original, 1 in replication study). In Task 3 four subjects mentioned for *penalty*, *constraint* and *rigid body* that the end of the haptic object was slipping through the cavity (original study).

Across all tasks two subjects in the original study commented that “all in all a hard stop is missing”. One person in the replication study said at the end of the study: “[*constraint*] and [*impulse*] should be combined, [*impulse*] for getting in contact, [*constraint*] for movement on the surface”. Another subject in the original study said: “[*impulse*] was the best in force but felt very rough as if splinter was inside. [*penalty*] and [*constraint*] felt like rubber”.

In both studies multiple participants expressed their difficulties in forming an overall ranking as they could not make out a clear winner and stated this explicitly. In the replication study this insecurity resulted in the inability of three participants to provide an overall ranking.

**Summary.** In both studies' significant differences of perceived realism in relation to the haptic rendering method and task could be found. Further, the pairwise comparisons of the haptic rendering methods in detail show similar results in both studies. The ranking of rendering methods did not correlate with the perceived realism of the rendering methods in the tasks.

## Discussion

The original and replication study are discussed in Sect. "Original study vs. replication study" before common limitations are given in Sect. "Limitations".

**Original study vs. replication study.** The results for the haptic rendering methods in the original and the replication study are very similar with only miniscule differences. The mean values of perceived realism of all haptic rendering methods are located in the mid-upper area of the scale, with mostly slightly higher values in the replication study. Further, the perceived realism differs in relation to the four haptic rendering methods and the three tasks in both studies, which is underlined by the significant interaction effects between haptic rendering method and task in both studies. In both studies the tasks were rated significantly different with Task 2 emerging as the highest rated, meaning that rotation could be simulated best.

RQ1 "Is any investigated haptic rendering method capable of delivering a realistic haptic feedback?". Although all haptic rendering methods performed fairly well in the original and replication study, there is no haptic rendering method which could be considered delivering an indistinguishable haptic feedback from the interaction with the real haptic object for the given tasks. This is not surprising given the worst-case scenario of a steel/steel contact. Since it is well known that haptic systems become less stable if stiffness is increased<sup>58</sup> and steel is one of the stiffest materials. However, we would assume that in a less stiff scenario a realistic sensation is achievable. Considering the acetabula reaming scenario where a bone-cartilage/steel contact with additional damping from the surrounding soft tissue is present, a realistic haptic feedback seems achievable.

RQ2 "Which investigated haptic rendering methods delivers the highest degree of perceived realism across all tasks?". The results of both studies show that there is no clear best haptic rendering method as neither haptic rendering method performed best in all three tasks. This is further substantiated by the participants' difficulties of forming an overall ranking. However, *penalty* clearly emerged as the worst from both studies, which was to be expected. This method's known stability problems and the reduced realism due to visual artefacts, such as overlapping, clearly led to its worst performance. In addition, the results of the ranking, in which *penalty* was most frequently ranked as worst, confirms the quantitative findings. Though, it clearly emerges, that *constraint* and *rigid body* were rated very similar in every task in both studies. This means that the *rigid body* simulation did not have any benefit over direct geometric projection to solve constraints on the perceived realism.

RQ3 "In which way do the investigated haptic rendering methods differ in perceived realism for different tasks?". Both studies showed that the haptic rendering methods were rated differently for each task. For Task 1 both studies show that *impulse* performed better than the other three haptic rendering methods which scores barely differed. That the steel/steel contact was simulated best by *impulse* confirmed our expectations from literature. The results for Task 2 show a clear grouping for *penalty* and *impulse* in a low rated group and *constraint* and *rigid body* in a high rated group. The fact that *penalty* was rated low was expected, as the visual artefacts are permanently shown in this task. It is interesting, that *impulse* performed worse, as this behavior was not documented before. We suspect that the small impulses that result from momentarily losing and coming into contact did not feel realistic. For Task 3 *penalty* emerged as the worst method from both studies whilst *impulse* and *constraint* performed similarly good. A difference could be found for *rigid body*, which had almost equal results than *constraint* and *impulse* in the original study, but was reported in-between *penalty* and the top group in the replication study. Although, this particular difference of *rigid body* to the top group was not significant, we would recommend focusing on *impulse* and *constraint* when selecting a haptic rendering method for applications resembling Task 3.

The presence ratings were almost identical in both studies and showed that a sufficient presence of the participants in VR could be achieved, despite the minimalist design of the virtual environment.

The analysis of the overall ranking question showed in both studies that the participants had difficulties to give an answer which is evident from their comments and the found correlations. Nonetheless, this backs the quantitative results that there is no best haptic rendering method. Further, the ranking also confirmed that *penalty* gave the worst haptic feedback.

In terms of repetitions there were less in the replication study but with a higher correction rate than in the original study. Though, both studies showed similar moderate correction values which resulted in the same conclusion, that the option for repeating and correcting the rating might not have been necessary.

The most surprising difference between the original and the replication study was the negative correlation of the participants height on the perceived realism of the haptic rendering methods in the replication study which was not present in the original study. The reasons therefore are highly speculative but might lay in the fixed height of the table respectively the arm rests of the chair used for the replications study, as the one used in the original study had none. Aside from this, the samples did not differ in any other demographic variable except age which, however, did not had an influence on the results.

Lastly, the comments from the participants and the observations from the principal investigators in both studies were similar with no particular difference. This further substantiates the conclusions drawn from both studies.

Summarizing, we can state that no haptic rendering method was able to deliver a sensation of a real steel/steel interaction, although a realistic behavior in less demanding scenarios seems likable. In any case we would recommend not to use *penalty*. Instead, implementations should concentrate on combining *impulse* with *constraint* or *rigid body* to achieve a realistic hard contact behavior in conjunction with a smooth rotation and force built up. Though, the detection of the transition point and the assurance of a consistent haptic behavior will be a challenging feat.

**Limitations.** The four haptic rendering methods were only investigated on one haptic device, the Virtuouse 6D. The findings of this study should therefore be investigated with other haptic devices. Currently, only robots like the KUKA LBR iiwa are capable of delivering similar high forces. Our findings are based on three simple tasks. Future studies should incorporate more complex movements. The order of the tasks in this experiment was not randomized to reflect the sequence during acetabula reaming. Further, the measure to give an overall ranking at the end of the experiment requested prevented a randomization of haptic rendering methods between tasks. However, we could not detect a bias of data. The position of the virtual haptic object had to be slightly adjusted manually in order to achieve a sufficient alignment with the real haptic object. Providing a visual cue was essential to ensure that the participants hit the cavity with the hand-drill. Visual distraction and potential slight misalignment between the virtual and real haptic object caused by the manually adjusted alignment could have biased the perceived realism evaluation especially for Task 1 (correct position of the hand-drill for first contact with the cavity), and Task 3 (green highlighted virtual cavity to mark the maximum position). To control this, presence was measured. The presence ratings in both studies showed that the participants felt no or only little irritations in VR. Future studies should address the pure haptic perception. In four cases the software had to be restarted due to technical problems. Two subjects did the study without white noise. Some subjects required a break or took off the headsets for a second. In VR both haptic objects looked identical, but different from the real haptic object. The 3D model of the haptic object was oversimplified as there was no guiding mechanism for uniaxial movement visible. Although, no participant commented on this, it cannot be entirely ruled out, that a different behavior was subconsciously expected. However, as the visual and haptic behavior of the real and virtual haptic object were always consisted a potential bias would have affected both conditions. The different instructions when to stop pushing in Task 3 (blocking vs. highlighting), might have influenced the participants scoring. Despite raising stiffness to the technical maximum of the Virtuouse 6D it was impossible to simulate an equal hard blocking for the virtual rendering condition as for the real haptic object. We assumed that many participants might not recognize this and keep on pushing, ultimately over pushing the Virtuouse 6D. However, we instructed the participants to only include the pushing but not the blocking at the end into to their scoring. Nonetheless, a bias of the ratings cannot be ruled out. However, even if it had a negative effect it was the same for all haptic rendering methods in Task 3. We checked the data for a potential influence in comparison to Task 1 and Task 2. However, no obvious influence was visible. In case there was potential negative bias it would be the same for all haptic rendering methods in Task 3. For future studies the visualization of the real haptic object in VR should be highlighted green so that the visual feedback matches.

## Conclusion

Our findings suggest that for bimanual high force tasks a realistic haptic feedback can be achieved, although not with a single haptic rendering method and for worst-case scenarios like a steel/steel interaction. We could also show that the task plays an important role on the perceived realism. *Penalty* clearly emerged as the worst haptic rendering whilst there was no clear best. *Impulse* performed best for hard contact simulation. However, for simulating rotations and 'pushing-in'-tasks with an increasing force *constraint* and *rigid body* delivered better results. Therefore, a combination of haptic rendering methods seems most promising. The methodical approach of performing an original study and confirming the results with a replication study significantly substantiate our conclusions, as only miniscule differences were found.

## Data availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 20 February 2023; Accepted: 5 July 2023

Published online: 11 July 2023

## References

- Escobar-Castillejos, D., Noguez, J., Neri, L., Magana, A. & Benes, B. A review of simulators with haptic devices for medical training. *J. Med. Syst.* **40**, 104. <https://doi.org/10.1007/s10916-016-0459-8> (2016).
- Dawe, S. R. *et al.* A systematic review of surgical skills transfer after simulation-based training. Laparoscopic cholecystectomy and endoscopy. *Ann. Surg.* **259**, 236–248. <https://doi.org/10.1097/SLA.000000000000245> (2014).
- Park, C. H., Wilson, K. L. & Howard, A. M. Examining the learning effects of a low-cost haptic-based virtual reality simulator on laparoscopic cholecystectomy. In Rodrigues, P. P. (ed) *IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS), 2013. 20 - 22 June 2013, University of Porto, Portugal* (IEEE, 2013), pp. 233–238.
- Zendejas, B., Brydges, R., Hamstra, S. J. & Cook, D. A. State of the evidence on simulation-based training for laparoscopic surgery. A systematic review. *Ann. Surg.* **257**, 586–593. <https://doi.org/10.1097/SLA.0b013e318288c40b> (2013).
- Garre, C. & Otaduy, M. A. Haptic rendering of objects with rigid and deformable parts. *Comput. Graph.* **34**, 689–697. <https://doi.org/10.1016/j.cag.2010.08.006> (2010).
- Otaduy, M. A., Okamura, A. & Subramanian, S. Haptic technologies for direct touch in virtual reality. In *ACM SIGGRAPH 2016 Courses* (eds Otaduy, M. A. *et al.*) 1–123 (ACM, 2016).

7. Kaluschke, M. et al. A shared haptic virtual environment for dental surgical skill training. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)* (IEEE, 2021), pp. 347–352.
8. Al-Saud, L. M. The utility of haptic simulation in early restorative dental training: A scoping review. *J. Dent. Educ.* **85**, 704–721. <https://doi.org/10.1002/jdd.12518> (2021).
9. Tavakoli, M., Patel, R. V. & Moallem, M. A haptic interface for computer-integrated endoscopic surgery and training. *Virtual Real.* **9**, 160–176. <https://doi.org/10.1007/s10055-005-0017-z> (2006).
10. Takayesu, J. K., Nadel, E. S., Bhatia, K. & Walls, R. M. Incorporating simulation into a residency curriculum. *CJEM* **12**, 349–353. <https://doi.org/10.1017/S1481803500012458> (2010).
11. Kamel, M., Eltahawy, E. A., Warford, R., Thrush, C. R. & Noureldin, Y. A. Simulation-based training in urology residency programmes in the USA: Results of a nationwide survey. *Arab. J. Urol.* **16**, 446–452. <https://doi.org/10.1016/j.aju.2018.06.003> (2018).
12. Syamlan, A. et al. Haptic/virtual reality orthopedic surgical simulators: A literature review. *Virtual Real.* <https://doi.org/10.1007/s10055-022-00666-y> (2022).
13. Pelliccia, L. et al. A cadaver-based biomechanical model of acetabulum reaming for surgical virtual reality training simulators. *Sci. Rep.* **10**, 14545. <https://doi.org/10.1038/s41598-020-71499-5> (2020).
14. 3D Systems GmbH. *TOUCH* (3D Systems GmbH, 2022).
15. Force Dimension. *Omega* (Force Dimension, 2022).
16. 3D Systems GmbH. *PHANTOM* (3D Systems GmbH, 2022).
17. Force Dimension. *Sigma* (Force Dimension, 2022).
18. Haption, S. A. *Virtouse6D* (HAPTION S.A., 2022).
19. Sagardia, M. et al. VR-OOS. In *IEEE Aerospace Conference, 2015. 7 - 14 March 2015, Big Sky, MT* (IEEE, 2015), pp. 1–17.
20. Kaluschke, M. et al. A virtual hip replacement surgery simulator with realistic haptic feedback. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (IEEE, 2018), pp. 759–760.
21. Kaluschke, M. et al. HIPS - A virtual reality hip prosthesis implantation simulator. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (IEEE, 2018), pp. 591–592.
22. Knopp, S., Lorenz, M., Pelliccia, L. & Klimant, P. Using industrial robots as haptic devices for VR-training. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (IEEE, 2018), pp. 607–608.
23. Wang, Y., Feng, L. & Andersson, K. Haptic force rendering of rigid-body interactions: A systematic review. *Adv. Mech. Eng.* **13**, 168781402110415. <https://doi.org/10.1177/16878140211041538> (2021).
24. Open Science Collaboration. Psychology. Estimating the reproducibility of psychological science. *Science* **349**, aac4716. <https://doi.org/10.1126/science.aac4716> (2015).
25. Swan II, J. E. The replication crisis in empirical science: implications for human subject research in mixed reality. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (IEEE, 2018), pp. xxxvi.
26. Ziadeh, T., Perret, J., Kaluschke, M., Knopp, S. & Lorenz, M. Review of haptic rendering techniques for hip surgery training. In *EuroVR 2020 Application, Exhibition & Demo Track* (eds Helin, K. et al.) 29–39 (VT Technical Research Centre of Finland, 2020).
27. Xia, P. New advances for haptic rendering: state of the art. *Vis. Comput.* **34**, 271–287. <https://doi.org/10.1007/s00371-016-1324-y> (2018).
28. Xu, H. & Barbic, J. Adaptive 6-DoF haptic contact stiffness using the gauss map. *IEEE Trans. Haptics* **9**, 323–332. <https://doi.org/10.1109/TOH.2016.2558185> (2016).
29. Dangxiao, W., Xin, Z., Yuru, Z. & Jing, X. Configuration-based optimization for six degree-of-freedom haptic rendering for fine manipulation. *IEEE Trans. Haptics* **6**, 167–180. <https://doi.org/10.1109/TOH.2012.63> (2013).
30. Li, Y., Tang, M., Zhang, S. & Kim, Y. J. Six-degree-of-freedom haptic rendering using translational and generalized penetration depth computation. In *2013 World Haptics Conference (WHC)* (IEEE, 2013), pp. 289–294.
31. Kaluschke, M., Weller, R. & Zachmann, G. A volumetric penetration measure for 6-DOF haptic rendering of streaming point clouds. In *2017 IEEE World Haptics Conference (WHC)* (IEEE, 2017), pp. 511–516.
32. Barbic, J. & James, D. L. Six-DoF haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Trans. Haptics* **1**, 39–52. <https://doi.org/10.1109/TOH.2008.1> (2008).
33. McNeely, W. A., Puterbaugh, K. D. & Troy, J. J. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '99* (ed. Waggenspack, W.) 401–408 (ACM Press, 1999).
34. Ortega, M., Redon, S. & Coquillart, S. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Trans. Visual. Comput. Graph.* **13**, 458–469. <https://doi.org/10.1109/TVCG.2007.1028> (2007).
35. Ryden, F. & Chizeck, H. J. A method for constraint-based six degree-of-freedom haptic interaction with streaming point clouds. In *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 2353–2359.
36. Chan, S., Conti, F., Blevins, N. H. & Salisbury, K. Constraint-based six degree-of-freedom haptic rendering of volume-embedded isosurfaces. In *2011 IEEE World Haptics Conference* (IEEE, 2011), pp. 89–94.
37. Constantinescu, D., Salcudean, S. E. & Croft, E. A. Haptic rendering of rigid contacts using impulsive and penalty forces. *IEEE Trans. Robot.* **21**, 309–323. <https://doi.org/10.1109/TRO.2004.840906> (2005).
38. Mirtich, B. & Canny, J. Impulse-based simulation of rigid bodies. In *Proceedings of the 1995 symposium on Interactive 3D graphics—SI3D '95* (ed. Zyda, M.) 181–ff (ACM Press, 1995).
39. Kuchenbecker, K. J., Fiene, J. & Niemeyer, G. Improving contact realism through event-based haptic feedback. *IEEE Trans. Visual. Comput. Graph.* **12**, 219–230. <https://doi.org/10.1109/TVCG.2006.32> (2006).
40. NVIDIA Corporation. *PhysX | GeForce* (NVIDIA Corporation, 2021).
41. Coumans, E. *Bullet Real-Time Physics Simulation* (Bullet Physics, 2022).
42. Microsoft. *Havok Physics* (Microsoft, 2022).
43. Smith, R. *Open Dynamics Engine* (Smith, Russell, 2004).
44. Catto, E. *Box2D. Box2D*
45. Bender, J., Erleben, K. & Trinkle, J. Interactive simulation of rigid body dynamics in computer graphics. *Comput. Graph. Forum* **33**, 246–270. <https://doi.org/10.1111/cgf.12272> (2014).
46. Ang, L. Y. L., Koh, Y. K. & Lee, H. P. The performance of active noise-canceling headphones in different noise environments. *Appl. Acoust.* **122**, 16–22. <https://doi.org/10.1016/j.apacoust.2017.02.005> (2017).
47. Kaluschke, M. et al. Realistic haptic feedback for material removal in medical simulations. In *2020 IEEE Haptics Symposium (HAPTICS)* (IEEE2020), pp. 920–926.
48. Lang, J. A new stereotest. *J. Pediatr. Ophthalmol. Strabismus* **20**, 72–74. <https://doi.org/10.3928/0191-3913-1983031-08> (1983).
49. Park, C., Park, J., Oh, S. & Choi, S. Realistic haptic rendering of collision effects using multimodal vibrotactile and impact feedback. In *2019 IEEE World Haptics Conference (WHC)* (IEEE2019), pp. 449–454.
50. Loconsole, C. et al. A fully immersive VR-based haptic feedback system for size measurement in inspection tasks using 3D point clouds. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (IEEE2017), pp. 758–763.
51. Tasaka, S., Harada, T., Kawamura, H., Fukuda, K. & Ikeda, A. Analysis of haptic perception of physical therapists using end-feel training robot. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (IEEE2019), pp. 4101–4106.

