# Medical Visualization and Simulation for Customizable Surgical Guides

# Medical Visualization and Simulation for Customizable Surgical Guides

**Proefschrift**

ter verkrijging van de graad doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. K.Ch.AM Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 1 September 2015 om 13:00 uur

door

**Thomas Kroes**

Master of Science in Industrial Design Engineering
Technische Universiteit Delft, Nederland
geboren te Nijmegen, Nederland.

*To Ferry.*

# Contents

# 1

# Introduction

Visualization is a vital part of human interaction, which relies upon vision to transmit information between people. The earliest known example dates back to 30.000 BC, in the form of the cave paintings found in Chauvet Cave, featuring graphical depictions of animals[1]. Visualization has evolved since then, and computer technology has revolutionized the ways in which we visually communicate. In this thesis, we are focused on medical visualization, a field of computer visualization which has been active since the late 1980s. It gives medical professionals unprecedented tools to improve patient treatment by allowing them to interact with complex medical images. It's mainly used for education, diagnosis and computer-assisted surgery (Preim and Botha, 2013). We focus on orthopaedic surgery, and in particular on the development of novel medical visualization tools for the planning of joint replacement surgery using a customizable surgical guide.

## 1.1 JOINT REPLACEMENT SURGERY

Joint replacement surgery, or arthroplasty, is a surgical procedure in which damaged natural joints are replaced by an (endo)prosthesis (artificial joint), as shown in Figure 1.1. The goal of this procedure is to restore the joint's normal function and to reduce pain. As a result, the patient's quality of life improves drastically. Arthroplasty is predominantly applied to the hip, knee and to a lesser extent the shoulder joint. From here on, we focus on total knee arthroplasty (TKA). The most common indication for TKA is osteoarthritis, followed by rheumatoid or other inflammatory arthritis, osteonecrosis, and ankylosed knee (Cho et al., 2013). During TKA, the shape of the

[1] The decorated cave of Pont d'Arc, known as Grotte Chauvet-Pont d'Arc, Ardèche

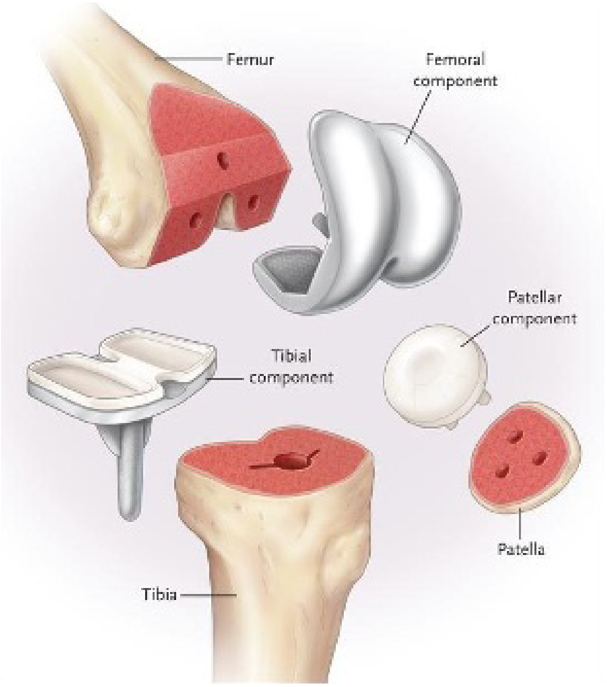


Figure 1.1: The shape of the distal femur, proximal tibia and patella need to be changed (e.g., sawing and drilling) in order to accommodate the prosthesis components (Leopold, 2009).

femur, tibia and patella have to be prepared, such that the prosthesis components can be installed correctly. It is important that all components of the prosthesis are properly aligned. Malalignment not only affects the function of the joint, but is also associated with overloading of the bone-prosthesis interface and subsequent failure by loosening (van Strien et al., 2009). In TKA, the femur and tibia components are aligned with the mechanical axis, as shown in Figure 1.2. The mechanical axis of the femur is a straight line that starts at the centre of the femoral head and ends in the middle of the intercondylar region. The mechanical axis of the tibia is also a straight line and starts from the centre of the tibial plateau and ends in the middle of the ankle. Because some of the reference points are occluded during surgery, guidance instruments are used to achieve alignment with the neutral mechanical axis.

### 1.1.1 COMPUTER-ASSISTED ORTHOPAEDIC SURGERY

Computer-assisted orthopaedic surgery (CAOS) refers to the integration of modern computer technology into the surgical pipeline of orthopaedic procedures, in an attempt to improve the surgical outcome and simplify surgical procedures. Figure 1.3 shows the main applications of CAOS, most relevant to our work are navigation and template based computer-assisted TKA. The image-based, and image-free navigation

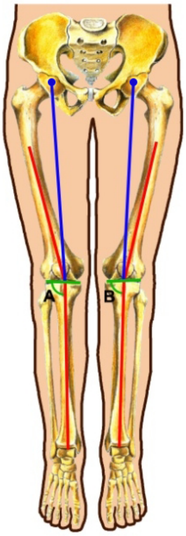
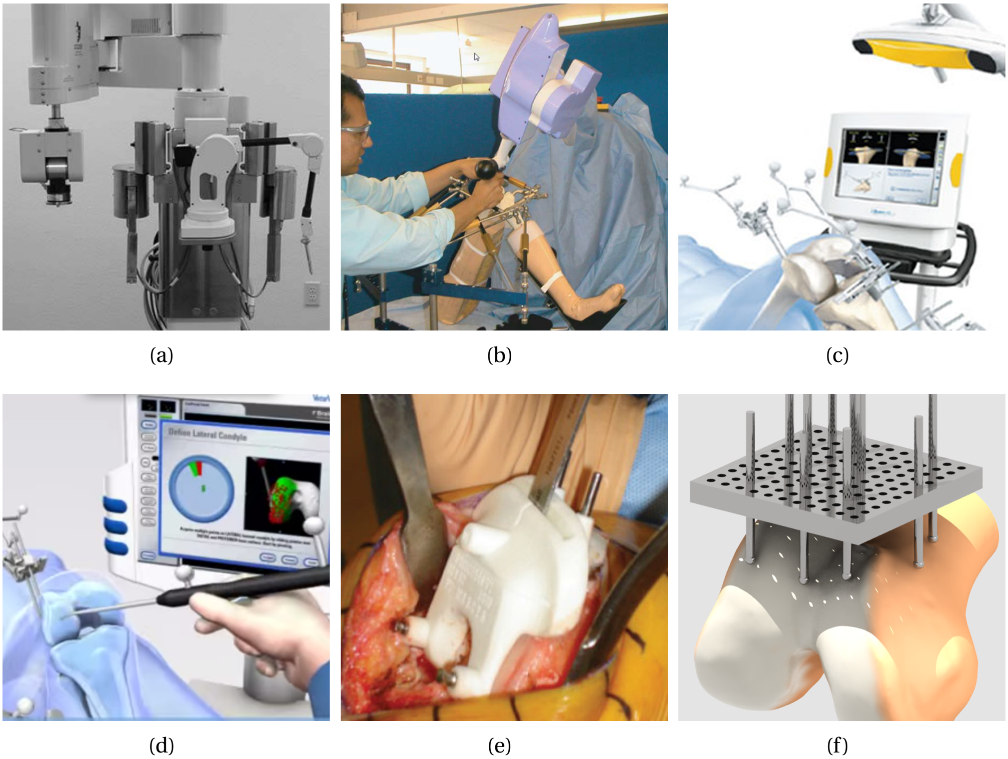


Figure 1.2: Aligning with the knee joint mechanical axis (blue line) is difficult because the center of the femoral head (blue dot) is obscured by several layers of tissue. For this reason, a femoral reference guide is inserted in the intramedullary canal, along the anatomical axis (red line) of the femur. A small rotation is then applied to align with the mechanical axis (Hosseinzadeh et al., 2013)

systems shown in Figure 1.3(c) and 1.3(d) are commercially available solutions for knee arthroplasty. Navigation is achieved with optical tracking, which means that auxilliary hardware such as camera systems have to be integrated in the operating theater. Computer navigated TKA is associated with improved prosthesis alignment, however, its use also interferes with standard surgical work-flow e.g., marker trees need to be attached to the patient and the surgeon must ensure that marker trees are not disrupted or temporarily occluded. Furthermore, the increased accuracy does not necessarily lead to improved function (Xie et al., 2012). It is however associated with a higher survival rate of the prosthesis (van Strien et al., 2009).

### 1.1.2 PATIENT-SPECIFIC TEMPLATE

A patient-specific template (PST) provides mechanical guidance during surgery, as shown in Figure 1.3(e). This type of CAOS is commercially available, and used in clinical practice. The shape of the PST is based on the patient anatomy, which is derived from CT/MRI images. Each individual guide is designed with CAD programs and manufactured with 3D printing technology. It provides mechanical constraints e.g., drilling and sawing, such that the planned alignment can be reproduced intraoperatively. The use of a PST mitigates some of the problems associated with navigation-based systems



(a) (b) (c)

(d) (e) (f)

**Figure 1.3:** Different forms of CAOS. *a:* active surgical robots, like the ROBODOC surgical assistant, are used in arthroplasty to prepare bones for prosthesis implantation. The surgical actions involved, such as milling and cutting, are performed autonomously, yet under supervision of the surgeon (Cohan, 2001) *b:* with semi-active surgical robots, like the ACROBOT, constraints are imposed on the surgical actions, for instance to prevent milling beyond a plane (Davies et al., 2006) Navigation systems are used in arthroplasty to continuously track the location of bones and instruments relative to each other, thereby assisting the surgeon in carrying out the surgical plan. *c:* Navigation based on preoperatively acquired CT images. *d:* Navigation based on intraoperative probing of the bone surface. *e:* Patient-specific Templates are reverse engineered surgical guides that have a unique fit with the patient and encapsulate one or more surgical actions, such as sawing and drilling *f:* Customizable surgical guides are mechanically adjustable surgical guides that also incorporate one or more surgical actions and rely on a special configuration step to establish a patient-specific unique fit .

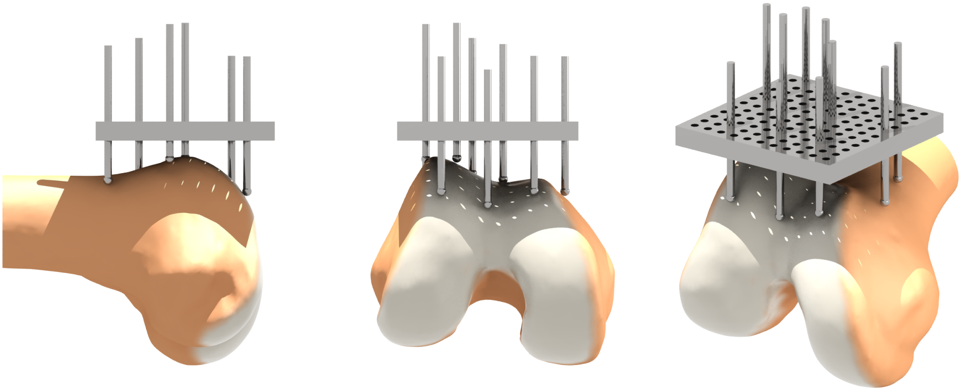


Figure 1.4: Artist impression of our exemplary pin-based CSG, which is discussed in detail in Section 3.4.1.

(such as occlusion and invasive marker trees). However, they can be used only once because they are unique and have an exact fit with a single patient. This also means that the planning cannot be changed during surgery. Furthermore, if the template does not fit correctly, for instance due errors that are introduced in the bone/cartilage reconstruction, the surgeon has to resort to conventional guidance instruments for the alignment of the prosthesis components.

### 1.1.3 CUSTOMIZABLE SURGICAL GUIDE

In this thesis, we focus on Customizable surgical guides (CSG). Much like PSTs, CSGs provide mechanical constraints e.g., drilling and sawing. However, in contrast to PSTs, the patient-specific fit is obtained by adjusting its mechanical configuration. They can be used for multiple patients, because they are mechanically adjustable. Further, the adjustability enables intraoperative changes to the surgical plan. Since the fit is encapsulated in the guide, camera tracking is also no longer required. An example of a pin-based CSG for TKA is shown in Figure 1.4.

For CSGs to become clinically feasible, a complete system for CSG-based arthroplasty is required that incorporates all the necessary hard- and software. In order to achieve this, two major challenges need to be addressed. First, an appropriate CSG needs to be designed that can be used in a clinical setting. Second, software tools need to be developed that allow an orthopaedic surgeon to plan and execute CSG-based arthroplasty. In this project[2], these challenges were addressed in parallel by two PhD students in Biomechanical engineering and Computer science from Delft University of Technology, in close cooperation with Leiden University Medical Center.

[2] Funded by the Dutch technology foundation (STW), Medis Specials BV, and Biomet Nederland BV

## 1.2 GOAL

The goal of this thesis is to develop new visualization tools that enable computer-assisted planning of CSG-based surgery. In order to achieve this goal, we address three major challenges:

**CSG Optimization** The CSG relies on sparse physical contact with bone to achieve a unique, stable and accurate connection with the patient (the fit). The adjustability of the CSG implies that many configurations are possible. However, given the sparsity of the contact, not all of them, and in most cases only a few lead to a good fit. Predicting the fit of an arbitrary CSG is difficult for humans, if not impossible because the physical interaction between the CSG and the bone is very complex. The challenge here is to incorporate an optimization algorithm in the planning that automatically optimizes the fit, given an arbitrary CSG design and patient.

**Contact Visualization** In the preoperative planning of CSG-based surgery, it is important to provide visual cues that help the surgeon inspect the shape of the bone and its surface (especially surface irregularities), and to communicate how the CSG fits on the bone (e.g., what the contact looks like). This is important because a surgeon might want to avoid certain areas on the bone/cartilage, or decide that the optimized contact is not ideal and choose an alternative CSG. Furthermore, the visualization of the contact might provide a rudimentary verification step during surgery, in addition to the intraoperative tactile feedback which is also an important indicator of proper fit. In order to provide the necessary visual cues, we develop an environmental lighting technique that can be added to the planning environment that adds shadows in areas where little environment light can penetrate, in order to enhance the perception of shape, surface detail, and CSG contact, thereby improving the communication of the simulation results with the surgeon. Making this rendering technique near real-time is another challenge that we address in this thesis.

**Maneuverability and Remote Visualization** A realistic planning needs to have a priori knowledge which docking trajectories are valid and which are not (ones that are not physically possible or are associated with too much risk e.g., damaging nerves). The size and location of the incision, shape of the CSG, and obstructions from various tissue types ultimately determines the space in which the surgeon can safely maneuver and subsequently dock the CSG (and other instruments). In essence, this is a visibility problem, where tissues such as skin, bones, ligaments and tendons acts as occluders. We also address the challenge of efficient computation and storage of visibility. The optimization of the CSG is computationally expensive and requires costly and high-end hardware. It is therefore not advisable to run the planning software on a regular machine, machine, as would typically be owned by a surgeon. Ideally, the planning system is easily accessible, using only a state-of-the-art web-browser and an internet connection, irrespective of the device used and its computing capabilities. Besides the broader audience, it also implies that a surgeon can then easily ask for a second opinion, and use it in their communication with the patient. It will be a challenge to develop such a system, while factoring in device-independence.

## 1.3 CONTRIBUTIONS

With this thesis, we present the following contributions to the field of medical visualization and CAOS.

- In Chapter 2, we present the results of a comprehensive and in-depth literature study on applications of CAS where medical visualization plays an important role. In this comprehensive literature review, we introduce classifications based on surgical tasks, visualization techniques, digital surgical plan transfer and CAS application areas. As a result, we were able to identify promising avenues for new research.

- In Chapter 3, we present a generic planning method for the semi-automatic configuration of CSGs using a combination of novel simulation and visualization techniques. Our prototype software implementation predicts how well an arbitrary CSG can reproduce a planned alignment using a combination of physical simulations and a genetic optimization. Furthermore, we validate the proposed method using a pin-based version of the CSG and a variety of 3D printed distal femora.

- We introduce a novel method to improve the visual depiction of contact between an arbitrary CSG and the host bone using an advanced and near real-time volume illumination technique in Chapter 4. This technique allows the surgeon to make better informed decisions and use his visual memory to serve as a guide during the actual procedure.

- In Chapter 5, we present a generic, web-browser based remote medical visualization system. With this system, all computationally intensive operations are performed on a remote server. As a result, no special hard- and software is required to interact with complex medical visualizations. Because the systems accepts multiple concurrent connections, it allows for remote collaboration and improved doctor-patient communication. We present this visualization system in the context of a photo-realistic medical volume visualization.

- Visibility information is vital in many areas of medical visualization, and occurs in many parts of the CAS pipeline e.g., access § 2.2.3 planning, resection § 2.2.4 planning and automatic view finding § 2.3.2. In Chapter 6 we present visibility sweeps, a generic and efficient method to compute and query visibility information in volumetric data sets.

In Chapter 7 we look back at the work and render conclusions. We also discuss exciting and interesting avenues for new research.

# Visualization in Computer-assisted Surgery[1]

Computer-assisted surgery (CAS) refers to the use of computer hard and software to assist surgeons during planning and execution of surgical procedures. The goal of CAS is to tailor surgical procedures to a patient and provide means to follow a surgical plan more closely. CAS commonly leads to smaller incisions, shorter operations, improved risk management and higher surgical accuracy. For this to work, the surgeon relies on pre and intraoperatively acquired medical images such as CT, MRI and 3D ultrasound. These images provide the surgeon with a wealth of anatomical, pathological and functional information, without having to perform a single incision.

Based on this information, a surgeon devises a patient-specific surgical plan, comprising a variety of field-dependent planning tasks. For instance, *access planning* is needed to minimize risk in brain surgery, where obtaining access to a treatment location inside the brain requires incisions that might sever vital neural connections, leading to impaired function and in the worst case, death. The use of computer hard and software also makes it possible to transfer the digital surgical plan to the operating theater, by providing passive and/or active guidance during a procedure, for instance with the use of surgical robots (active) or camera navigation (passive).

CAS has been researched for over three decades, one of the first reported studies is by (Vannier et al., 1983), where reconstructions of patient-specific CT images were used to plan cranio-facial surgery. Since then, advances in medical imaging and medical visualization have led to a more widespread appreciation and adoption of CAS.