52. Fucentese, S. F. *et al.* Evaluation of a virtual-reality-based simulator using passive haptic feedback for knee arthroscopy. *Knee Surg. Sports Traumatol. Arthrosc. Off. J. ESSKA* **23**, 1077–1085. <https://doi.org/10.1007/s00167-014-2888-6> (2015).
53. Garfield Roberts, P. *et al.* Validation of the updated ArthroS simulator: Face and construct validity of a passive haptic virtual reality simulator with novel performance metrics. *Knee Surg. Sports Traumatol. Arthrosc. Off. J. ESSKA* **25**, 616–625. <https://doi.org/10.1007/s00167-016-4114-1> (2017).
54. Bartlett, J. D., Lawrence, J. E. & Khanduja, V. Virtual reality hip arthroscopy simulator demonstrates sufficient face validity. *Knee Surg. Sports Traumatol. Arthrosc. Off. J. ESSKA* **27**, 3162–3167. <https://doi.org/10.1007/s00167-018-5038-8> (2019).
55. Bouchard, S. *et al.* Reliability and validity of a single-item measure of presence in VR. In *Proceedings Second International Conference on Creating, Connecting and Collaborating through Computing (IEEE2004)*, pp. 59–61.
56. Skarbez, R., Brooks, J. F. P. & Whitton, M. C. A survey of presence and related concepts. *ACM Comput. Surv.* **50**, 1–39. <https://doi.org/10.1145/3134301> (2018).
57. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences* Lawrence Erlbaum Associates 2nd edn. (Lawrence Erlbaum Associates, 1988).
58. Hulin, T., Albu-Schäffer, A. & Hirzinger, G. Passivity and stability boundaries for haptic systems with time delay. *IEEE Trans. Contr. Syst. Technol.* **22**, 1297–1309. <https://doi.org/10.1109/TCST.2013.2283372> (2014).

### Acknowledgements

The authors would like to thank Georg Matschewsky for manufacturing the steel parts and Franziska Klimant for proof reading. The project on which this paper is based was funded by the German Federal Ministry of Education and Research under the funding code 16SV8356. The responsibility for the content of this publication lies with the author.

### Author contributions

M.L. participated in the conception and design of the research question, the design and coordination of the study, data collection, the technical implementation, creating the graphics and interpreting the results. A.H. participated in the design of the study, statistical analysis, creating the graphics and interpreting the results. M.K. participated in the design and coordination of the study, data collection, the technical implementation, and interpreting the results. T.Z. participated in the technical implementation and interpreting the results. N.P. participated in the design and coordination of the study, data collection and interpreting the results. M.K. participated in the design and coordination of the study, data collection and interpreting the results. J.P. participated in the technical implementation and interpreting the results. S.K. participated in the technical implementation and interpreting the results. A.D. participated in the data collection, statistical analysis, interpreting the results and provided critical comments on the manuscript. P.K. participated in the technical implementation and provided critical comments on the manuscript. G.Z. participated in the conception and design of the research question, data collection and interpreting the results. A.C.B. participated in interpreting the results and provided critical comments on the manuscript. All authors were involved in drafting the manuscript. All of them read and approved the final version of the manuscript.

### Funding

Open Access funding enabled and organized by Projekt DEAL.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to M.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023

THE EFFECT OF 3D STEREOPSIS AND HAND-TOOL ALIGNMENT ON LEARNING EFFECTIVENESS AND SKILL TRANSFER OF A VR-BASED SIMULATOR FOR DENTAL TRAINING

*Outline*

---

	<b>Abstract</b> . . . . .	247
1	<b>Introduction</b> . . . . .	247
2	<b>Related work</b> . . . . .	249
3	<b>Simulator</b> . . . . .	251
	3.1 Stereo rendering . . . . .	252
	3.2 Hand-tool alignment . . . . .	252
	3.3 Visual perception . . . . .	254
	3.4 Automated outcome scoring . . . . .	255
4	<b>User study</b> . . . . .	258
5	<b>Results</b> . . . . .	259
	5.1 Groups . . . . .	260
	5.2 3D rendering modes . . . . .	261
	5.3 Hand-tool alignment . . . . .	262
	5.4 Suitability for assessment . . . . .	263
	5.5 Learning transfer . . . . .	264
	5.6 Eye-tooth distance . . . . .	265
6	<b>Discussion</b> . . . . .	266
7	<b>Conclusion</b> . . . . .	269
	<b>Author Contributions</b> . . . . .	269
	<b>References</b> . . . . .	270

---

*Bibliography Entry*

Kaluschke, M., Yin, M. S., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2023). **The Effect of 3D Stereopsis and Hand-Tool Alignment on Learning Effectiveness and Skill Transfer of a VR-based Simulator for Dental Training.** *PLOS ONE*, 18(10):e0291389

*Author's Contribution*

The author designed and planned the user-study, implemented the simulator, data logging and all other software, ran the study, did the data analysis, wrote the manuscript and reviewed and edited the manuscript.

*Copyright*

Copyright © 2023 Kaluschke et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## RESEARCH ARTICLE

# The effect of 3D stereopsis and hand-tool alignment on learning effectiveness and skill transfer of a VR-based simulator for dental training

Maximilian Kaluschke<sup>1</sup>, Myat Su Yin<sup>2</sup>, Peter Haddawy<sup>2\*</sup>, Siriwan Suebnukarn<sup>3</sup>, Gabriel Zachmann<sup>1</sup>

1 Computer Graphics and Virtual Reality, University of Bremen, Bremen, Germany, 2 Faculty of ICT, Mahidol University, Bangkok, Thailand, 3 Faculty of Dentistry, Thammasat University, Bangkok, Thailand

\* [peter.had@mahidol.ac.th](mailto:peter.had@mahidol.ac.th)



## OPEN ACCESS

**Citation:** Kaluschke M, Yin MS, Haddawy P, Suebnukarn S, Zachmann G (2023) The effect of 3D stereopsis and hand-tool alignment on learning effectiveness and skill transfer of a VR-based simulator for dental training. PLoS ONE 18(10): e0291389. <https://doi.org/10.1371/journal.pone.0291389>

**Editor:** Artak Heboyan, Yerevan State Medical University Named after Mkhitar Heratsi, ARMENIA

**Received:** December 28, 2022

**Accepted:** August 29, 2023

**Published:** October 4, 2023

**Peer Review History:** PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pone.0291389>

**Copyright:** © 2023 Kaluschke et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data Availability Statement:** The complete dataset, and R-script files are available from the FigShare database (URL <https://figshare.com/s/>)

## Abstract

Recent years have seen the proliferation of VR-based dental simulators using a wide variety of different VR configurations with varying degrees of realism. Important aspects distinguishing VR hardware configurations are 3D stereoscopic rendering and visual alignment of the user's hands with the virtual tools. New dental simulators are often evaluated without analysing the impact of these simulation aspects. In this paper, we seek to determine the impact of 3D stereoscopic rendering and of hand-tool alignment on the teaching effectiveness and skill assessment accuracy of a VR dental simulator. We developed a bimanual simulator using an HMD and two haptic devices that provides an immersive environment with both 3D stereoscopic rendering and hand-tool alignment. We then independently controlled for each of the two aspects of the simulation. We trained four groups of students in root canal access opening using the simulator and measured the virtual and real learning gains. We quantified the real learning gains by pre- and post-testing using realistic plastic teeth and the virtual learning gains by scoring the training outcomes inside the simulator. We developed a scoring metric to automatically score the training outcomes that strongly correlates with experts' scoring of those outcomes. We found that hand-tool alignment has a positive impact on virtual and real learning gains, and improves the accuracy of skill assessment. We found that stereoscopic 3D had a negative impact on virtual and real learning gains, however it improves the accuracy of skill assessment. This finding is counter-intuitive, and we found eye-tooth distance to be a confounding variable of stereoscopic 3D, as it was significantly lower for the monoscopic 3D condition and negatively correlates with real learning gain. The results of our study provide valuable information for the future design of dental simulators, as well as simulators for other high-precision psycho-motor tasks.

## 1 Introduction

Development of expertise in dentistry requires extensive training of specific dexterous skills. A dental surgeon's skill increases with practice, as evidenced by the strong correlation between

Geddea3f66489716fba0, DOI number [10.6084/m9.figshare.22362322](https://doi.org/10.6084/m9.figshare.22362322).

**Funding:** This work was partially supported by a grant from the Mahidol University Office of 638 International Relations to Haddawy in support of the Mahidol-Bremen Medical 639 Informatics Research Unit (MIRU), and by a fellowship from the Hanse-Wissenschaftskolleg 640 Institute for Advanced Study. No additional external funding was received for this study.

**Competing interests:** The authors have declared that no competing interests exist.

dental surgeon skill and practice time [1, 2]. Therefore, dental schools have long used different forms of simulation to provide students with practise opportunities for these particular skills before ever practicing on live patients. The early, and still most commonly used, simulators consist of a mannequin head (so-called phantom head) with plastic teeth. These simulators can be used by the students to practice various procedures. Depending on the procedure, the teeth are either simple inexpensive solid plastic teeth (for simple procedures such as caries removal) or more expensive plastic teeth with different layers and internal anatomy (for complex procedures, such as root canal access opening). Upon completion of a procedure, a dental instructor scores the outcome based on visual inspection. As the teeth are significantly altered during practice, they can only be used effectively for a single time, resulting in high operational cost.

In recent years, VR-based dental simulators have increased in popularity due to enabling technological advancements, combined with concrete benefits of the approach [3–5]. VR simulators offer high-fidelity simulations that are reusable, resulting in considerably lower operational costs, and they can be configured to provide trainees practice on a variety of different cases. They also have the ability to record accurate data on individual performance, which provides the opportunity for trainees to receive objective feedback to facilitate learning. VR simulators show significant real-world learning effects for the virtually trained surgical procedures [6–8]. In addition, medical trainers' growing need for objective and automated assessment tools [9, 10] could be addressed by VR-based simulators. In contrast to plastic teeth, simulator outcomes can be scored automatically [11], provided the simulator is suitably designed and implemented.

The requirements for a VR simulator to be an effective teaching tool and to be an effective assessment tool are closely related but distinct. To be an effective teaching tool, practice time in the simulator must translate into significant improvement in real-world performance. To be an effective assessment tool, real-world skill level must translate into simulator performance, without requiring significant time to learn the idiosyncrasies of the simulator. It is possible for a simulator to satisfy one of the requirements but not the other. For example, a simulator that is difficult to use may still result in real-world performance gains, yet not be useful as an assessment tool. This was the case in some early VR dental simulators that displayed results on a 2D monitor [12]. These two requirements can be thought of in terms of two types of transferability: simulator to the real-world and real-world to the simulator.

With the variety of VR technologies available, dental simulators have been developed using a wide variety of different VR configurations. Display technologies used include traditional 2D monitors, 3D monitors, half-mirrored displays, and head-mounted displays (HMD), the latter three of which provide stereoscopic depth perception. Instrument manipulation is achieved with and without haptic feedback. In addition, the use of HMDs and half-mirrored displays supports hand-tool alignment in which the user sees the dental instrument in the same location as their physical hand. In contrast, 2D and 3D monitors do not provide such alignment. While each new dental simulator typically is associated with some form of evaluation study [13–15], only few comparative studies have been carried out to determine the benefits of the simulation aspects associated with the various available VR technologies being used and none examine the impact of those factors on teaching effectiveness or assessment suitability, as measured by transferability.