[1] This chapter is the result of a collaboration between T. Kroes and C. P. Botha. The initial draft has been thoroughly revised by C. P. Botha and appeared as a book chapter in Visual Computing for Medicine (Preim and Botha, 2013) with the input of B. Preim. This thesis chapter is a condensed and revised version of the original draft, and uses the book chapter as a reference.

The sheer amount of available literature clearly indicates that it is (still) an active area of research.

Medical visualization plays a crucial role in CAS. For instance, without medical visualization it would be impossible to intuitively explore anatomical, pathological and functional information or to virtually perform a surgical action. At the time of this writing, to the authors knowledge, no in-depth study is present that analyzes the exact role of medical visualization in CAS. In order to fill this gap, we collected a comprehensive database of CAS related articles where visualization plays a major role. This exhaustive search has led to a database of over 400 studies. In the remainder of this chapter we give a more thorough description of CAS, and analyze the literature from different perspectives.

## 2.1 INTRODUCTION

Surgery is a specialty of medicine in which diseases are diagnosed and treated with manual and instrumental techniques. Surgery can be open or minimally invasive. Minimally Invasive Surgery (MIS) aims to be less invasive for the patient by reducing the size of the incision and lesions to tissues. Examples of MIS are *laparoscopy*, which refers to procedures performed in the abdomen or pelvis with camera assistance. Another example of MIS is *interventional radiology*, in which procedures are performed under image guidance, mostly involving the use of catheters. In open surgery, the surgeon has direct access to the operating site. Incisions are typically larger because the operating site needs to be accessible by hands and surgical devices.

Applications of CAS typically follow the computer-assisted surgery pipeline as illustrated in Figure 2.1. The CAS pipeline starts with acquisition of patient-specific data. Common types of acquired data are standard planar X-ray, ultrasound, CT and MRI. The acquired images need to be processed (segmented) in order to extract meaningful patient-specific data. In the visualization step, the surgical plan is devised, based on a combination of (3D) visualization of the patient's anatomy/pathology and extracted features of the patient-specific data.

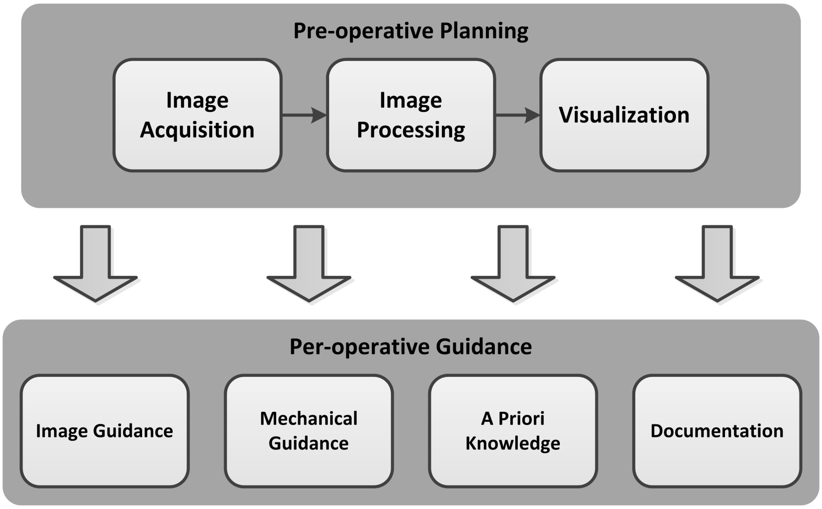


**Figure 2.1:** The computer-assisted surgery pipeline.

The resultant surgical plan is then integrated into the surgical pipeline. The surgeon can be explicitly guided during surgery, by means of image guidance (e.g., tracking of instruments) or through mechanical guidance (e.g., robotic milling). Other types of guidance are more abstract but important nonetheless, they include the obtained a priori knowledge through planning (mental imaging) and through documentation (e.g., visualization of the planning printed on paper). The role of guidance is discussed in Section 2.4.

Visualization plays a vital role in computer-assisted surgical guidance and planning. It allows surgeons to explore anatomy/pathology, perform measurements, predict the outcome of surgery and to rehearse the actual procedure. There are several surveys, overviews and systematic reviews that discuss computer-assisted surgery in general. In this chapter we are primarily interested in applications of computer-assisted surgery where visualization plays an important role. In the remainder of this chapter, we investigate how visualization is used in surgical planning and guidance by introducing four classifications. In Section 2.2 we identify the major tasks involved with surgical planning. In Section 2.3 we discuss how patient-specific data is visually represented, how interaction takes place and the way simulation is integrated in the computer-assisted surgery pipeline. In Section 2.4 we discuss how a digital surgical plan can be transferred to the operating theater. We conclude this chapter with Section 2.5 in which we study the most prominent application areas of computer-assisted surgery: maxillofacial surgery, orthopaedic surgery, neurosurgery and hepatic surgery.

## 2.2 TASKS

When looking at computer-assisted surgery systems in general, we can distinguish five principal tasks. Most of the computer-assisted surgery systems we studied incorporate one or more of these to enable computer-assisted planning and guidance.

### 2.2.1 ANATOMY EXPLORATION

The goal of this task is to extend the existing knowledge of human anatomy with patient-specific anatomy. This is particularly important in cases where the anatomy is very complex e.g., surgery of complex fractures (Cimerman and Kristan, 2007) and pelvic soft tissue surgery (Smit et al., 2012). By looking at the patient-specific images in different ways, using different visual representations (§ 2.3.1), the surgeon is able to better understand important spatial relationships between tissues. The result of this task is an improved anatomical mental image of the patient which helps during surgery to perform complex tasks.

### 2.2.2 PATHOLOGY EXPLORATION

The goal of pathology exploration is to gain insight into pathological structures by exploring (processed) medical images in a (3D) visualization environment. During this task, the surgeon studies the topology, location and embedding of pathological structures. In addition, the surgeon can perform measurements e.g., volume of a tumor or distance from a blood vessel (§ 2.3.2). In most CAS systems, the pathological tissues have been extracted prior in the image processing step, as shown in Figure 2.1 and are visually represented on top of the patient-specific anatomy or in isolation. The result of this tasks is an increased spatial understanding of how pathological structures are embedded, which is important in access (§ 2.2.3) and resection (§ 2.2.4) planning.

### 2.2.3 ACCESS PLANNING

The goal of access planning is to reach a target structure in the human body e.g., tumor while minimizing the risk of lesions to surrounding tissues (e.g., nerves and vessels) or the time it takes to reach the target structure. Without computer assistance, surgeons plan surgery solely on the basis experience and anatomical knowledge. In recent years, we see several examples where access planning is automated (Navkar et al., 2010; Shamir et al., 2010) and improved using visualization methods (Khlebnikov et al., 2011; Rieder et al., 2011), Figure 2.2 shows an example of computer-assisted access planning, where the surgeon is able to quickly explore various trajectory candidates and investigate their associated risk. In a 3D visualization where medical images from various sources are fused together, the surgeon is able to determine the impact of an arbitrary trajectory on the surrounding tissues (Rieder et al., 2008). Good examples are found in neurosurgery, as shown in Section 2.5.2. The result of this planning task is a (curved) trajectory, that is used to guide the surgeon during a procedure, for instance with *Image Guided Surgery*.

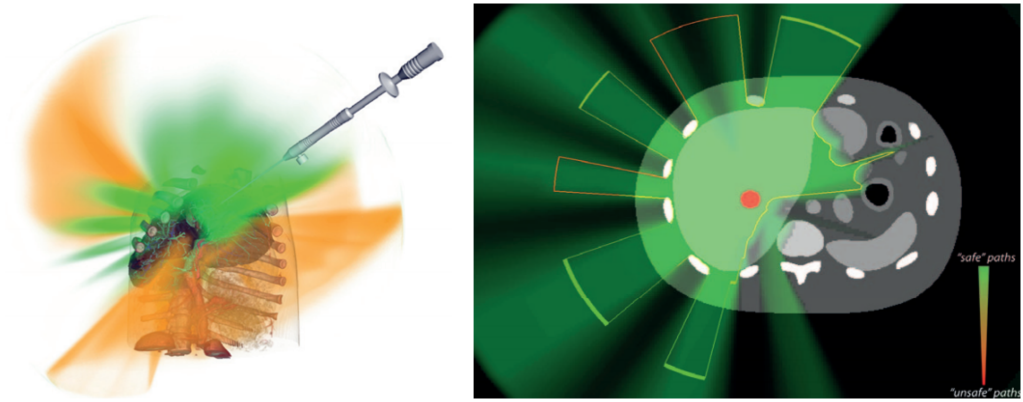


Figure 2.2: Example of how visualization can assist in tumor access planning. *Left:* A tumor behaves as a volumetric light source, which creates an illumination volume which is color mapped and rendered with direct volume rendering. *Right:* 2D representation of the access planning (Khlebnikov et al., 2011)

### 2.2.4 RESECTION PLANNING

In surgery, resection refers to the partial removal of an organ, gland or tumor and surrounding tissue. This is mostly done to remove diseased or necrotic tissue. The goal of resection planning is to increase the chance that risky structures such as tumors are removed and minimize the amount of healthy tissue that is removed. This is achieved by segmentation of at-risk structures, including a safety-margin, and exploration of the surrounding structures to perform a risk assessment. In addition, it can be important to explore the effect of resection on vascular structures surrounding the tumor (Lamata et al., 2008), for instance instance in hepatic surgery, as shown in Section 2.5.4. Virtual resection planning produces visualizations where the planned resection region is clearly encoded as well as the amount of risk in surrounding tissues. In computer-assisted surgery systems with camera guidance, instruments can be tracked with respect to the patient, thus enabling the surgeon to follow the planned resection more closely.