In this paper we examine the impact of the two major selling features of HMDs for virtual simulators, 3D stereoscopic rendering and hand-tool alignment, on the teaching effectiveness and the suitability as an assessment tool of a VR dental simulator. These features are not possible to achieve with traditional monitors or 3D monitors. Students were trained on an immersive VR simulator while systematically and independently controlling for each of the two

aspects of the simulation. Learning gains were measured in two ways, by doing pre- and post-testing on realistic plastic teeth, as well as by assessing the virtual training outcomes using a novel automated scoring metric.

We measure teaching effectiveness in terms of learning gains between pre- and post-testing on realistic plastic teeth. We measure suitability for assessment, in terms of the correlation between real-world pre-testing score and the virtual score of the first simulator session following the pre-testing. Based on these metrics and the two VR technology aspects (stereoscopic 3D vision and hand-tool alignment), we formulate four hypotheses:

$H_{V_{learn}}$  Stereoscopic vision has a positive impact on the learning effectiveness of the simulator, as measured by real learning gains.

$H_{A_{learn}}$  Hand-tool alignment has a positive impact on the learning effectiveness of the simulator, as measured by real learning gains.

$H_{V_{assess}}$  Stereoscopic vision has a positive impact on the simulator's suitability for assessment, as measured by initial simulator performance.

$H_{A_{assess}}$  Hand-tool alignment has a positive impact on the simulator's suitability for assessment, as measured by initial simulator performance.

## 2 Related work

With an increasing trend of using 3D stereo-projected images to create realistic virtual learning environments, there is an ongoing debate as to whether stereo-projected images are a necessary feature of simulators [16–18]. A comprehensive review conducted by McIntire et al. [16], found that in 15% of over 180 experiments from 160 publications, stereoscopic 3D display either showed a marginal benefit over a 2D display or the results were mixed or unclear, while in 25% of experiments, stereoscopic 3D display showed no benefit over non-stereo 2D viewing. They concluded that stereoscopic 3D displays are most useful for tasks involving the manipulation of objects and for finding/identifying/classifying objects or imagery. The majority of these studies used 3D monitors for the stereoscopic 3D condition and displayed the same image to both eyes or used 2D monitors for the monoscopic 3D condition. Buckthought et al. [19] showed that dynamic perspective changes enhance depth ordering performance. Therefore, the depth information conveyed through monoscopic 3D inside an HMD which can be freely moved and moved closer and further could provide more helpful depth information when compared to 2D monitors, as the dynamic perspective changes provide depth cues.

de Boer et al. [20] investigate the differences in students' performance in carrying out manual dexterity exercises with the Simodont dental trainer simulator (The MOOG Industrial Group; [www.moog.com](http://www.moog.com)) in 2D and 3D versions. 3D vision in the dental trainer was based on the projection of two images superimposed onto the same screen through a polarising filter. 2D vision was obtained by turning off one of the two projectors such that only one image was projected onto the screen. All of the students in both the 2D and 3D vision groups wore polarised glasses during the practice sessions and when testing to keep the environmental factors constant. The task consisted of using a dental drill to remove material from a cube and the outcomes were automatically scored. The results showed that students working with 3D vision achieved significantly better results than students who worked in 2D. In an administered questionnaire, participants also indicated that they preferred the 3D vision setting. Students reported having an unpleasant experience in working with 2D vision while wearing the glasses. The probable reason is that only one eye received an image through the polarized glasses. In a

related study, Al-Saud and colleagues [21] examined the effects of stereopsis on dentists' performance with the Simodont dental simulator. Thirteen qualified dentists were recruited and asked to perform a total of four different dental manual dexterity tasks under non-stereoscopic and stereoscopic vision conditions with direct and indirect (mirror) observation. The tasks consisted of removal of material from a geometric shape embedded in a cube of material. Automated scoring was based on amounts of target and non-target material removed. Stereoscopic 3D was the simulator's normal operation and was achieved as in the previously mentioned study [20]. To produce 2D images, the simulator was engineered to output a single image to both eyes. The study found out that depth related errors were significantly higher under non-stereoscopic viewing but lateral errors did not differ between conditions. Both studies used the commercial Simodont simulator on a 3D monitor (which displays monoscopic 3D in one condition, thereby acting like a 2D monitor). 3D monitors do not allow for unrestricted head tracking and do not support hand-tool alignment, which our simulator supports through the use of HMD and calibration of the haptic devices.

Collaco et al. [22] investigated the effects of (full) immersion and haptic feedback on inferior alveolar nerve anesthesia technical skills training. Their experimental study consists of preceptorship and training phases. During the preceptorship phase, one of the groups received the anesthesia instructions from the dental instructor on a full HD TV screen, while the participants from the remaining three groups observed the anesthesia procedure from the instructor's perspective in an immersive condition using the HMD. In the training phase, the participants in one of the groups in the immersive condition during the previous preceptorship stage performed the anesthesia injection using the full HD TV screen while the remaining three groups performed the task with the HMD in the immersive condition. The results showed that participants without immersive displays had less accurate needle insertion points, though needle injection angle and depth were not significantly different between the groups. The needle insertion point here needs to be found without haptic feedback. As such it differs considerably from the root canal opening, since the bur can touch the tooth with drilling disabled for orientation with the help of haptic feedback. Due to these differences we expect stereo vision to have a smaller positive effect on performance and on learning.

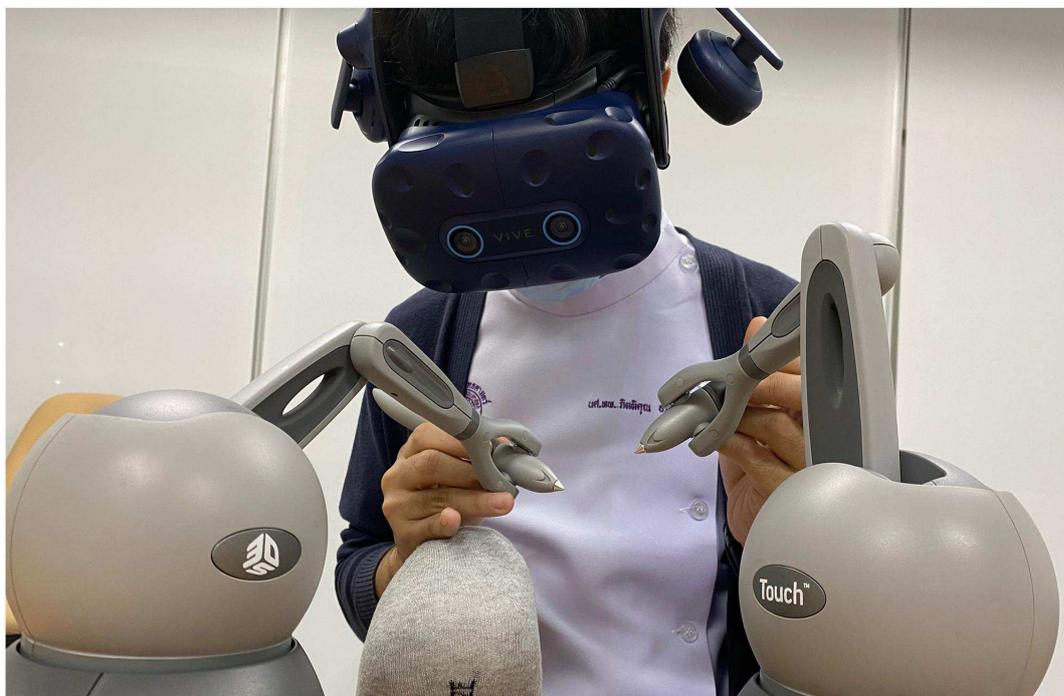
In manipulating tools, users receive information from two feedback loops: the body-related proximal feedback loop (proximal action effect) such as tactile sensations from the moving hand, and from the effect in distal space, such as the visual feedback from the movement of effect points of the tool (distal action effect). Establishing the mapping between the moving hand and the moving effect part of the tool can add challenges to the human information processing systems. According to Sutter et al. [23], if information from proximal and distal feedback loops are equally important for controlling actions, any discrepancy between them would be a constant source of interference to the user. Users of conventional desktop-based VR simulators using haptic interfaces are familiar with this scenario while manipulating the haptic device and observing the action effects on a display monitor. Meanwhile, in HMD simulators, the spatial gap between the hand and the resulting movement can be eliminated by manipulating the virtual camera position and rotation to the user in such a way that the user sees and feels as if he is manipulating the dental tools on the patient's teeth. Although more realistic, it is interesting to note that in this condition the vision may be afforded with a higher weighting than other sensory information; a situation often referred to as visual capture. Although visual information is invaluable for executing skillful manual tasks, visual capture can produce powerful illusory effects with individuals misjudging the size and position of their hands. Moreover, if vision of the hand/tool is available in the operating area it should be recognized that there might well be interference that would impair motor performance and learning, as there is a shift in attentional focus to the outcome of actions rather than the actions themselves.

Wilkie et al. [24] studied whether visual capture can interfere with an individual's rate of motor learning in a laparoscopic surgery setting. They investigated the adaptation to distorted visual feedback in two groups: a direct group directly viewed the input device, while an indirect group used the same input device but viewed their movements on a remote screen. When distortion exists between hand and tool movement, then visual capture is an issue and participants in the indirect group performed better than those in the direct group. However, when no distortions were applied, participants in the direct group performed better than participants in the indirect group. In the dental domain, there is typically no distortion present for drilling tasks. Similarly, Sutter et al. [23] conducted several experiments aiming to investigate the underlying motor and cognitive processes and the limitations of visual predominance in tool actions. Their major finding is that when transformations are in effect the awareness of one's own actions is quite low. These findings suggest that hand-tool alignment will have a profound effect in our user study on learning effect and performance.

The effect of stereoscopic vision inside an HMD on dental surgery simulator suitability for assessment, user performance, and skill transfer have not been investigated previously. Even in the context of arbitrary use-cases, stereoscopic 3D inside HMDs has not been investigated systematically by using the same technology but removing the depth cue of stereopsis. Additionally, the effects of hand-tool alignment have also not been investigated yet, although it is a prominent feature in modern dental surgery simulators. This study attempts to fill both of these gaps.

### 3 Simulator

We developed a VR dental surgery simulator with haptic feedback, in which students can practice caries removal, crown preparation, and root canal access opening (see Fig 1, for the students' perspective see Fig 2). The simulator was developed using Unreal Engine (UE) 4.26. An HTC Vive Pro Eye with a combined resolution of  $2880 \times 1600$  and eye sensors was used to display stereo images from the UE SteamVR plugin. Eye tracking of the HMD user was implemented using the SRanipal Unreal plugin. The dental virtual hand-piece and mirror are each controlled by a Geo-Magic Touch haptic device (Phantom) with 6 degrees-of-freedom (DOF) input and 3 DOF output. Haptic feedback is provided to simulate the interaction between the hand piece and virtual tooth. The sound of the drill is also simulated. The virtual patient was modeled using the Metahuman framework [25] and imported into our UE scene. The virtual human is rendered with high fidelity visuals including subtle idle animations of the face and mouth, such as eye blinking and movement of the tongue. We made sure to not include animations that would alter the location of the tooth. We added a transparency texture to the virtual teeth texture, which allows us to hide one of the teeth (#36) of the Metahuman model. In its place, we inserted a new tooth that we modeled by hand with guidance from CT scans of similar teeth and approved by an expert dentist. At runtime, we render the tooth by using the UE Procedural Mesh Component (PMC). We generate triangles of modified tooth regions in a CUDA library, which are then fed to UE's PMC. The library approximates the tooth surface by a metaball surface that is discretized at runtime using a parallel marching cubes implementation with a resolution of  $90 \times 135 \times 90$ . We compute the haptic feedback outside of the UE main loop, so as not to be limited by the rendering frame rate. The force is computed according to the algorithm presented in [26], which uses an inner spheres volume representation. The tooth enamel is made up of 100k, the dentin by 170k, and the pulp by 10k spheres. We tuned the force, drilling, and friction parameters by our subjective impression of drilling the real plastic teeth that students usually practice on, with approval by an expert dentist. These plastic teeth closely resemble the feeling of drilling real teeth and are anatomically correct.



**Fig 1.** The VR dental surgery simulator is used by a dentistry student to practice root canal access opening on tooth #36. The VR HMD and haptic input/output devices allow for an intuitive control with realistic haptic feedback (in alignment condition). The monitor shows the image that the student is seeing on the HMD.

<https://doi.org/10.1371/journal.pone.0291389.g001>

### 3.1 Stereo rendering

The standard VR rendering is set up to be at a realistic scale, such that the user has a natural stereo impression from the two different images that are sent to the eye. This setting will later be referenced as the “stereo” condition during the user study. To investigate the effect that stereo vision has on the learning effect, we implemented a rendering mode that renders the virtual scene without stereoscopy. We implemented monoscopic 3D by rendering identical images for the left and right eye. This setting will later be referenced as the “mono” condition during the user study. Another possibility to achieve monoscopic 3D is to have a screen-space shader that blanks out one eye. However, we found, similarly to [20], that it creates an unpleasant feeling.

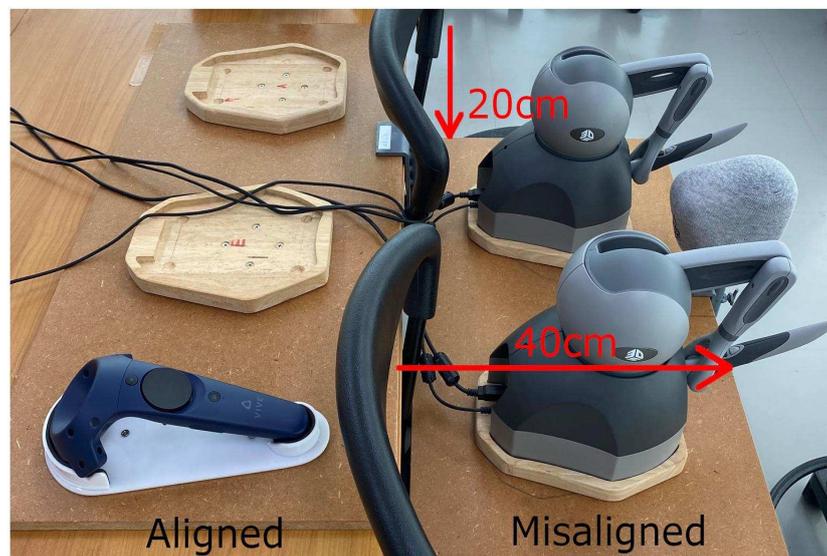
### 3.2 Hand-tool alignment

The force feedback devices are registered with the HTC Vive VR system by using a VR controller dock that is mounted on a board with a static offset to both haptic device bases (Fig 3 shows the misalignment condition). Inside the game engine, we define the virtual position of the haptic device origins of the mirror ( $p_M$ ) and the drill ( $p_D$ ) inside the scene, with the virtual distance set to the physical distance between them, 30 cm in our setup. When we run the simulator in a new VR configuration (new light house locations or new haptic device locations), a calibration procedure is manually initiated by a key-press. We calculate the virtual VR controller target origin  $p_{C_t}$  by applying an offset to the mirror origin, that we previously defined. The offset



**Fig 2.** In-game view of the VR dental surgery simulator, in which a student is performing root canal access opening on tooth #36.

<https://doi.org/10.1371/journal.pone.0291389.g002>



**Fig 3.** The calibration of the haptic devices with the HTC Vive VR system is implemented using a VR controller with a static offset. The “hand-tool misalignment” is achieved by calibrating and then moving the haptic devices forward and downward in front of the table, as shown here. Republished from [27] under a CC BY license, with permission from IEEE, original copyright 2022.

<https://doi.org/10.1371/journal.pone.0291389.g003>

needs to be measured and tuned by hand, in our setup, the offset is a translation of  $\Delta p = (22\text{cm}, 26\text{cm}, -7\text{cm})$  and a rotation of  $\Delta\theta = (0^\circ, 0^\circ, 90^\circ)$ . We then define the target VR controller origin as

$$p_{C_T} = p_M + \Delta p \quad (1)$$

$$\theta_{C_T} = \theta_M + \Delta\theta \quad (2)$$

where the rotation angles are simply added together. Now given the virtual target VR controller location  $p_{C_T}$  and actual physical VR controller location  $p_C$ , we calculate the difference and add it the VR camera location  $p_{VR}$ :

$$p'_{VR} = p_{C_T} - p_C \quad (3)$$

$$\theta'_{VR} = \text{Delta}(\theta_{C_T}, \theta_C) \quad (4)$$

where  $\text{Delta}(A,B)$  calculates the difference by subtraction  $A - B$  followed by normalization to the range of  $[-180, 180]$ . By doing this, we align the virtual tools and haptic device handles, within the accuracy of the VR tracking. We call this condition “hand-tool alignment” (as shown in Fig 1).

To define the contrasting condition, “hand-tool misalignment”, we do the same calibration, but additionally offset the real haptic devices. We moved the haptic devices down by 20 cm and forward by 50 cm, relative to the table (see Fig 3). We chose this offset to simulate a misalignment setting that resembles the offset on a desktop monitor in VR.

### 3.3 Visual perception

Dentists make heavy use of their eyes, during dental surgery, such as to check bur depth and bur orientation, as well as in pauses that occur between drilling a tooth, to precisely inspect their own progress. Modern HMDs allow for easy tracking of gaze behavior, with appropriate sensors already built-in, which is the case for the HTC Vive Pro Eye. This made eye tracking easy to implement into our simulator, however the accuracy was a challenge. Since the objective of this study required only determining the the eye-tooth distance, we used a simple form of eye tracking to determine at which point in time the user is looking at the tooth and where his eyes are located. Based on the “cyclops eye” (it is the mid-way point between the left and right eye, here in world coordinates) and tooth position, we can determine if the gaze ray hits the tooth, and in those instances, we log the current eye position and tooth hit position. Using this data, we determined the mean eye-tooth distance over an entire trial and regarded each trial as a separate sample point.

The human eye can naturally see much more detail than the HTC Vive Pro Eye can display with its limited resolution. This is very apparent when looking at small objects in VR, such as a human tooth and its individual features such as the root canal orifices. Since the accommodation range puts a lower bound on the distance of our eyes to the tooth, the screen resolution of the tooth is highly limited. If one looks at Fig 2, one can see that at a viewing distance of around 23 cm, the tooth takes up a miniscule amount of the screen. We estimate the area to be around  $119 \times 119$  pixels, taking up only 0.31% of the already limited HMD screen resolution. For healthy people between 20 and 25 years old, the accommodation near point and convergence near point are 9.92 cm and 7.18 cm [28], which set a physical limit for how closely objects can be focused. However, in case of stereo vision, we suspect that this lower bound is much higher inside a VR HMD, such as the HTC Vive Pro Eye. From our subjective tests, the

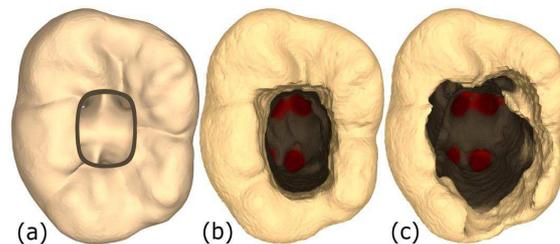
near point that can be focused on is in the range of 20 to 25 cm. One possible explanation for this could be that the HMD's limited field of view increases the stereo disparity and makes interocular correlation especially difficult, which limits the range in which binocular vision works effectively [29]. For the monoscopic 3D condition, the stereo disparity is always 0, no matter how close or far objects are. Therefore, there is no lower bound for the distance that objects can be focused on in monoscopic 3D, so participants of this condition can move as close as they desire to the tooth, unlike participants within the stereoscopic 3D condition. Based on the lower focus bound in the stereo 3D condition, we expect the eye-tooth distance to be lower for stereo 3D, with an average distance around 20 to 25 cm.

### 3.4 Automated outcome scoring

Dental outcomes are usually scored by an expert in dentistry. This score might appear subjective, however they follow a close set of objective measures, which makes it a robust scoring system that is mostly objective. For example, when we let two independent expert dentists score our data set, the experts' scores had excellent reliability ( $\kappa = 0.87$ , and intra-class correlation (ICC) of 0.98). In the dental scoring system, each of the four cardinal tooth walls and the pulp floor is visually observed and rated for errors by the expert. The criteria for rating the errors can be summed up in the following way:

- +0 Access to all orifices without an excess cavity.
- +1 Access to all orifices with minor over-drilling.
- +2 Incomplete removal of pulp chamber roof and/or excessive over-drilling.
- +3 Unidentified canals and/ or perforation.

The overall error rating for a tooth is taken to be the sum of the error ratings of the walls and pulp floor. Therefore, the error ranges from 0 to 15, with lower values indicating better performance (examples shown in Fig 4). Based on the excellent conformity of the two experts, we used the mean error value in our analysis. It would augment the simulator to implement an automated scoring system based on the outcomes achieved inside the simulator. Our automated score should highly correlate with the experts rating of those virtual outcomes. However, as our user study is comprised of 40 participants, each running 6 trials, we have 240 total outcomes. It was not feasible to ask the experts to evaluate each one of the 240 virtual outcomes, as it is too much data. Therefore, we needed to compress the data set to essential



**Fig 4. Different conditions of tooth #36.** (a) Ideal root-canal access opening. (b) A root canal access opening with a low error score. All orifices are accessible with little over-drilling. (c) A root canal access opening with a high error score, as multiple walls are over-drilled and not smooth. Republished from [27] under a CC BY license, with permission from IEEE, original copyright 2022.

<https://doi.org/10.1371/journal.pone.0291389.g004>

outcomes that sample the complete value range that all of the 240 outcomes encompass as uniformly as possible. We proceeded with the implementation in four steps:

- i). Generate an ideal drilling outcome. We generated the tooth in Fig 4 (left) by consulting an expert dentist, to verify that there is no under-drilling or over-drilling present, and that all walls and floor are well-shaped and have smooth edges. All four orifices are visible from the access opening (though not necessarily from the same angle).
- ii). Select from existing binary classification metrics one that generates normally distributed scores for the total outcome range. Additionally, we manually checked random samples visually to check if they are sensible based on the previously shown expert scoring system, evaluated by a dental novice. Here we looked at 24 of the state-of-the-art metrics (many of which are presented in [30]) and selected the F1-score [31] to be most ideal for further processing.
- iii). Select 20 samples that uniformly cover a wide range of the total value range with the previously chosen metric (F1-score) and let experts score these outcomes, without knowledge of the F1-score. The experts received each sample as a 3D mesh, which they could rotate and inspect on their personal computer. Again, the experts had excellent reliability ( $\kappa = 0.89$ , and ICC = 0.998).
- iiii). Implement a new metric and fine-tune it such that correlates well with the expert scores.

Through exploration we found that the F1-score, which is the harmonic mean of Sensitivity and Precision, can be improved upon. We developed a new metric we call *Dentist* (abbreviated by *D*), which combines the two scores of Sensitivity *S* and Precision *P*

$$P = \frac{TP}{TP + FP}, \quad S = \frac{TP}{TP + FN}$$

we adjust the value range by linear interpolation

$$\tilde{P} = \frac{P - 0.95}{1 - 0.95}, \quad \tilde{S} = \frac{S - 0.2}{1 - 0.2}$$

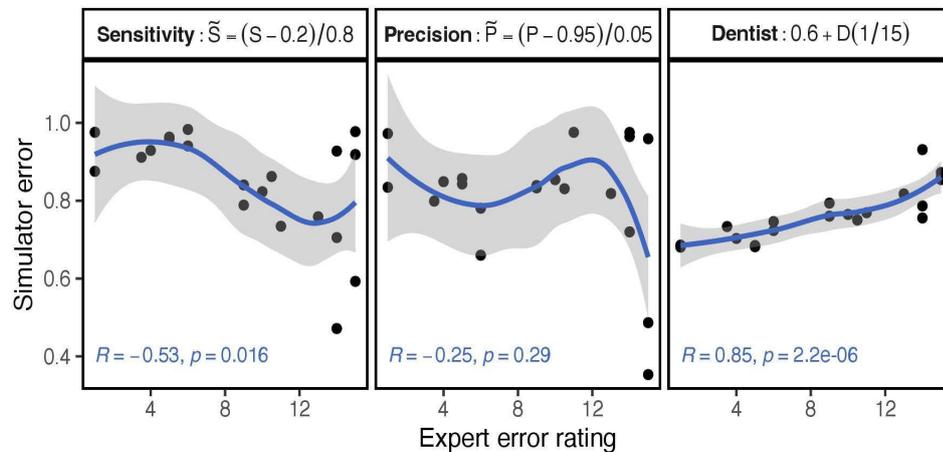
given those, we define Dentist *D* as

$$D = \left(1 - \frac{1.5\tilde{S} + \tilde{P}}{2.5}\right)15 \quad (5)$$

$$= \frac{15(32 \cdot FP \cdot TP + 3 \cdot FN \cdot TP + 35 \cdot FN \cdot FP)}{4(TP + FN)(TP + FP)} \quad (6)$$

It is essentially a weighted mean of  $\tilde{S}$  and  $\tilde{P}$ , though the values are flipped to represent a distance rather than a similarity, as well as multiplied by 15 to match the dentists' rating system. The value ranges of *S* and *P* are adjusted, because  $P \in (0.96, 0.995]$  whereas  $S \in (0.2, 1)$ . Therefore we adjusted both components to occupy roughly the same value range, the full range [0, 1].

We found *S* correlates well with penalization of over-drilling, whereas *P* correlates well with penalization of under-drilling. The two metrics complement each other well, as can visually be inspected in the individual metric correlation plot in Fig 5. When looking at the individual correlations, the highs and lows of both functions balance out to be nearly straight in our weighted sum. Interestingly, most metrics exhibit a similar shape to that of the Sensitivity



**Fig 5. Correlation of our metric and the basic metrics.** The line is a polynomial regression to illustrate the different curvatures. There is a high correlation of our scoring metric with the ratings of two independent experts'. In this graph, we show the metric in the value range of [0, 1], with 1 as ideal, to compare with other similarity metrics. In general however, the scoring metric is chosen to be in the range of [0, 15], with 0 as the ideal outcome, same as the dentist' rating system.

<https://doi.org/10.1371/journal.pone.0291389.g005>

correlation, so most would probably penalize over-drilling more. At runtime, we extract 3D voxels from the inner spheres volume by defining an implicit surface and discretizing it on a  $90 \times 135 \times 90$  grid. The same data is also used to generate the triangles, normals and colors to represent the rendered geometry at runtime. The extraction needed spatio-temporal optimization to run at interactive rates. Based on these voxel values, we can compute the standard binary voxel classification sums:

- *TP* (True Positive): Correctly undrilled voxels.
- *TN* (True Negative): Correctly drilled voxels.
- *FP* (False Positive): Incorrectly undrilled voxels.
- *FN* (False Negative): Incorrectly drilled voxels.

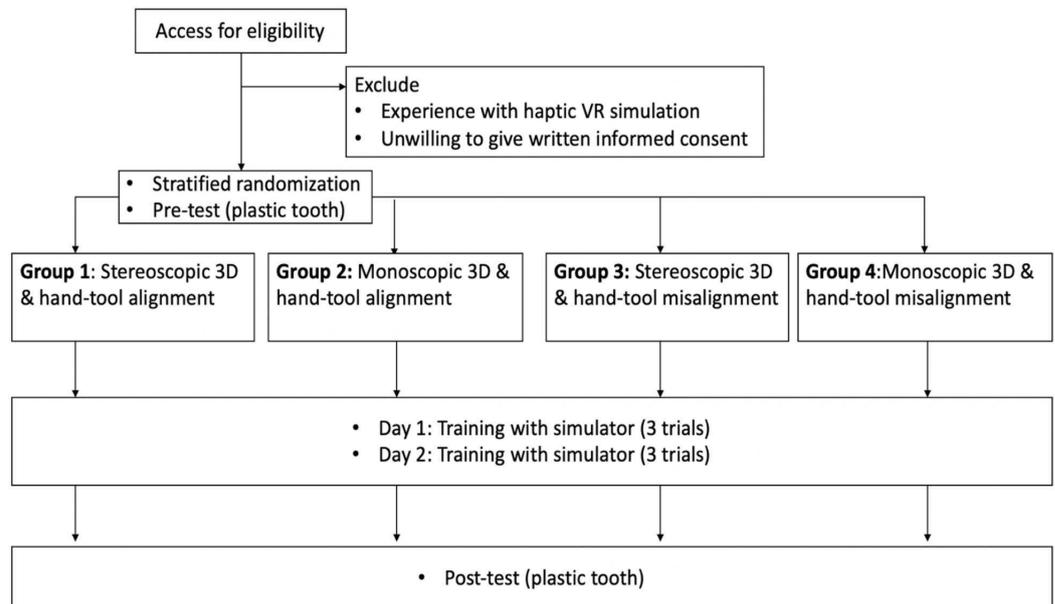
As we can see from their respective definitions, *FN*, which is the penalty count for over-drilling, is only found in *S*, which explains why it penalizes over-drilling. *P* penalizes under-drilling more as it only incorporates *FP* as the measure for error. The correlation of *D* with the expert rating is of high degree with  $R = 0.85, p < 0.0001$  (see Fig 5). Of the 24 common classification metrics that we evaluated, the best reach a correlation of approx.  $-0.65$ . Another existing scoring method achieves an information-based measure of disagreement (IBMD) [32] of  $0.04-0.21$  [11]. Besides an ideal drilling outcome, an expert has to additionally contract and expand the drilling region to create a min and max region that is used as weights in the non-linear scoring function. Contrary to that, our method only requires a single ideal drilling outcome to compare against and achieves a similarly low IBMD of  $0.09$  (for the 20 essential outcomes measured against the expert ground truth). We did not use the IBMD at the scoring design stage as it measures absolute error, but for our user study, we find relative correctness to be more important. In the following the training score as well as the training gain will be calculated based on the Dentist metric. A future improvement could be to acquire more rated samples and use supervised learning to better approximate the experts' rating system.