### 2.2.5 RECONSTRUCTION PLANNING

Reconstructive surgery takes place when certain anatomical structures have been damaged by genetic defects, trauma or disease. Reconstructive surgery aims to repair those structures in such a way that it will restore function and lead to an aesthetically pleasing result. In the planning stage, the surgeon can virtually perform the procedure. Bones can be cut and relocated and foreign bodies can be added, for instance to lock two bones together, or plan the alignment of an endoprostheses, as shown in Section 2.2.6. In addition, some computer-assisted surgery systems can also predict the effect on aesthetics, primarily in the face region (Mollemans et al., 2007). The result of this task is a complete surgical strategy. Since the planning is non-invasive, alternative surgical approaches can be tested. Examples of computer-assisted reconstructive surgery planning are discussed in Section 2.5.1.

### 2.2.6 IMPLANT PLANNING

Implants refer to any foreign objects that are introduced into the human body, and include joint prostheses, screws for spinal fusion, and so forth. The goal of computer-assisted implant planning is to choose the appropriate type of implant, minimize damage to surrounding structures during surgery and optimize the function and life span of the implant. In the case of computer-assisted joint replacement surgery, as shown in Section 2.5.3, the 3D visualization and interaction tools allow the surgeon to choose the appropriate size and location of the implant, and to predict the outcome of implantation, for instance the range of motion (Krekel et al., 2006) and altered stress distributions in the bone (Dick et al., 2008). In the case of thoracic pedicle screw insertion, the screws should be secure in the vertebra and should avoid the spinal cord at all costs (Rajasekaran et al., 2007). The result from the planning stage can be used to guide the surgeon during implantation, using for instance optical tracking.

## 2.3 VISUALIZATION TECHNIQUES

CAS systems rely on a wide variety of visualization techniques. They visually represent patient-specific anatomical and pathological data (§ 2.3.1), enable interaction with it (§ 2.3.2) mimic actual surgery and predict the outcome of surgery (§ 2.3.3).

### 2.3.1 VISUAL REPRESENTATION

Visual representation techniques are used in CAS to display all entities that are important in decision making during the planning and execution phase of surgical procedures. What all CAS systems have in common is that they incorporate tools to visually represent the patient anatomy and pathology, as shown in Section 2.3.1.1. Further, they allow the surgeon to perform virtual surgical actions, including foreign bodies such as instruments and implants which also need to be visually represented, as shown in Section 2.3.1.2. Finally, the results from outcome simulation need to be visually integrated in the planning pipeline, as shown in Section 2.3.3.2.

#### 2.3.1.1 ANATOMY AND PATHOLOGY

Visual representation of patient-specific anatomy entails the accurate and faithful visual depiction of patient-specific medical data. In the majority of CAS systems, CT and MRI are used to assist the surgeon pre and intraoperatively. The volume data acquired from these imaging modalities is typically explored in the following ways.

*Multi-planar reformation*, or slicing, is a widely used method to visually represent medical volume data. With this basic technique, the original volume data is re-sampled along an arbitrarily oriented plane. In most cases, reformation along the standard medical views will take place (axial, saggital and coronal), however in some cases (spinal surgery, in which the reformation plane is aligned with the access trajectory, as shown in Figure 2.3) it can be beneficial to align the reformation plane to anatomical/pathological structures using specific landmarks.

In *curved planar reformation* (CPR) (Kanitsar et al., 2002), the original volume data is also resampled, but along a curved planar structure, aligned with for instance a blood vessel. Figure 2.4 gives an example of how CPR can be applied to the planning stage of *orthognatic surgery*, which is part of *oral and maxillofacial surgery* (§ 2.5.1).

The volume data can also be visually represented with *surface rendering*, as shown in Figure 2.6, which requires a polygonal (intermediate) representation of the volume data, typically generated by iso-surfacing or segmented surfaces. This is still a widely used visual representation, as its memory requirements are smaller and users typically have experience with the use of this representation, as its the oldest and most common form of representation.

*Direct Volume Rendering* (DVR), on the other hand, does not require an intermediate representation, such a surface mesh. Ray-casting is currently the de-facto standard

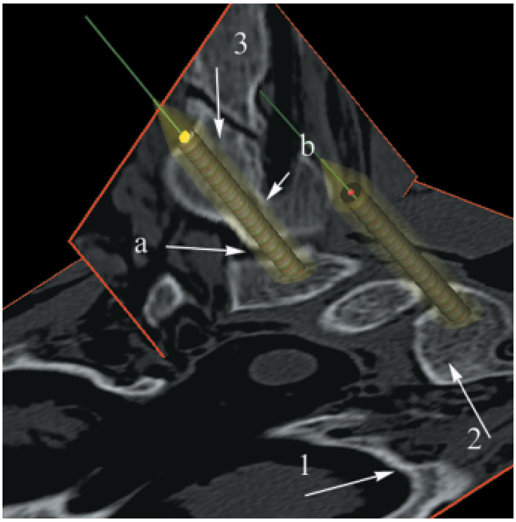


**Figure 2.3:** Virtual planning of spinal fixation surgery. In this example of multi-planar reformation, a slice is aligned to the planned screw trajectory (Van Cleynenbreugel et al., 2002)

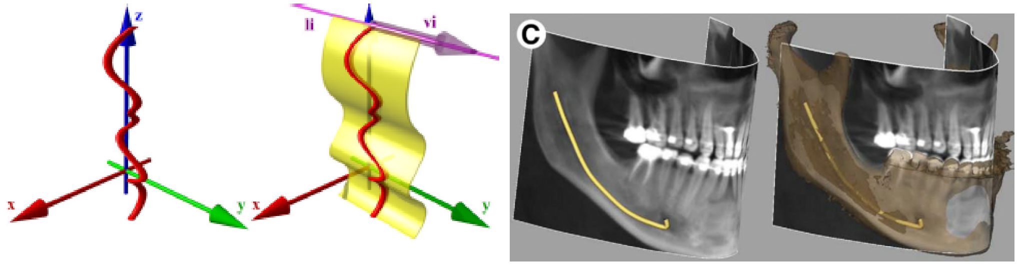


**Figure 2.4:** *Left:* Principle of Curved Planar Reformation. A line and vector of interest are combined to form a curved plane which is used to re-sample the volume data (Kanitsar et al., 2002) *Right:* Example of how CPR can be applied to a volumetric data set in order to create a commonly used panoramic image of the manidble (Orthopantomogram) (Swennen et al., 2009)

for DVR, in which rays are cast through the volume in a front to back manner. Since this type of visual representation does not require an intermediate form, countless different visual representations can be generated on the fly. The majority of CAS systems however still use surface meshes.

The former visual representations are general, they are applied to a wide variety of applications. Other visual representations are more specific to the type of surgery or surgical task. For instance the visual representation of vessels, good examples can be found in hepatic surgery, as shown in Section 2.5.4.

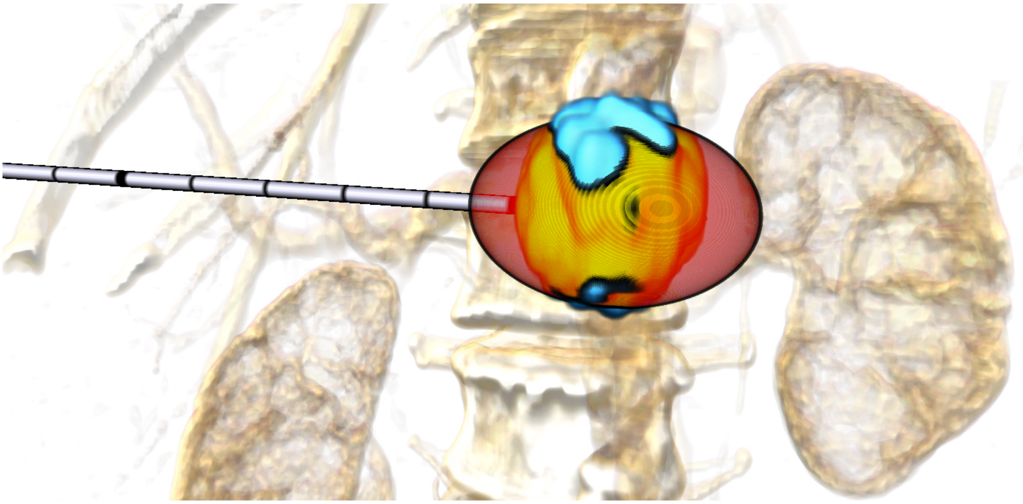


Figure 2.5: Example of volume rendering applied in virtual tumor ablation surgery. The areas surrounding the tumor region are made semi-transparent, this way the view is not obstructed and the surgeon can still view the embedding of the tumor in the surrounding structures (Rieder et al., 2009)

#### 2.3.1.2 VIRTUAL SURGICAL ACTIONS

Some CAS systems allow the surgeon to perform a procedure, or parts of it, virtually. This is called *process simulation* and is discussed in Section 2.3.3.1. In these virtual procedures, target structures are altered e.g., cutting, sawing and rearranging, in some cases to accommodate foreign bodies, such as orthopaedic implants. For instance, in the case of knee replacement surgery, parts of the distal femur bone need to be resected in order to make room for the prosthesis.