#### 4 User study

After receiving ethical approval from the Institutional Review Board from Mahidol and Thammasat universities, we invited students enrolled in the Faculty of Dentistry of Thammasat University to participate in our study. We recruited 40 participants (12 male, 28 female) and conducted a randomized controlled study. All participants were fifth year dental students, between 20 and 24 years of age and gave verbal consent to record anonymized data. They were not admitted to the study if any of the following criteria were present: (i) had received prior experience with the simulation, or (ii) received below 70% marks in knowledge assessment of endodontic cavity preparation, as this indicates insufficient theoretical knowledge to start practicing motor skill. The participants were randomly assigned to one of the four groups:

- Group 1: Stereoscopic 3D & hand-tool alignment
- Group 2: Monoscopic 3D & hand-tool alignment
- Group 3: Stereoscopic 3D & hand-tool misalignment
- Group 4: Monoscopic 3D & hand-tool misalignment

The task for the participants was to perform access opening on the virtual tooth during the training session and on a plastic tooth (lower left molar; tooth number 36; <http://www.nissin-dental.net/>) in pre- and post-training assessment sessions. A student's ability to perform the root canal access opening on such plastic teeth will predict with high reliability their ability to perform the task on real human teeth. Participants were briefly instructed on the use of the simulator, the experiment flow and the requirements of the access opening. As shown in the study flowchart (Fig 6), the training of each participant took place on two separate days. The



**Fig 6. Flowchart that shows the user study procedure.** Republished from [27] under a CC BY license, with permission from IEEE, original copyright 2022.

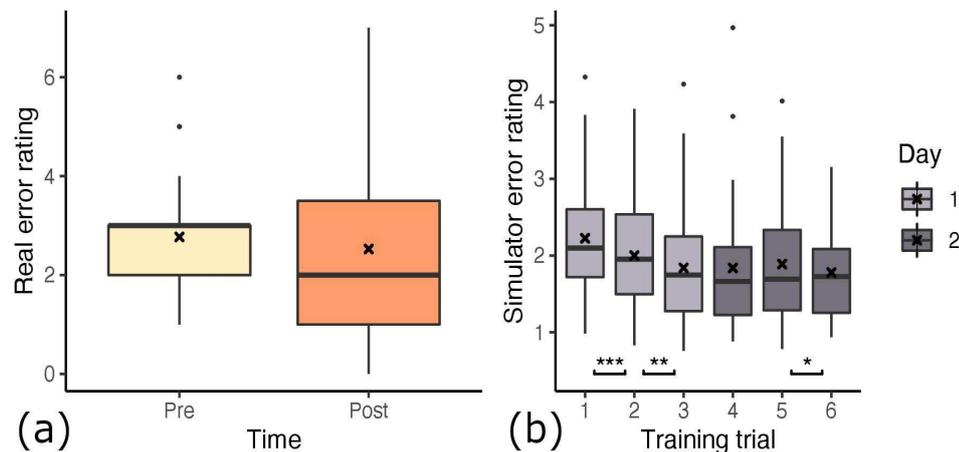
<https://doi.org/10.1371/journal.pone.0291389.g006>

first day consisted of briefing, pre-test, and the first training session consisting of three trials using the simulator. The time for each trial inside the simulator on the first day was an average of 7.71 min (ranging from 1.11 to 26.53 min). After each trial, students could inspect their drilling result in detail on a separate computer screen (see Fig 4), which is not included in the above times. The second training session of three trials with the simulator, along with the follow-up post-test and answering two questionnaires, took place afterwards on day 2. Here, the trials took an average of 5.18 min (ranging from 1.79 to 15.51 min), which is significantly faster than on the first day ( $t(178.15) = 3.8, p < 0.001$ ). There was a gap of four to seven days between days 1 and 2 of training. The pre- and post-test plastic teeth were independently scored by two experts. As we mentioned in section 3.4, the individual scores had overwhelming conformity. Therefore, we used the mean value of the two experts' scores in the following analysis.

## 5 Results

The error scores for the pre-test range from 1 to 6.5, whereas the post-test scores range from 0 to 7 (see Fig 7). We define the error change  $e_{\Delta}$  for each student as the difference between pre-test error score,  $e_0$ , and post-test error score,  $e_1$ , so  $e_{\Delta} = e_1 - e_0$ . With this,  $e_{\Delta}$  defines the inverse learning gain for each student. The learning gain is normally distributed around the mean  $M = -0.375$  with a median of  $-0.5$  and standard deviation  $SD = 1.87$ . The value range is  $-5$  to  $4$ . We determined 3 outliers based on inter-quartile range analysis, resulting in removing the following learning gains:  $\{-5, -5, +4\}$ . These outliers are very unusually high and low learning gains which we feel do not represent an effect of the participant group but rather an inherent property of the participant. After removing outliers, the distribution is centered around the slightly larger  $M = -0.24$  with the same median of  $-0.5$  and  $SD = 1.43$ .

Looking at the pre- and post-error, we observe an overall decrease of students' error score from pre- ( $M = 2.77, SD = 1.19$ ) to post-training ( $M = 2.53, SD = 1.56$ ) of the root canal access opening inside our simulator. A paired one-tailed t-test shows a mean difference of  $-0.2432$ , with significance of  $p = 0.153$  ( $t(36) = 1.037$ ). Based on the  $p$ -value, we can not determine whether the students' overall improvement in performance is caused by the training. On the



**Fig 7. Differences in paired error ratings with respect to time.** Gathered over all participants, regardless of condition groups. There was a 4-7 day wash out period between day 1 and 2. (a) Error determined by expert dentists on real outcome. (b) Error determined by algorithm on simulator outcome. Consecutive trials are compared for differences in means (\*:  $p < 0.05$ , \*\*:  $p < 0.01$ , \*\*\*:  $p < 0.001$ ).

<https://doi.org/10.1371/journal.pone.0291389.g007>

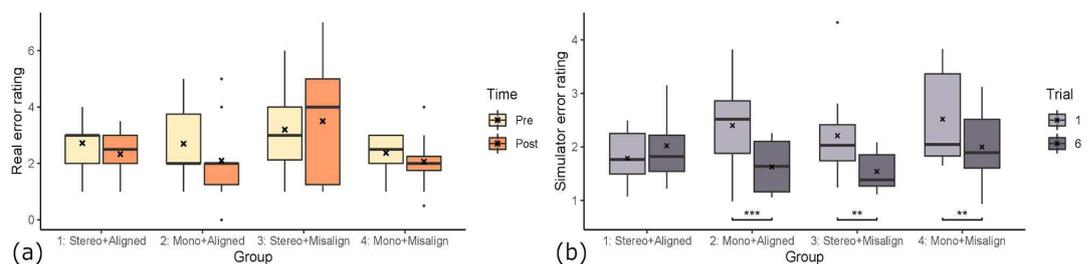
other hand, the participants' scores, measured inside the simulator (see Section 3.4 for scoring details), improved on average. However, here the score was significantly better at trial 6 compared to trial 1 ( $t(36) = 14.7, p < 0.0001$ ). In fact, when looking at each trial score individually (see Fig 7b), we see a significant improvement from trial 1 to 2 of  $-0.22$  ( $t(36) = -3.38, p < 0.001$ ), from trial 2 to 3 of  $-0.16$  ( $t(36) = -2.85, p < 0.01$ ) and from trial 5 to 6 of  $-0.11$  ( $t(36) = -1.77, p < 0.05$ ). From trial #3 to #4 and from #4 to #5, we observed no improvement in the simulator outcome scores.

### 5.1 Groups

Between the four groups (as detailed in 4) we found differences in how well participants learned the task of root canal access opening. To determine the learning effect we compare each participants' pre-test error score to their post-test error score. The statistical significance is determined here by a paired one-tailed t-test with the hypothesis that the post-test error scores are lower than the paired pre-test error scores. As the learning gain is normally distributed, we used the parametric t-test. The distribution of pre- and post-test error rating per group are visualized in Fig 8a. The significant tests showed that none of the learning effects of the four groups are statistically significant.

We found that participants of group 1 "stereo & aligned" performed slightly better at the post-test ( $M = 2.33, SD = 0.90$ ) compared to the pre-test ( $M = 2.72, SD = 0.97$ ) with a mean difference of  $-0.389$  ( $t(8) = 0.902, p = 0.197$ ). Participants of group 2 "mono & aligned" improved their drilling performance between pre- ( $M = 2.7, SD = 1.25$ ) and post-test ( $M = 2.1, SD = 1.45$ ). The difference in error score of  $-0.6$  is substantial ( $t(9) = 1.327, p = 0.109$ ). Participants of group 3 "stereo & misaligned" on average scored worse in the post-test ( $M = 3.5, SD = 2.12$ ) compared to the pre-test ( $M = 3.2, SD = 1.57$ ) with a mean difference of  $0.3$  ( $t(9) = 0.586, p = 0.714$ ). The scores of participants in group 4 "mono & misalignment" improved in the post-test ( $M = 2.06, SD = 1.08$ ) compared to the pre-test ( $M = 2.38, SD = 0.744$ ). This is an improvement of  $-0.313$  in the error score ( $t(7) = 0.637, p = 0.272$ ). A one-way ANOVA showed no statistically significant differences between the mean learning gains of the groups ( $F(1, 35) = 0.335, p = 0.555$ ).

Interestingly, the simulator score changes showed different results (see Fig 8b). Here, all groups except group 1 increased their simulator scores significantly between the first (#1) and last trial (#6). The group with stereoscopic 3D and hand-tool alignment (group 1) did not improve or worsen their score significantly, going from  $M = 1.79, SD = 0.52$  to  $M = 2.20, SD = 0.7$  ( $t(8) = -2.15, p = 0.968$ ). When doing a non-paired t-test between the simulator score



**Fig 8. Group influence on learning gain.** The improvement of the groups after 6 training trials inside the simulator. (a) Error determined by expert dentists on real outcome. Groups 1 & 3 improved less than 2 & 4. (b) Error determined by algorithm on simulator outcome. Groups 2,3 & 4 improved significantly (\*\*:  $p < 0.01$ , \*\*\*:  $p < 0.001$ ).

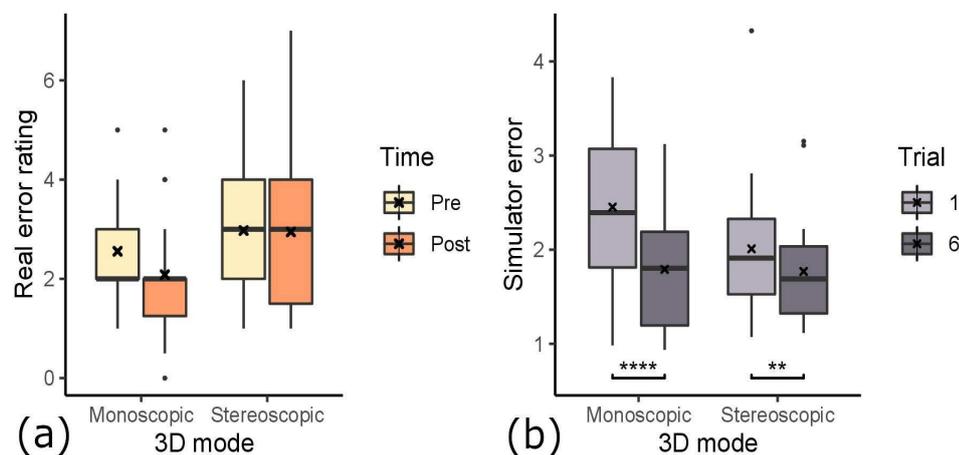
<https://doi.org/10.1371/journal.pone.0291389.g008>

of group 1 on trial #1 and any groups simulator score at trial #6, there is no significant difference. The group with monoscopic 3D and hand-tool alignment (group 2) improved significantly from  $M = 2.4$ ,  $SD = 0.84$  to  $M = 1.63$ ,  $SD = 0.49$  ( $t(9) = 4.78$ ,  $p < 0.001$ ). The group with stereoscopic 3D and hand-tool misalignment (group 3) improved significantly from  $M = 2.21$ ,  $SD = 0.88$  to  $M = 1.54$ ,  $SD = 0.36$  ( $t(9) = 3.44$ ,  $p < 0.01$ ). The group with monoscopic 3D and hand-tool misalignment (group 4) improved significantly from  $M = 2.52$ ,  $SD = 0.91$  to  $M = 2.0$ ,  $SD = 0.73$  ( $t(7) = 4.05$ ,  $p < 0.01$ ). The fact that group 1 is the only group that did not improve their simulator score could indicate that the group 1 setting (stereoscopic 3D and hand-tool alignment) is the easiest to learn, as their trial #6 simulator scores do not significantly differ from the other groups' score. We explore this thought more in section 5.4.

When doing a one-way ANOVA of the learning gains, measured with simulator error rating, there is no significant difference between the groups ( $F(1, 35) = 2.42$ ,  $p = 0.129$ ). Therefore, there are no statistically significant differences between the mean learning gains when measured inside the simulator.

### 5.2 3D rendering modes

To examine the effect that stereoscopic rendering had on the participants' performance (see Fig 9), we regard the data of group 1 & 3 as one set of data ("stereo"), and 2 & 4 as the other set of data ("mono"). We thereby control for the alignment condition. The "stereo" group's pre-test error ratings ( $M = 2.97$ ,  $SD = 1.31$ ) decreased by 0.0263 for the post-test ( $M = 2.95$ ,  $SD = 1.72$ ). The one-tailed t-test showed that the increase is likely a result of random chance ( $t(18) = 0.078$ ,  $p = 0.4695$ ). Therefore the students in the "stereo" group did not improve because of the training. In contrast, the "mono" group's post-test error ratings ( $M = 2.08$ ,  $SD = 1.26$ ) improved compared to the pre-test error ratings ( $M = 2.56$ ,  $SD = 1.04$ ). This large difference of  $-0.472$  have a statistical significance of  $p = 0.082$  ( $t(17) = 1.45$ ). This means the students of the "mono" group did improve because of the training in VR. This suggests that students performed better after training in the "mono" condition, which is not the case for the "stereo" condition. To measure the effect of the 3D rendering mode on the learning



**Fig 9. The effect of 3D rendering mode on learning effect.** (a) Error determined by expert dentists on real outcome. (b) Error determined by algorithm on simulator outcome (\*\*:  $p < 0.01$ , \*\*\*\*:  $p < 0.0001$ ). For both assessment methods, the monoscopic rendering mode is associated with larger performance improvement.

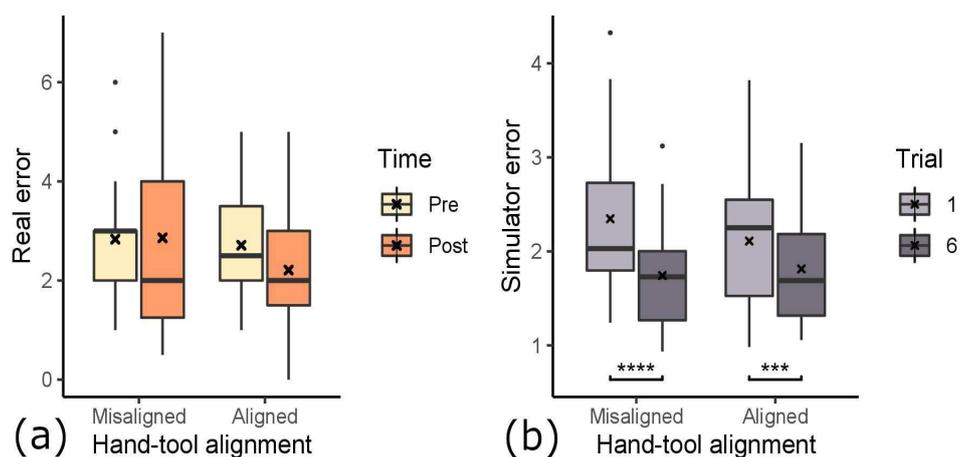
<https://doi.org/10.1371/journal.pone.0291389.g009>

effectiveness we compared the mean learning gains using a parametric two-tailed t-test. The differences of means of the learning gain between “mono” ( $M = -0.472$ ) and “stereo” ( $M = -0.026$ ) is 0.446, however the difference is not statistically significant ( $p = 0.348$ ).

We also looked at the influence of 3D rendering mode on the in-simulator learning gain (see Fig 9b). A t-test showed no statistically significant difference for the learning gain inside the simulator ( $t(34.965) = 1.174, p = 0.249$ ). The simulator error ratings for monoscopic and stereoscopic rendering modes were both improved. However, the simulator learning gain was larger for the monoscopic 3D condition ( $M = -0.66, SD = 0.90$ ), similarly to the influence on the real-world learning gain. The training gains in the stereoscopic condition were  $-0.24$  on average ( $SD = 0.99$ ). A t-test revealed that in both conditions, the simulator error ratings were statistically significantly lower after 6 trials compared to the first trial. In the monoscopic condition, the simulator error rating was reduced from 2.45 ( $SD = 0.85$ ) to 1.79 ( $SD = 0.62$ ) ( $t(17) = 7.95, p < 0.0001$ ). In the stereoscopic condition, the simulator error rating was reduced from 2.01 ( $SD = 0.74$ ) to 1.77 ( $SD = 0.59$ ) ( $t(18) = 2.70, p < 0.0001$ ).

### 5.3 Hand-tool alignment

To determine the impact of hand-tool alignment on participants' performance (see Fig 10), we regard the data of group 1 & 2 as one set of data (“aligned”), and 3 & 4 as the other set of data (“misaligned”), controlling for the stereo factor. The misalignment group did slightly worse on their post-test ( $M = 2.86, SD = 1.85$ ), compared to their pre-test ( $M = 2.83, SD = 1.31$ ). This small difference of 0.0278 was however shown by the t-test to be likely by random chance ( $t(17) = 0.078, p = 0.531$ ). Therefore the participants of the group “misalignment” did not improve by virtual training. However, the “alignment” group improved from their pre-test ( $M = 2.71, SD = 1.1$ ) by  $-0.5$  from their post-test ( $M = 2.21, SD = 1.19$ ). The t-test shows a statistical significance of  $p = 0.0598$  ( $t(18) = 1.635$ ). This suggests that the participants of the “alignment” group improved their error ratings because of the virtual drilling training. This shows, that virtual hand-tool alignment is important for effective training using a virtual simulator.



**Fig 10. The effect of hand-tool alignment on learning effect.** (a) Error determined by expert dentists on real outcome. The alignment of hands & tools shows better performance improvement. (b) Error determined by algorithm on simulator outcome (\*\*:  $p < 0.01$ , \*\*\*\*:  $p < 0.0001$ ), with no noticeable effect.

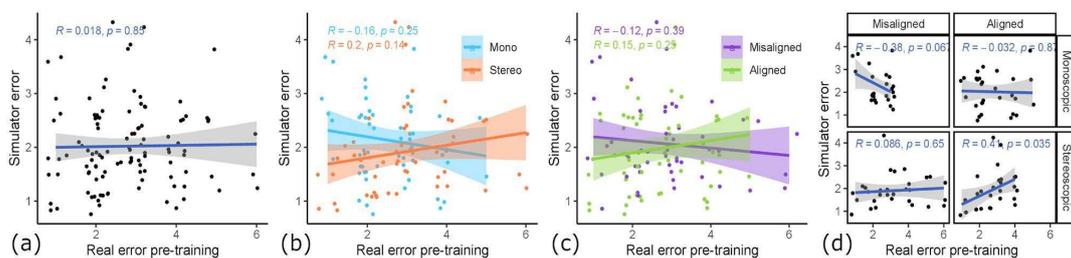
<https://doi.org/10.1371/journal.pone.0291389.g010>

We also examined the influence of hand-tool alignment on the in-simulator learning gain (see Fig 10b). In both conditions, participants improved their error ratings significantly. The aligned condition improved significantly from 2.11 ( $SD = 0.76$ ) to 1.81 ( $SD = 0.61$ ) ( $t(18) = 4.10, p < 0.001$ ). The misaligned condition improved slightly more with significantly lowering of the error from 2.35 ( $SD = 0.88$ ) to 1.74 ( $SD = 0.59$ ) ( $t(17) = 5.09, p < 0.0001$ ). We did not find any influence of the hand-tool alignment on the simulator learning gains ( $t(33.08) = -0.97, p = 0.17$ ). The participants in the aligned condition improved their simulator error rating by an average of  $-0.60$  ( $SD = 1.05$ ), and the misaligned condition improved on average by  $-0.30$  ( $SD = 0.87$ ).

#### 5.4 Suitability for assessment

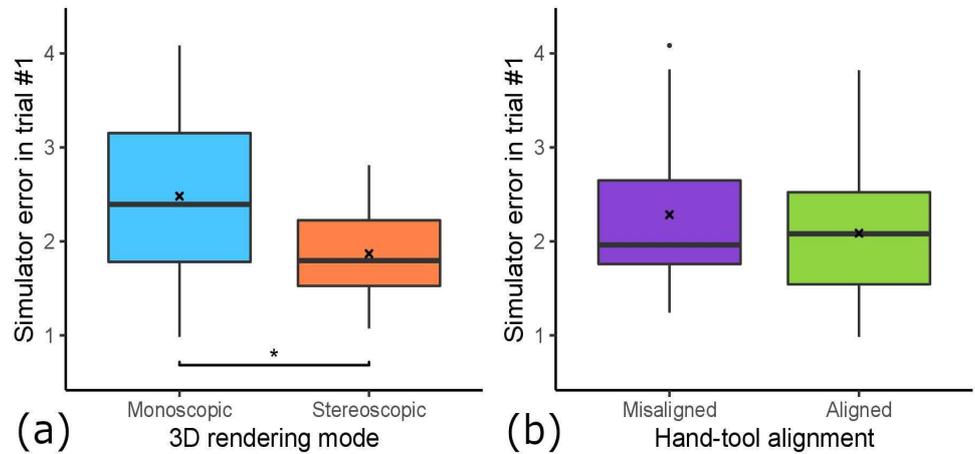
Suitability for assessment describes the transfer from previously acquired real psychomotor skills to the simulator. We quantify the suitability by the correlation of pre-training score on plastic teeth and in-simulator performance at the first training session.

When looking at Fig 11a, we can see that the correlation in all samples is very low and insignificant ( $R = 0.018, p = 0.85$ ). When looking at the factor hand-tool alignment (see Fig 11c), we can see that the aligned condition produces a better skill transfer from pre-training to simulator ( $R = 0.15, p = 0.25$ ) compared to the misaligned condition, which is even negative ( $R = -0.12, p = 0.39$ ). When looking at the mean differences in initial simulator performance (see Fig 12b), there is a small difference. The aligned condition resulted in a slightly lower initial simulator error ( $M = 2.09, SD = 0.74$ ) compared to the misaligned condition ( $M = 2.39, SD = 0.95$ ) ( $t(35.98) = 1.107, p = 0.138$ ). Similarly, we see an influence of 3D rendering mode (see Fig 11b) as a factor on the skill transferability from pre-training error to simulator error. Here, stereoscopic 3D had a positive correlation ( $R = 0.2, p = 0.14$ ), whereas monoscopic 3D had a negative correlation ( $R = -0.16, p = 0.25$ ). Additionally, there is a statistically significant impact on mean initial simulator performance (see Fig 12a). The stereoscopic 3D condition resulted in significantly lower initial simulator error ( $M = 1.99, SD = 0.73$ ) compared to the monoscopic 3D condition ( $M = 2.48, SD = 0.92$ ) ( $t(36.09) = 1.86, p < 0.05$ ). When looking at the four groups (see Fig 11d) with the factors combined, we can see that group 1 (stereoscopic 3D & hand-tool alignment) shows by far the strongest skill transfer correlation, with a moderate, significant correlation ( $R = 0.41, p < 0.05$ ) between pre-training error and simulator error. The other groups either had a low correlation, and group 4 (monoscopic & misaligned) even had a moderate negative correlation ( $R = -0.39, p = 0.067$ ), meaning students with good real-world skill tended to perform worse in the simulator than those with bad real-world skill.



**Fig 11. Relationship between pre-training score and initial simulator score.** The correlation between error rating pre-training (as measured by expert dentists) and initial simulator performance (as measured by simulator error ratings on day 1). (a) All samples, no correlation. (b) Influence of 3D rendering mode. Stereo 3D shows a positive and mono 3D a negative correlation. (c) Influence of hand-tool alignment. Alignment shows a positive and misalignment a negative correlation. (d) Influence of condition groups. The combination of stereo 3D & aligned (group 1) shows a moderate positive correlation.

<https://doi.org/10.1371/journal.pone.0291389.g011>



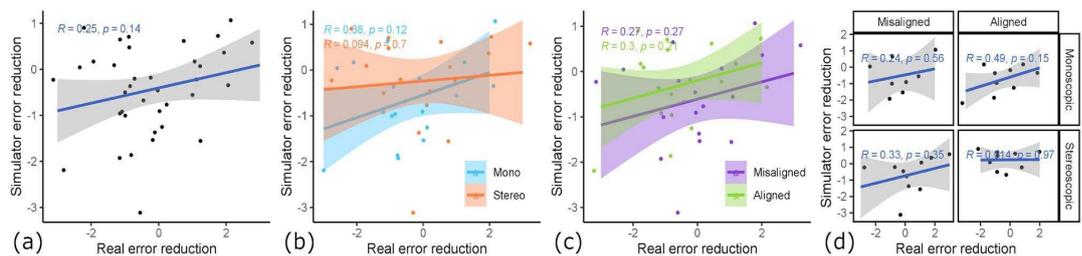
**Fig 12. The influence of both factors on initial simulator performance.** (a) Influence of 3D rendering mode. Stereo 3D shows significantly lower initial simulator performance than mono 3D (\*:  $p < 0.05$ ). (b) Influence of hand-tool alignment. Alignment shows lower initial simulator performance than misalignment.

<https://doi.org/10.1371/journal.pone.0291389.g012>

### 5.5 Learning transfer

Learning transfer describes the transfer of psychomotor skills learned inside the simulator to real-world assessed skill. We analyze the factor influences on learning transfer by looking at the correlation of real learning gain to virtual learning gain. Real learning gain is measured by pre- and post-training skill assessment, as rated by expert dentists. Virtual learning gain is measured by looking at the automated error rating of trial #1 and trial #6.

When looking at Fig 13a, there is an overall moderate correlation between simulator gain and training gain ( $R = 0.25$ ,  $p = 0.14$ ). The hand-tool alignment factor had almost no influence on the learning transfer (see Fig 13c), where in the aligned condition, the correlation is similarly high ( $R = 0.3$ ,  $p = 0.21$ ) like in the misaligned condition ( $R = 0.27$ ,  $p = 0.27$ ). However, the 3D rendering mode had a noticeable impact on the learning transfer (see Fig 13b). The monoscopic condition showed a moderate correlation between simulator gain and learning gain ( $R = 0.38$ ,  $p = 0.12$ ), whereas the stereoscopic condition showed almost no correlation ( $R = 0.094$ ,  $p = 0.7$ ).



**Fig 13. Relationship between real and virtual learning gain.** The correlation between learning gain as measured by real outcomes (pre- to post-training) vs. learning gain measured by simulator outcomes (trial #1 to trial #6). (a) All samples, moderate correlation. (b) Influence of 3D rendering mode. Mono 3D shows a moderate correlation, while stereo 3D shows no correlation. (c) Hand-tool alignment has no influence. (d) Influence of condition groups. Group 2 (mono 3D & aligned) shows the highest correlation ( $R = 0.49$ ).

<https://doi.org/10.1371/journal.pone.0291389.g013>

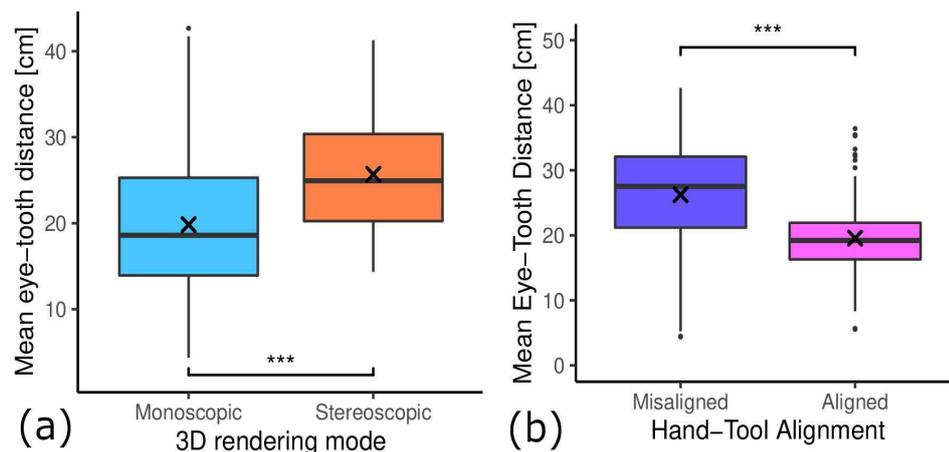
( $R = 0.094$ ,  $p = 0.7$ ). Furthermore, looking at the condition combinations (see Fig 13d), we can see that almost all groups showed a positive correlation, except for group 1 showing no correlation ( $R = 0.014$ ,  $p = 0.97$ ). Group 2 had the highest correlation ( $R = 0.49$ ,  $p = 0.15$ ), this indicates that in our setup, the conditions monoscopic 3D & hand-tool alignment create a learning environment that best translates the acquired skill to the real world. Please note that all correlations here are statically insignificant, since we only have 37 total data points which are even less when split up, however, the overall tendency for a positive correlation does not change in any subset of the data.