#### 2.3.1.3 OUTCOME SIMULATION

Section 2.3.3 discusses numerous simulation techniques that predict the outcome of surgical actions. The goal of visual representation in this context is to enrich the virtual planning visualization with appropriate visual representations of the simulation data. This approach allows surgeons to explore what-if scenarios and also enables them to explore alternative surgical plans, for instance, by interactively predicting stress distributions in the femur under load, with and without endoprosthesis, as shown in Figure 2.13, or predicting the range of motion after inserting an endoprosthesis in the shoulder joint, as shown in Figure 2.14. This thesis deals with the simulation and visualization aspects of adaptive instrument based joint replacement surgery. In this context we describe a method to enhance the visual depiction of simulation results, by modeling the interaction and contact between a *Customizable Surgical Guide* and bone, as shown in Chapter 4.

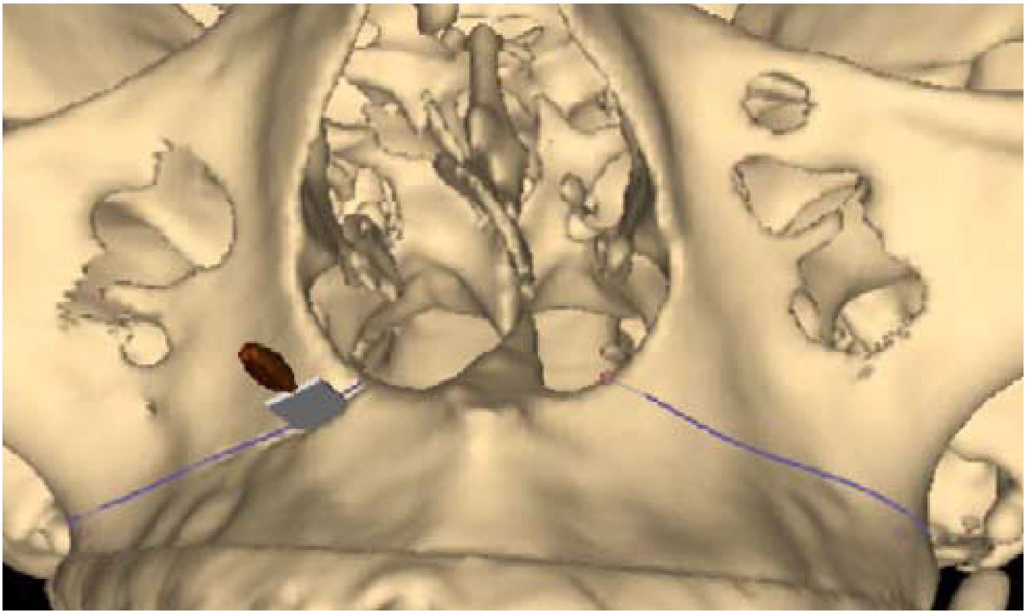


Figure 2.6: Virtual planning of mandibular bone repositioning in *orthognatic surgery*. In addition to a surface rendering of the skull, the virtual planning also includes the planned cutting path and a mock-up of the cutting instrument (Sohmura et al., 2004)

### 2.3.2 INTERACTION

CAS systems rely on various interaction modalities in order to facilitate virtual planning and intraoperative guidance. Basic interaction techniques include camera positioning, data filtering, cutting and object appearance. Depending on the type of CAS, more advanced techniques are required.

Automatic view finding allows a user to define a point of interest, usually by clicking, upon which the most suitable camera parameters are calculated, taking into account the local topology surrounding occlusion, and also previous camera parameters (Kohlmann et al., 2007; Mühler et al., 2007). Determining the occlusion around a point, or region of interest is a computationally expensive task. In chapter Chapter 6, we describe a novel method to improve the performance of occlusion querying in (medical) volume data.

In some types of surgery, the surgeon looses the sensation of touch, either because surgery takes place through a very small incision, or the instruments are indirectly operated by the surgeon, through for instance operating robots like the da Vinci Surgical System. In virtual surgery, the sensation of touch is usually not there to begin with. In these cases, mechanical actuators are used to mimic the tactile feedback a surgeon normally experiences when touching instruments, and tissue. There are still a number of challenges to be solved in implementing haptic feedback for telesurgery, including force sensing at the surgical site, sensor / actuator asymmetry in the practical case where perfect force sensing is not available and the availability of both force and tactile sensing (Okamura, 2004). We briefly discuss a number of examples of haptic feedback in CAS.

Haptic feedback is used in orthopaedic surgery to re-align bone fragments after trauma (Harders et al., 2007) using collision detection and to augment virtual total knee arthroplasty surgery (Jakopec et al., 2003). Haptic feedback is also used in orthognatic surgery, a surgical field where bone re-positioning is quite common (Sohmura et al., 2004). Input data can also be temporal, for instance by mimicking the feeling of a beating heart (Nakao et al., 2002).

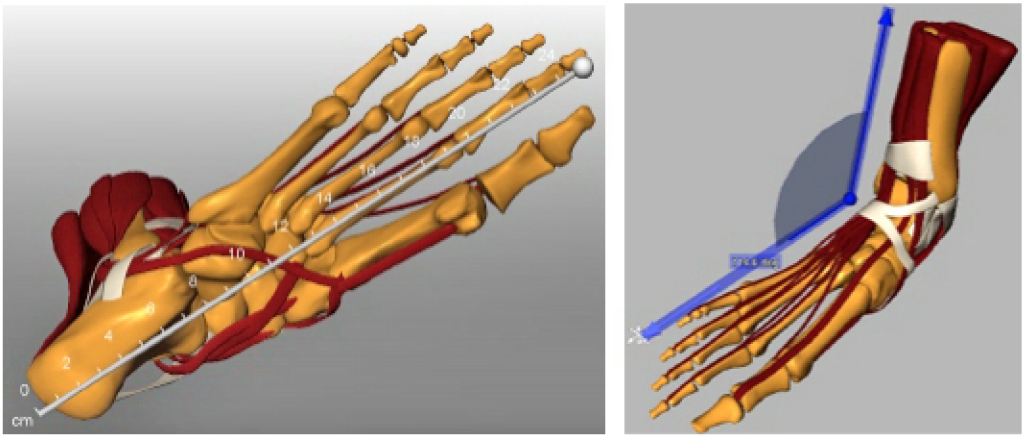


Figure 2.7: Examples of measurement 3D widgets. *Left:* Distance measurement. *Right:* Angle measurement (Preim et al., 2002)

Measurement also plays an important role in CAS. Accurate spatial measurements are needed to devise a virtual surgical plan. Surgeons typically want to measure distance, volume (e.g., hepatic surgery) and angles (e.g., orthopaedic surgery). This is established with virtual measuring devices (also called widgets) which have a visual representation in the planning environment and can be interacted with, as shown in Figure 2.7. Computer-assisted oral and maxillofacial surgery uses underlying cephalometry to measure abnormalities in the facial region, as shown in Section 2.5.1.

### 2.3.3 SIMULATION

We distinguish two major types of simulation in computer-assisted surgery: *process simulation*, which mimics the act of real surgery (e.g., interaction of instruments with soft tissue), as shown in Section 2.3.3.2, and *outcome simulation* which tries to predict the outcome of surgical actions (e.g., effect of mandible relocation on aesthetic appearance of the face), as shown in Section 2.3.3.1. In the next section we discuss both types of simulation in detail.

#### 2.3.3.1 PROCESS SIMULATION

Process simulation is involved with mimicking the act of surgery. This type of simulation is found in surgical simulators, which create virtual environments in which a surgical procedure can be executed on a virtual patient, with the objective to measure surgical skills, educate, train and rehearse surgical procedures. Surgical simulators are typically used for needle insertion procedures (Chentanez et al., 2009), catheters and guidewires, minimally invasive surgery (Cotin et al., 1999; Radetzky, 2002; Lee et al., 2010) and open surgery. The main goal is to establish a user experience that has the highest possible fidelity with real surgery, by accuractely modelling soft tissue, tool tissue interaction (e.g., cutting, suturing), haptic feedback, bleeding and anatomical appearance.

Modeling of soft tissues and its interaction with medical instruments is an active area of research, not only in surgical simulators but also in areas such as computer graphics (Nealen et al., 2006) and animation. Soft tissue simulation serves to mimick the deformable and elastic nature of skin and other non-rigid structures in the human body. In surgical simulators, soft tissues are either simulated by physically based mass-spring models[2] (Lee et al., 1995; Keeve et al., 1996) or by *Finite element* (FE) models (Bro-Nielsen, 1996). Mass-spring models have the advantage of being computationally cheap, but at the expense of physical accuracy and stability. FE models on the other hand are more physically accurate, but also require more computation. All state of the art soft tissue simulators for surgical simulation have in common that certain assumptions are made with respect to the material being simulated e.g., isotropy and homogeneity, which is acceptable for surgical simulation but not for patient-specific simulation in the context of *outcome simulation* in surgical planning.