### 5.6 Eye-tooth distance

We compared the mean eye-tooth distance for participants in both 3D rendering conditions and found a large influence of the 3D rendering mode (see Fig 14a). The monoscopic condition had a significantly lower mean eye-tooth distance ( $M = 19.83$ ,  $SD = 8.19$ ) compared to the stereoscopic condition ( $M = 25.68$ ,  $SD = 6.82$ ) ( $t(207.52) = -5.71$ ,  $p < 0.001$ ).

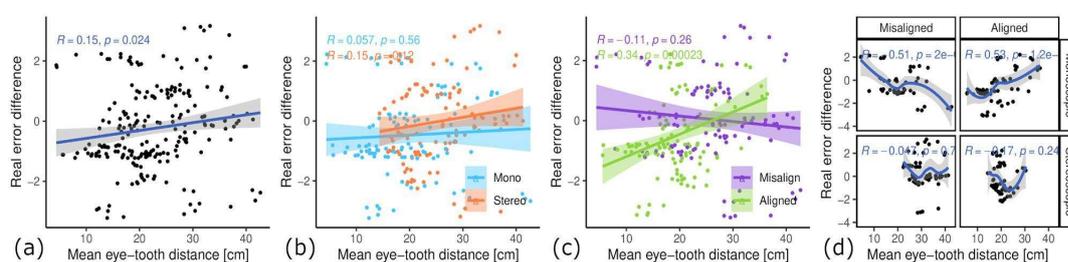
The hand-tool alignment had a similarly large effect on the mean eye-tooth distance. However, that is easily explained by the fact that we implemented misaligned hands and tools by calibrating with an offset that will result in the virtual tooth being further away from the participant. Therefore they had to move their head closer to the tooth to get the same eye-tooth distance, which some participants did not do. In the hand-tool aligned condition, the mean eye-tooth distance was significantly lower ( $M = 19.42$ ,  $SD = 5.99$ ) compared to the misaligned condition ( $M = 26.41$ ,  $SD = 8.47$ ) ( $t(183.9) = 6.97$ ,  $p < 0.001$ ).

As the learning gain influence of 3D rendering mode was counter-intuitive for us, we suspected that it could be a result of the accommodation near point limitation of stereoscopic rendering in combination with the limited resolution, which resulted in stereoscopic 3D condition having a tooth with effectively lower resolution. To analyze this, we correlated the average eye-tooth distance with the real-world learning gains (see Fig 15). Over all data points, there was a weak but statistically significant positive correlation between mean eye-tooth distance and learning gain ( $R = 0.15$ ,  $p < 0.05$ ). Further analysis showed that if we control for



**Fig 14. The influence of both factors on mean eye-tooth distance.** (a) Influence of 3D rendering mode. Mono 3D shows significantly lower eye-tooth distance than stereo 3D. (a) Influence of hand-tool alignment. Alignment shows significantly lower eye-tooth distance than misalignment. (\*\*\*:  $p < 0.001$ ).

<https://doi.org/10.1371/journal.pone.0291389.g014>



**Fig 15. Relationship of mean eye-tooth distance and learning gains.** (a) All samples show a weak correlation. (b) Influence of 3D rendering mode. Both correlations are weak. (c) Influence of hand-tool alignment. Alignment shows a moderate positive correlation. (d) Influence of condition groups. Group 2 (alignment & mono 3D) shows a strong positive correlation and group 4 (misalignment & stereo 3D) a strong negative correlation. The samples in the mono 3D have a global minimum and maximum at either extremes, whereas stereo 3D has a global minimum in the middle and the performance to the extremes gets worse. This suggests that there is an optimal distance for stereo 3D, a value after which the stereo vision suffers because of the large stereo disparity. For mono 3D (& aligned) the shorter the distance to the tooth, the better the learning performance.

<https://doi.org/10.1371/journal.pone.0291389.g015>

hand-tool alignment, data points in both conditions mono and stereoscopic 3D showed no or only weak correlation with mean eye-tooth distance (see Fig 15b). However, when controlling for 3D rendering mode, the data points in the aligned hands and tools showed a moderate positive correlation with strong statistical significance ( $R = 0.34, p < 0.001$ ) (see Fig 15c). The data points in the misaligned condition had a weak negative correlation between eye-tooth distance and learning gain ( $R = -0.11, p = 0.26$ ). If we look at the data points inside the aligned condition (see Fig 15d, second column) in the combination with monoscopic 3D (see Fig 15d, second column, first row), there is an even stronger correlation between mean eye-tooth distance and learning gain ( $R = 0.53, p < 0.0001$ ). Interestingly, the stereoscopic 3D & aligned group (see Fig 15d, second column, second row) shows a global maximum learning gain at around 23 cm mean eye-tooth distance.

## 6 Discussion

Learning transfer describes the extent to which skill acquisition translates from the acquisition modality to a target modality. In this study the acquisition modality is the simulator and the target modality is performance on realistic plastic teeth, as evaluated by dental experts. To assess the learning transferability of our simulator, we looked at (i) the absolute real learning gain and (ii) the correlation of learning gain and simulator learning gain. We hypothesized both experimental variables, 3D rendering mode and hand-tool alignment, to have a positive impact on the learning transferability of our VR simulator. Our results suggest that stereoscopic 3D had no statistically significant impact on the real-world learning gains (see Fig 9a). However, the mean learning gain was higher for the monoscopic 3D condition, which is the opposite of our hypothesis. We formulated this hypothesis based on our intuition of an additional depth cue increasing performance and the findings of McIntire et al. [16], which reported that 60% of user studies showed that stereoscopic 3D had a positive impact on performance. A more careful consideration of McIntire et al. in hindsight shows that they were focused on in-simulator performance, while we are concerned with learning gain. In fact, when looking at performance, purely measured by simulator error rating on trial #1 (see Fig 12a), stereoscopic 3D had a significant positive impact on user performance, consistent with the findings of McIntire et al. Additionally, McIntire's literature review spans a wide variety of tasks, whereas complex surgery on a small object (like in our user study) is a very uncommon task that puts special requirements on the display, especially resolution. However, we expected

the increased user performance to also translate to increased learning effectiveness. This was also not the case for the virtual learning gains (see Fig 9b). We further investigated the impact of stereoscopic 3D on learning effectiveness by correlating the virtual and real learning gains across all conditions, thereby controlling for the spread in virtual learning gains (see Fig 13). We found that the overall correlation was moderate, which suggests that participants that increased their simulator score also tended to increase their real-world score. Stereoscopic 3D had a negative impact on the correlation, compared to the monoscopic 3D condition. In fact, when comparing the learning correlations of group 1 (stereo & aligned) and group 2 (mono & aligned), group 2 has a strong correlation, whereas group 1 has none. This also suggests that skills learned inside the simulator in monoscopic 3D translate better to the real-world. Our use-case involves looking at a small object to make out fine details. Therefore we also recorded and analyzed eye tracking data (see Fig 14), which shows a significant impact of 3D rendering mode on mean eye-tooth distance, with users of monoscopic 3D having a significantly lower mean distance compared to stereoscopic 3D. Interestingly, we also noted a much lower standard deviation in the stereo 3D condition and a global minimum at around 15 cm that is larger than the expected near point at 9.92 cm [28], which suggests an optical lower bound in the stereoscopic 3D rendering. By correlating the mean eye-tooth distance per trial to the real-world learning gain, we found that in group 1 (stereo & aligned) the optimal learning gain is achieved in the middle of the distribution, at 23 cm distance, whereas this optimum is located at the extremes for other groups. By our estimation, the near-point in the simulator with stereoscopic 3D is located at the same distance of 23 cm. We suspect the user is trying to be as close to the tooth as possible to maximize the resolution of the tooth, while also being far enough away to be able to focus the tooth. In fact, group 2 (mono & aligned) showed a strong linear relationship between eye-tooth distance and real-world learning gain, which shows that the distance explains over 50% of the learning gains, as it results in a higher tooth resolution on screen. This intuitively makes sense, as there is no perceivable near-point in the monoscopic 3D condition and participants can essentially look as close to the tooth as they like. They thereby increase the tooth resolution on screen and receive more information, which could be regarded as immediate feedback of the drilling procedure, as they could see more details. Participants in the stereo condition did not have a chance to receive this form of immediate feedback. As it has been shown many times, timely feedback has a significant positive impact on learning effectiveness compared to delayed feedback [33–35]. Thus, our first hypothesis  $H_{V_{learn}}$  could not be confirmed. However, it is likely that eye-tooth distance is a confounding variable that explains the counter-intuitive influence of 3D rendering mode on learning gain. Future studies that incorporate small objects in VR should control their stereopsis to allow for a near-point that is realistic for the target task [28]. We suspect that when controlling for the tooth resolution in the described manner, stereoscopic 3D could have a positive impact on learning effectiveness of a VR simulator.

Hand-tool alignment had a positive impact on the learning effectiveness of the simulator, with higher real-world learning gains in the alignment condition (see Fig 10). This confirms our hypothesis  $H_{A_{learn}}$ . However, when correlating virtual learning gains and real-world learning gains (see Fig 13), we found no significant impact from hand-tool alignment, similar to the 3D rendering mode. Both conditions, aligned and misaligned, showed a moderate correlation between virtual and real learning gains, meaning both conditions translate the learned skills similarly to the real world. This indicates, that in the aligned condition, participants that did not improve substantially in the simulator scoring still tended to improve at the real task, which was not the case for the misaligned condition. In fact, we could see a slightly larger simulator learning gain for the misaligned condition (see Fig 10b). Based on these findings, we

showed that a simulator with hand-tool misalignment, such as when using a desktop monitor, is more likely to have weak learning transfer. Users of these kinds of simulators could be more likely to learn the intricacies of the simulator, not the real task.

Skill assessment is the process of determining a person's skill in a certain task or field, compromised of a set of tasks. Often, this skill is the foundation to determine if a person has also acquired expertise in this task or field. It is essential for the assessment tool to have accurate and reliable skill evaluation. To determine the feasibility of our simulator as a skill assessment tool, we looked at the correlation between the pre-training error, which is the ground truth of the student's current skill level, and the simulator error ratings on day 1. We hypothesized that both variables would positively impact the skill transfer from real-world to the VR simulator. Many studies find stereoscopic 3D to have a positive impact on performance in virtual surgical tasks [20, 21], which led us to hypothesize stereoscopic 3D would also have a positive impact on skill transfer. This follows the logic that higher virtual performance indicates intuitive usability of the simulator, which should better translate real-world experience to simulator experience. A simulator with intuitive usability would help identify the simulator's suitability as an automated and objective skill assessment tool, which is something the medical community is looking for [9, 10]. To analyze the intuitive usability, we mainly considered the trials on day 1 of the virtual training, as the data shows a learning curve that starts plateauing on day 2. When looking at the simulator scores on trial #1 (see Fig 12), we found a significant positive impact of stereoscopic 3D on the score of trial #1. We further correlated the pre-training real-world score with the simulator score on day 1 (see Fig 11). Here, we found that the 3D rendering mode had an impact on the correlation, with stereo 3D showing a moderate positive correlation, while mono 3D showed a moderate negative correlation. These findings confirm our third hypothesis  $H_{V_{assess}}$ , that stereoscopic 3D has a positive impact on skill transfer.

We expected hand-tool alignment to have a similar effect. However this is mostly based on our intuition, as we did not find studies that deal with this issue. The simulator error in trial #1 was only slightly lower in the aligned condition compared to the misaligned condition (see Fig 12). Interestingly, when correcting for the real-world skill by correlating pre-training error and simulator error on day 1, we found that hand-tool alignment had a positive impact. Although the impact is lower than the effect of the 3D rendering mode, it still shows that aligned hands and tools improve skill transfer as participants with low pre-training error tended to also have low simulator error on day 1 in this condition. Therefore, our data suggest that our last hypothesis  $H_{A_{assess}}$  is confirmed.

In fact, when looking at both variables together, the effect accumulates. Resulting in the skill correlation being the highest for the group 1 samples (stereo & aligned), with the initial simulator performance correlating over 40% with the expert pre-training assessment. This confirms that this setting is the most intuitive one, as it best translates users' already predominant preparation skill. We can even see the simulator being the basis of development for an automated, reliable and objective skill assessment tool in this setting.

The connection of our simulator and reality is very interesting to look at. Previous studies [20, 21] examined performance and learning differences in dental simulators with stereoscopic and monoscopic rendering. In those studies the task was carried out on simulated geometric objects. Evaluation of skill was done within the simulator, with automated scoring based on material removed. By contrast, our study used the endodontic task of root canal access opening. Evaluation of learning gains was done using pre- and post-testing on realistic plastic teeth, with scoring done by dental instructors using the standard method used in clinical teaching. Thus, it can be argued that our study is done in a more realistic setting and includes evaluation of transferability of learned skills. Transferability is important to evaluate since it is entirely

possible to attain a high level of skill in a simulator, yet not have this in-simulator skill translate to real-world tasks.

## 7 Conclusion

This is the first study to analyse the effect of different aspects of VR realism on transferability of dental skills from VR simulation training to real-world tasks and vice versa. We have found that the alignment of the physical and virtual tools had a positive impact on students' learning gains, compared to students with misaligned physical and virtual tools. Hand-tool alignment was also helpful in increasing simulator usability, suggesting it is easier to adapt to the simulator and is better suited for skill assessment.

Surprisingly, we observed that in our setting monoscopic 3D rendering provided students with more helpful training compared to stereoscopic 3D, as their learning gain was higher. Although it must be noted that our limited sample size did not yield statistical significance. However, this counter-intuitive finding might be confounded by the eye-tooth distance, which was found to be significantly lower for the monoscopic 3D condition. Therefore, future studies need to control for eye-tooth distance, for example by enforcing a similar lower bound in the monoscopic condition, since such a lower bound naturally exists for stereo vision. The stereo vision near point should also be controlled, as we found the near point inside a VR HMD to be larger than in the real world. One possible explanation for this is the limited field of view of HMDs restricting reference points common for both eyes at high inter-ocular disparity. However, despite the large near point, stereoscopic 3D had a significantly better skill transfer, as it showed a high correlation with participants pre-training score. This shows that it is easier for participants to manifest their real-world skill inside the simulator when using stereo 3D. Consequently, it is the desired rendering mode when using a simulator for skill assessment purposes.

## Author Contributions

**Conceptualization:** Maximilian Kaluschke, Myat Su Yin, Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Data curation:** Maximilian Kaluschke, Myat Su Yin.

**Formal analysis:** Maximilian Kaluschke.

**Funding acquisition:** Myat Su Yin, Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Investigation:** Maximilian Kaluschke.

**Methodology:** Maximilian Kaluschke, Myat Su Yin, Siriwan Suebnukarn, Gabriel Zachmann.

**Project administration:** Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Resources:** Myat Su Yin, Peter Haddawy, Gabriel Zachmann.

**Software:** Maximilian Kaluschke.

**Supervision:** Myat Su Yin, Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Validation:** Maximilian Kaluschke, Myat Su Yin, Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Visualization:** Maximilian Kaluschke.

**Writing – original draft:** Maximilian Kaluschke, Myat Su Yin, Peter Haddawy, Siriwan Suebnukarn, Gabriel Zachmann.

**Writing – review & editing:** Maximilian Kaluschke.

## References

1. Sendyk DI, Chrcanovic BR, Albrektsson T, Wennerberg A, Zindel Deboni MC. Does Surgical Experience Influence Implant Survival Rate? A Systematic Review and Meta-Analysis. *International Journal of Prosthodontics*. 2017; 30(4). <https://doi.org/10.11607/ijp.5211> PMID: 28697203
2. Mordechai F, Tali C, Jonathan M, Ori P, Yaron B, Ram S, et al. The effect of type of specialty (periodontology/oral surgery) on early implant failure: a retrospective “Big-Data” study from a nation-wide dental chain in Israel. *Clinical Oral Investigations*. 2022; 26(10):6159–6163. <https://doi.org/10.1007/s00784-022-04565-z> PMID: 35759088
3. Wang D, Li T, Zhang Y, Hou J. Survey on multisensory feedback virtual reality dental training systems. *European Journal of Dental Education*. 2016; 20(4):248–260. <https://doi.org/10.1111/eje.12173> PMID: 26547278
4. Li Y, Ye H, Ye F, Liu Y, Lv L, Zhang P, et al. The Current Situation and Future Prospects of Simulators in Dental Education. *Journal of Medical Internet Research*. 2021; 23(4):e23635. <https://doi.org/10.2196/23635> PMID: 33830059
5. Towers A, Field J, Stokes C, Maddock S, Martin N. A scoping review of the use and application of virtual reality in pre-clinical dental education. *British dental journal*. 2019; 226(5):358–366. <https://doi.org/10.1038/s41415-019-0041-0> PMID: 30850794
6. Murbay S, Chang JWW, Yeung S, Neelakantan P. Evaluation of the introduction of a dental virtual simulator on the performance of undergraduate dental students in the pre-clinical operative dentistry course. *European Journal of Dental Education*. 2020; 24(1):5–16. <https://doi.org/10.1111/eje.12453> PMID: 31278815
7. Vincent M, Joseph D, Amory C, Paoli N, Ambrosini P, Mortier É, et al. Contribution of Haptic Simulation to Analogic Training Environment in Restorative Dentistry. *Journal of Dental Education*. 2020; 84(3):367–376. <https://doi.org/10.21815/JDE.019.187> PMID: 32176342
8. Urbankova A, Engebretson SP. Computer-assisted dental simulation as a predictor of preclinical operative dentistry performance. *Journal of dental education*. 2011; 75(9):1249–1255. <https://doi.org/10.1002/j.0022-0337.2011.75.9.tb05169.x> PMID: 21890855
9. Uoshima K, Akiba N, Nagasawa M. Technical skill training and assessment in dental education. *Japanese Dental Science Review*. 2021; 57:160–163. <https://doi.org/10.1016/j.jdsr.2021.08.004> PMID: 34567290
10. Hermens F, Flin R, Ahmed I. Eye movements in surgery: A literature review. *Journal of Eye Movement Research*. 2013; 6(4). <https://doi.org/10.16910/jemr.6.4.4>
11. Yin MS, Haddawy P, Suebnukarn S, Rhienmora P. Automated outcome scoring in a virtual reality simulator for endodontic surgery. *Computer Methods and Programs in Biomedicine*. 2018; 153:53–59. <https://doi.org/10.1016/j.cmpb.2017.10.001> PMID: 29157461
12. Suebnukarn S, Hataidechadusadee R, Suwannasri N, Suprasert N, Rhienmora P, Haddawy P. Access cavity preparation training using haptic virtual reality and microcomputed tomography tooth models. *International endodontic journal*. 2011; 44(11):983–989. <https://doi.org/10.1111/j.1365-2591.2011.01899.x> PMID: 21623838
13. Eve EJ, Koo S, Alshihri AA, Cormier J, Kozhenikov M, Donoff RB, et al. Performance of dental students versus prosthodontics residents on a 3D immersive haptic simulator. *Journal of dental education*. 2014; 78(4):630–637. <https://doi.org/10.1002/j.0022-0337.2014.78.4.tb05715.x> PMID: 24706694
14. Wang D, Zhao S, Li T, Zhang Y, Wang X. Preliminary evaluation of a virtual reality dental simulation system on drilling operation. *Bio-medical materials and engineering*. 2015; 26(s1):S747–S756. <https://doi.org/10.3233/BME-151366> PMID: 26406071
15. Yin MS, Haddawy P, Suebnukarn S, Kulapichitr F, Rhienmora P, Jatuwat V, et al. Formative feedback generation in a VR-based dental surgical skill training simulator. *Journal of Biomedical Informatics*. 2021; 114:103659. <https://doi.org/10.1016/j.jbi.2020.103659>
16. McIntire JP, Havig PR, Geiselman EE. Stereoscopic 3D displays and human performance: A comprehensive review. *Displays*. 2014; 35(1):18–26. <https://doi.org/10.1016/j.displa.2013.10.004>
17. Mon-Williams M, Warm JP, Rushton S. Binocular vision in a virtual world: visual deficits following the wearing of a head-mounted display. *Ophthalmic and Physiological Optics*. 1993; 13(4):387–391. <https://doi.org/10.1111/j.1475-1313.1993.tb00496.x> PMID: 8278192
18. Pavaloiu IB, Sandu SA, Grigorescu SD, Dragoi G. Inexpensive dentistry training using virtual reality tools. In: *Proceedings of the 10th International Technology, Education and Development Conference*, Valencia, Spain; 2016. p. 7–9.

19. Buckthought A, Yoonessi A, Baker CL. Dynamic perspective cues enhance depth perception from motion parallax. *Journal of Vision*. 2017; 17(1):10–10. <https://doi.org/10.1167/17.1.10> PMID: [28114478](https://pubmed.ncbi.nlm.nih.gov/28114478/)
20. De Boer I, Wesselink P, Vervoorn J. Student performance and appreciation using 3D vs. 2D vision in a virtual learning environment. *European Journal of Dental Education*. 2016; 20(3):142–147. <https://doi.org/10.1111/eje.12152> PMID: [26072997](https://pubmed.ncbi.nlm.nih.gov/26072997/)
21. Al-Saud LM, Mushtaq F, Mirghani I, Balkhoyor A, Keeling A, Manogue M, et al. Drilling into the functional significance of stereopsis: the impact of stereoscopic information on surgical performance. *Ophthalmic and Physiological Optics*. 2017; 37(4):498–506. <https://doi.org/10.1111/opo.12393> PMID: [28656672](https://pubmed.ncbi.nlm.nih.gov/28656672/)
22. Collaço E, Kira E, Sallaberry LH, Queiroz AC, Machado MA, Crivello O Jr, et al. Immersion and haptic feedback impacts on dental anesthesia technical skills virtual reality training. *Journal of Dental Education*. 2021; 85(4):589–598. <https://doi.org/10.1002/jdd.12503> PMID: [33274441](https://pubmed.ncbi.nlm.nih.gov/33274441/)
23. Sutter C, Sülzenbrück S, Rieger M, Müsseler J. Limitations of distal effect anticipation when using tools. *New Ideas in Psychology*. 2013; 31(3):247–257. <https://doi.org/10.1016/j.newideapsych.2012.12.001>
24. Wilkie RM, Johnson RL, Culmer PR, Allen R, Mon-Williams M. Looking at the task in hand impairs motor learning. *Journal of neurophysiology*. 2012; 108(11):3043–3048. <https://doi.org/10.1152/jn.00440.2012> PMID: [22993255](https://pubmed.ncbi.nlm.nih.gov/22993255/)
25. Epic Games Inc. Digital Humans—Metahuman Creator; 2022. Available from: <https://www.unrealengine.com/en-US/digital-humans> [cited 2022-12-17].
26. Kaluschke M, Weller R, Hammer N, Pelliccia L, Lorenz M, Zachmann G. Realistic Haptic Feedback for Material Removal in Medical Simulations. In: 2020 IEEE Haptics Symposium (HAPTICS); 2020. p. 920–926.
27. Kaluschke M, Yin M S, Haddawy P, Suebnukarn S, Zachmann G The Impact of 3D Stereopsis and Hand-Tool Alignment on Effectiveness of a VR-based Simulator for Dental Training In: 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI); 2022, pp. 449–455.
28. Hashemi H, Pakbin M, Ali B, Yekta A, Ostadimoghaddam H, Asharlous A, et al. Near points of convergence and accommodation in a population of university students in Iran. *Journal of ophthalmic & vision research*. 2019; 14(3):306.
29. Kane D, Guan P, Banks MS. The Limits of Human Stereopsis in Space and Time. *Journal of Neuroscience*. 2014; 34(4):1397–1408. <https://doi.org/10.1523/JNEUROSCI.1652-13.2014> PMID: [24453329](https://pubmed.ncbi.nlm.nih.gov/24453329/)
30. Tharwat A. Classification assessment methods. *Applied Computing and Informatics*. 2020;.
31. Chinchor N. MUC-4 Evaluation Metrics. In: Proceedings of the 4th Conference on Message Understanding. MUC4'92. USA: Association for Computational Linguistics; 1992. p. 22–29. Available from: <https://doi.org/10.3115/1072064.1072067>.
32. Costa-Santos C, Antunes L, Souto A, Bernardes J. Assessment of disagreement: a new information-based approach. *Annals of epidemiology*. 2010; 20(7):555–561. <https://doi.org/10.1016/j.annepidem.2010.02.011> PMID: [20538199](https://pubmed.ncbi.nlm.nih.gov/20538199/)
33. Taxipulati S, Lu HD. The Influence of Feedback Content and Feedback Time on Multimedia Learning Achievement of College Students and Its Mechanism. *Frontiers in Psychology*. 2021; 12. <https://doi.org/10.3389/fpsyg.2021.706821> PMID: [34955945](https://pubmed.ncbi.nlm.nih.gov/34955945/)
34. Shute VJ. Focus on formative feedback. *Review of educational research*. 2008; 78(1):153–189. <https://doi.org/10.3102/0034654307313795>
35. Fyfe ER, Rittle-Johnson B. Feedback both helps and hinders learning: The causal role of prior knowledge. *Journal of Educational Psychology*. 2016; 108(1):82. <https://doi.org/10.1037/edu0000053>

---

REFLECTING ON EXCELLENCE: VR SIMULATION FOR LEARNING INDIRECT VISION IN COMPLEX BI-MANUAL TASKS

*Outline*

---

	<b>Abstract</b> . . . . .	273
1	<b>Introduction</b> . . . . .	273
2	<b>Related Work</b> . . . . .	274
	2.1 Psychomotor Training in Dentistry . . . . .	274
	2.2 VR Simulators . . . . .	274
	2.3 Eye Tracking in VR Simulators . . . . .	274
3	<b>Virtual Simulator</b> . . . . .	274
	3.1 Anatomical Tooth Model . . . . .	275
	3.2 Haptic Device Registration . . . . .	275
	3.3 Simulation of Optical Magnification in VR . . . . .	276
	3.4 Eye Tracking . . . . .	276
	3.5 Measurement of Gaze Behavior . . . . .	277
	3.6 Scoring . . . . .	277
4	<b>User Study Design</b> . . . . .	278
	4.1 Participants . . . . .	278
	4.2 Procedure . . . . .	278
5	<b>Results</b> . . . . .	279
	5.1 Learning Effect . . . . .	279
	5.2 Eye Tracking . . . . .	279
6	<b>Discussion</b> . . . . .	280
7	<b>Conclusion</b> . . . . .	281
	<b>References</b> . . . . .	281

---

*Bibliography Entry*

Kaluschke, M., Weller, R., Yin, M. S., Hosp, B. W., Kulapichitr, F., Haddawy, P., Suebnukarn, S., and Zachmann, G. (2024). **Reflecting on Excellence: VR Simulation for Learning Indirect Vision in Complex Bi-Manual Tasks**. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*

*Author's Contribution*

The author designed and planned the user-study, implemented the simulator, data logging and all other software, assisted in running the study, did the data analysis, wrote the manuscript and reviewed and edited the manuscript.

*Copyright*

Copyright © 2024 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

*This paper has been omitted due to copyright considerations.*

## Schriftliche Arbeiten

### Eigenständigkeitserklärung und Einverständniserklärung zur Überprüfung mit Plagiatssoftware

#### Studierenden-Angaben

Matrikelnr.	2475447
Nachname	Kaluschke
Vorname	Maximilian

Titel der Arbeit:

**Immersive Medical VR Training Simulators with Haptic Feedback**

#### A) Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Alle Teile meiner Arbeit, die wortwörtlich oder dem Sinn nach anderen Werken entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht. Gleiches gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet, dazu zählen auch KI-basierte Anwendungen oder Werkzeuge. Die Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht.

- Ich habe KI-basierte Anwendungen und/oder Werkzeuge genutzt und diese im Anhang "Nutzung KI basierte Anwendungen" dokumentiert.

#### B) Einverständniserklärung zur elektronischen Überprüfung der Arbeit auf Plagiate

Eingereichte Arbeiten können nach § 18 des Allgemeinen Teil der Bachelor- bzw. der Masterprüfungsordnungen der Universität Bremen mit qualifizierter Software auf Plagiatsvorwürfe untersucht werden.

Zum Zweck der Überprüfung auf Plagiate erfolgt das Hochladen auf den Server der von der Universität Bremen aktuell genutzten Plagiatssoftware.

- Ich bin damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum oben genannten Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.
- Ich bin nicht damit einverstanden, dass die von mir vorgelegte und verfasste Arbeit zum o.g. Zweck dauerhaft auf dem externen Server der aktuell von der Universität Bremen genutzten Plagiatssoftware, in einer institutionseigenen Bibliothek (Zugriff nur durch die Universität Bremen), gespeichert wird.

Das Einverständnis der dauerhaften Speicherung des Textes ist freiwillig.

Die Einwilligung kann jederzeit durch Erklärung gegenüber der Universität Bremen, mit Wirkung für die Zukunft, widerrufen werden.

Weitere Informationen zur Überprüfung von schriftlichen Arbeiten durch die Plagiatsoftware sind im Nutzungs- und Datenschutzkonzept enthalten. Diese finden Sie auf der Internetseite der Universität Bremen.

Mit meiner Unterschrift versichere ich, dass ich die obenstehenden Erklärungen gelesen und verstanden habe und bestätige die Richtigkeit der gemachten Angaben.

Unterschrift