Especially in surgical simulators, there is a high demand for responsive user interfaces, which implies that soft tissue simulations should run at interactive speeds. With the introduction of *General Purpose computation on Graphics Processing Units* (GPGPU), programmers have extremely potent computational wokhorses at their disposal (Speedups of over 100 x are quite common)[3]. As a result, an increasing number of algorithms are being made compatible for GPGPU, such as a GPU accelerated spring mass system for surgical simulation (Mosegaard et al., 2005), a GPU accelerated FE solver (Taylor et al., 2008; Comas et al., 2008; Han et al., 2010) and cutting simulations with haptic feedback (Courtecuisse et al., 2010).

Most surgical simulators incorporate soft tissue simulation, but there are also examples of virtual cutting performed on bony structures (Pflesser et al., 2002; Agus et al., 2002; Morris et al., 2008), even with bone dust simulation, as shown in Figure 2.8.

The level of surgical immersion can be enhance by increasing the visual appearance of anatomy, as shown in Figure 2.10, and interaction with instruments, for in-

[2] Mass-spring models originate from the computer graphics and animation community where there was a need to interactively convey facial animation

[3] Currently, two major platforms exist: CUDA and OpenCL

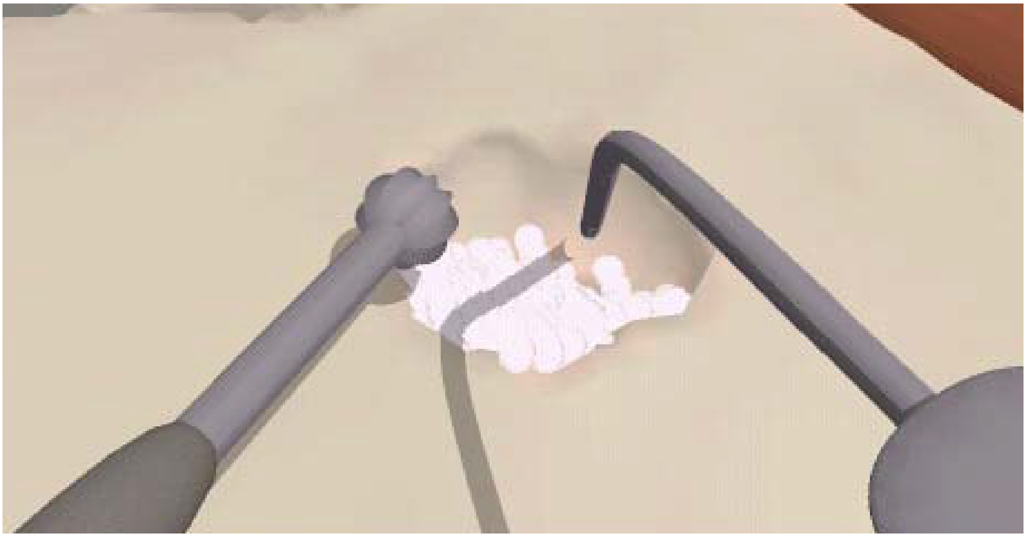


Figure 2.8: Bone dust simulation. The user has removed a volume of bone, which has now accumulated as bone dust. The physical simulation has allowed the bone dust to fall to the bottom of the drilled area. The user is preparing to remove the bone dust with the suction device (Morris et al., 2008)

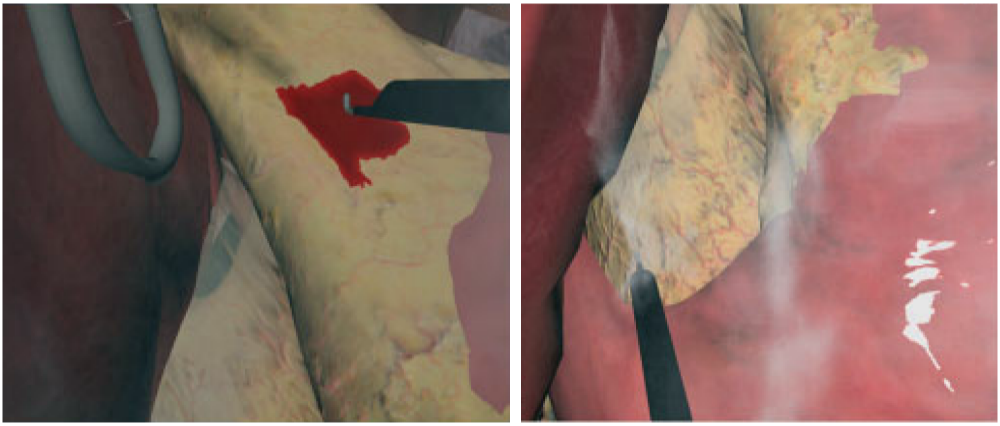


Figure 2.9: Examples of process simulation in a laparoscopy simulator. *Left:* Bleeding simulation. *Right:* Smoke simulation (Halic et al., 2010)

stance bleeding and smoke due to cauterization, as shown in Figure 2.9.

#### 2.3.3.2 OUTCOME SIMULATION

Whereas process simulators model the surgical procedure itself, including for example tissue-tool interaction, outcome simulation techniques attempt to predict the results

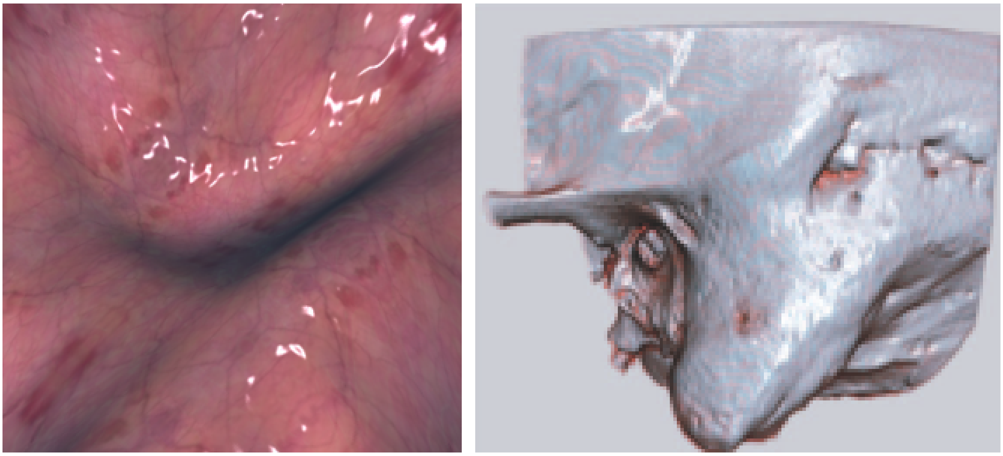


**Figure 2.10:** *Left:* Example of a light reflectance model which increases the perceived level of realism during simulation of *minimally invasive surgery* (ElHelw et al., 2005) *Right:* Emulation of wet bone surfaces during temporal bone surgical simulation (Kerwin et al., 2009)

of virtually performed surgical actions so that *what-if* scenarios can be explored. Outcome simulators are primarily used in CAS to predict the effect of surgical procedures on *aesthetics*, *implant function*, *kinematics* and *blood flow*.

*Aesthetics* simulators are most often found in *plastic* and *oral and maxillofacial* surgery (§ 2.5.1), where reasons for undergoing treatment are often driven by cosmetic needs. Aesthetic outcome simulation primarily involves simulation of soft tissues, and its interaction with muscle and bones. The majority of aesthetic outcome simulation literature concentrates on *orthognatic* surgery, involved with modifying soft and bony structures. State of the art planning systems for *orthognatic* surgery predict the aesthetics with the use of finite element modeling algorithms (Keeve et al., 1996; Westermark et al., 2005; Chabanas et al., 2006; Bottino et al., 2008; Barbarino et al., 2008), of which some also take the underlying muscles into account (Chabanas et al., 2003; Gladilin et al., 2003, 2004; Kim et al., 2010a,b). Other application areas include breast (augmentation) surgery, as shown in Figure 2.12 and rhinoplasty[4] (Xie and Zhu, 2010).

*Implant function* simulators, are used in CAOS to predict the lifespan and function of an endoprosthesis postoperatively. The type, size and placement of endoprostheses has a profound effect on the lifespan of an endoprosthesis, therefore algorithms are developed that simulate the structural integrity of an implanted endoprosthesis. The primary aim is to provide insight in the distribution of forces and stresses in the implant and the surrounding bone, as shown in Figure 2.13. As a result, these algorithms are capable to predict the likelihood that the implant will migrate and subsequently loosen (iannotti et al., 2005; Couteau et al., 2001; Maurel et al., 2005), which is one of

[4] Plastic surgery performed on the nose in order to enhance or reconstruct its appearance and to restore function

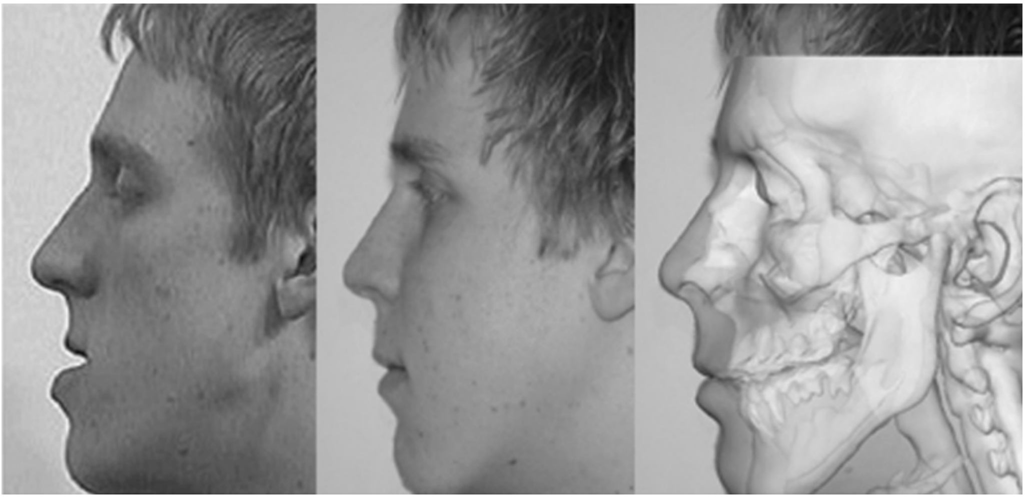


**Figure 2.11:** The two images on the left depict the pre- and postoperative situation respectively, whilst the third image shows an overlay of the postoperative photograph and the altered planning model with predicted facial soft tissue (Zachow et al., 2004).

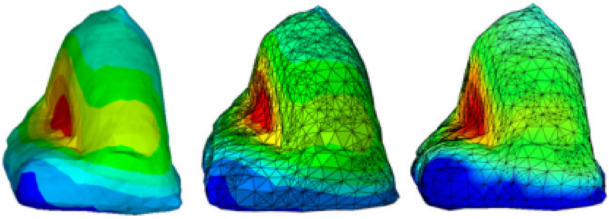


**Figure 2.12:** Total breast tissue displacement for distributions calculated with (from left to right): static implicit FE (ANSYS 11.0), dynamic explicit FE (ABAQUS 6.8) and a dynamic GPU-based FE program (Han et al., 2010)

the most common factors why endoprostheses fail. The results of the simulation are fed back to the surgeon, enabling the refinement of the placement.

*Kinematics* simulators predict the effect of surgical procedures on the human locomotor system. A good example of this is the prediction of the postoperative range of motion (Digioiaiii et al., 2000; Hu et al., 2001; Favre et al., 2008; Krekel et al., 2009) and impingement (Digioia et al., 2000; Hu et al., 2001) after an endoprosthesis has been placed inside a patient. Figure 2.14 shows an example of a shoulder range of motion visualization.

*Blood flow* outcome simulators model and predict the postoperative outcome in terms of changed blood flow. One such example is that of virtual stenting (Cebral et al., 2001; Xiong and Taylor, 2010), where the effect of a placed stent on the blood flow in and around an aneurism is simulated. By predicting the effect a given stent and

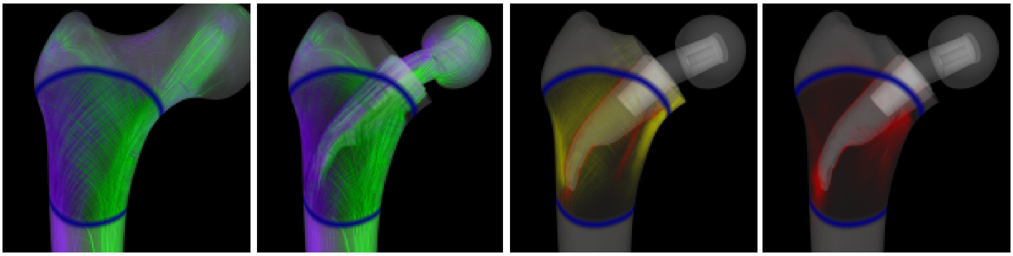


Figure 2.13: Interactive visualizations of simulated stress tensor fields for a human femur under load. From left to right, 1. Stress directions and magnitudes in the physiological state (violet = tension, green = compression). 2. Principal stresses after a simulated implant surgery. 3. Change of normal stresses with respect to the principal stress directions of the physiological state (red = increase, yellow = decrease). 4. Change of shear stresses (Dick et al., 2009)

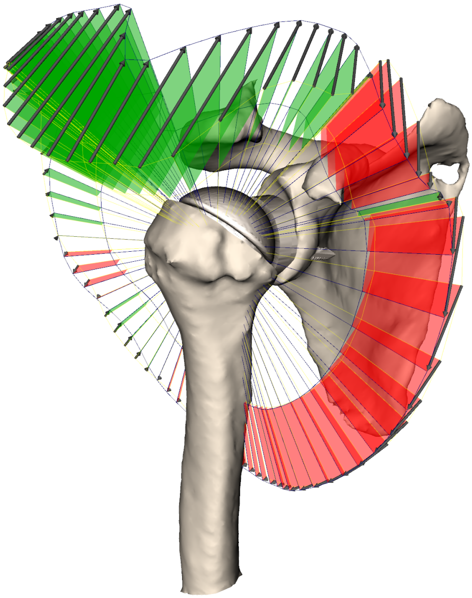


Figure 2.14: Visualization of shoulder *range of motion* (ROM), for a specific prosthesis and placement configuration. Green planes show improvement and red planes a decrease in ROM relative to a user-configured baseline. The ROM is computed and updated on-the-fly (Krekel et al., 2006)

placement will have on for example the pressure and wall shear stress, the placement can be optimized to help guarantee a good outcome. Figure 2.15 shows an example of wall shear stress visualization before and after the virtual stenting.

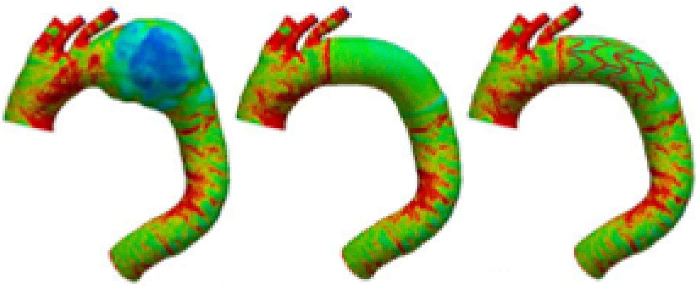


Figure 2.15: *Left:* Situation before the stenting. *Middle:* Model with aneurysm removed. *Right:* Post-stenting situation. The surfaces have been color-coded with the simulated wall shear stress (Xiong and Taylor, 2010)

## 2.4 GUIDANCE APPROACHES

Guidance approaches refer to methods through which a surgical plan can be transferred to the operating theater. There are implicit ways to do so, for instance by relying on the visual memory of the surgeon that was obtained during surgery. There are also more explicit ways in which a surgical plan can be transferred to the operating theater, for instance with the use of camera navigation. In this section we discuss four major ways in which digital surgical plans are transferred to the operating theater. *Mental model* based guidance (§ 2.4.1) relies on tacit knowledge the surgeon obtained during planning. *Documentation* based guidance (§ 2.4.2) relies on passive physical documents that are added to the operating room. *Image* based guidance (§ 2.4.3) tracks instruments in relation to the patient increase accuracy. Finally, *mechanical* guidance (§ 2.4.4) relies on (actuated) mechanical instruments to provide guidance during surgery.

Figure 2.16 shows the four guidance approaches along with a number of sub-types of each approach. These guidance approaches can also be seen as surgical planning *transfer modalities*, as they exist to transfer information from the planning into actual surgical actions.

### 2.4.1 MENTAL MODEL

Mental model based guidance is an implicit way of transferring a (digital) surgical plan to the operating theater. This type of guidance is implicit, it depends on the tacit knowledge the surgeon has acquired during planning. This type of guidance is omnipresent, in both computer-assisted and conventional surgery.

Illustrative examples of mental model based guidance include one the first generated 3-D surface visualizations from CT data for the planning of 200 craniofacial surgeries (Vannier et al., 1983). In another study, the use of 3D visualizations for cardiothoracic surgery planning (twenty-three complex cases) by three experienced sur-

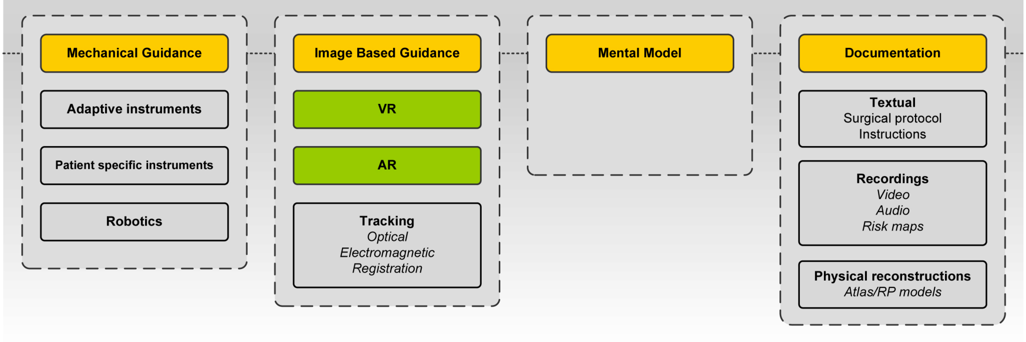


**Figure 2.16:** Schematic overview of the four identified guidance approaches: *Mechanical Guidance, Image Based Guidance, Mental Model* and *Documentation*

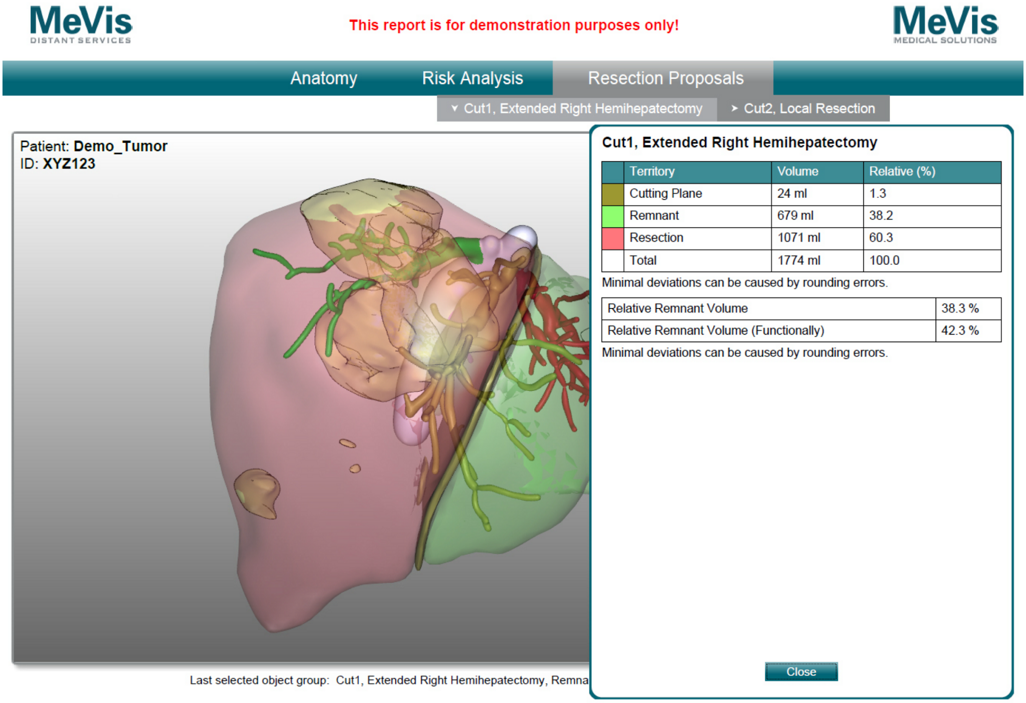
geons was assessed (Hemminger et al., 2005). In this study, the surgeons changed their plans in 65% of the cases based on the visualizations, and in a questionnaire showed a strong preference for the visualizations, claiming that some of the procedures would not even be possible without.

Even though this type of guidance plays an important role in transferring a surgical plan to the operating theater, its isolated use in CAS is contestable since the quality of transfer depends on the surgeon and cannot be accurately measured. On the other hand, solutions based on this have a far easier time gaining traction amongst surgeons, because they are still performing the surgery themselves. In addition, no certification is required, as is the case with explicit systems that need to go in the OR.

### 2.4.2 DOCUMENTATION

In this type of implicit guidance, the planning phase yields a physical document that is used during surgery, one can think of for instance printouts of a 3D visualization or a 3D printed model of the surgical site, enhanced with planning information. The document itself is inactive; it does not participate in the actual surgery or provide active guidance, its sole purpose it to extend the surgeons mental model of the planning information.

Good examples of documentation-based guidance are found in computer-assisted hepatic surgery (§ 2.5.4), where a simplified form of the planning is encapsulated in a document that can be printed and consulted during surgery, for instance resection maps (Lamata et al., 2008) and risk maps (Hansen et al., 2010; Hansen, 2012). MeVis Distant Services, a surgical planning company, also employs this guidance approach to transfer a collection of liver resection proposals in the form of an interactive PDF, to its surgical clients. Figure 2.17 shows an example of such a report. The embedded 3-D visualizations can be interacted with, both through rotation and also through toggling the visibility of the displayed structures.



Patient: **Demo_Tumor**
ID: **XYZ123**

**Cut1, Extended Right Hemihepatectomy**

| | Territory | Volume | Relative (%) |
|---|---|---|---|
| 🟫 | Cutting Plane | 24 ml | 1.3 |
| 🟩 | Remnant | 679 ml | 38.2 |
| 🟥 | Resection | 1071 ml | 60.3 |
| | Total | 1774 ml | 100.0 |

Minimal deviations can be caused by rounding errors.

| | |
|---|---|
| Relative Remnant Volume | 38.3 % |
| Relative Remnant Volume (Functionally) | 42.3 % |

Minimal deviations can be caused by rounding errors.

Last selected object group: Cut1, Extended Right Hemihepatectomy, Remna

**Figure 2.17:** Example of liver resection documentation prepared by MeVis Distant Services for its surgical clients.

Another interesting example of documentation-based guidance is through use of 3D printing. In this case, all relevant planning entities are printed in 3D, and as such serve as a visual (Petzold et al., 1999), and in some cases tactile guide during surgery (D'Urso et al., 1999).

### 2.4.3 IMAGE BASED GUIDANCE

Image Guided Surgery is a class of computer-assisted surgery in which patient-specific preoperative images are augmented with real-time tracked objects, such as surgical instruments, in order to guide the surgeon (Peters, 2006). To achieve this, the preoperative images of the patient are associated or *registered* to the real patient so that the instruments can be visualized in relation to the patient.

IGS starts with acquisition of tomographic images from the patient, preferably as close to the time of operation as possible. At the start of surgery, preoperative imaging and medical instruments are associated with the patient. To this end, the surgeon either identifies corresponding anatomical landmarks on both the anatomy of the virtual and real patient, or obtains images during surgery and registers them to the preoperative images. Once the virtual and real patient are aligned, medical instruments are tracked with respect to the patient by employing one or more tracking technologies (Hassfeld and Mühling, 2001). Optical tracking is currently the de facto standard in IGS, followed by electromagnetic, ultrasound and mechanical tracking.

During surgery, the surgeon utilizes visualizations of patient anatomy along with instruments on a display device near the operating site. This means that the surgeon has to switch focus between the operating site and the display device. In an *Augmented Reality* (AR) environment, the view on the operating site is augmented with patient-specific images and medical instruments. This is accomplished by projecting images on top of the patient anatomy, or projecting images on a video-projected see-through *head mounted display* (HMD), or by combining visualizations with a live video feed of the patient on a separate display. (Lorensen et al., 1993) first reported using such an AR setup for the planning and guidance of a brain tumor resection procedure, while (Fuchs et al., 1998) applied AR for the guidance of laparoscopic surgery. Figure 2.18 shows another compelling example of computer-assisted hepatic surgery, in which the surgical plan is projected onto the surface of the liver.

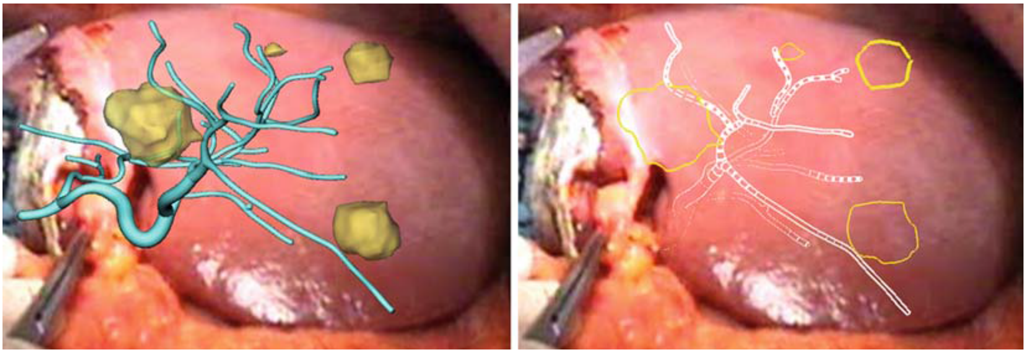


**Figure 2.18:** Augmentation of hepatic surgery by projecting the surgical plan onto the liver. *Left:* Standard surface rendering of the vascular structures and the tumor. *Right:* The same geometry, but this illustrative representation of the planning visually encodes deeper seated vessels by adjusting their opacity based on the depth in the liver. This provides extra visual cues to the surgeon where lesions are located, hence the surgical resection plan can be followed more closely (Hansen et al., 2010)

### 2.4.4 MECHANICAL GUIDANCE

Mechanical guidance provides guidance during surgery using actuated or non-actuated mechanical devices. Mechanical guidance can be achieved with surgical robotics (§ 2.4.4.1), patient-specific instruments (§ 2.4.4.2) and adaptive instruments (§ 2.4.4.3).

#### 2.4.4.1 SURGICAL ROBOTICS

Surgical robots allows surgeons to follow a surgical plan more precisely with the use of robotic manipulators.

Tele-manipulators are the most common form of surgical robotics. In this type of surgery, the surgical input is converted to electrical signals and transmitted digitally to the robotic actuators at the operating site, similar to fly-by-wire systems in aircrafts. This implies that, as long as the communication latency is acceptable (sending commands and receiving the video stream), the robot can be manipulated, irrespective of the distance between operation console and surgical robot. Further, the fact that the input from the surgeon is converted to electronic signals enables the reduction of physiologic tremors and allows motion to be scaled.

Examples of tele-manipulators are the Da Vinci (Intuitive Surgical Inc., Mountain View, CA), as shown in Figure 2.19 and Zeus (Computer Motion Inc., Santa Barbara, CA) system. Tele-manipulators are commonly seen in minimally invasive, or keyhole surgery.

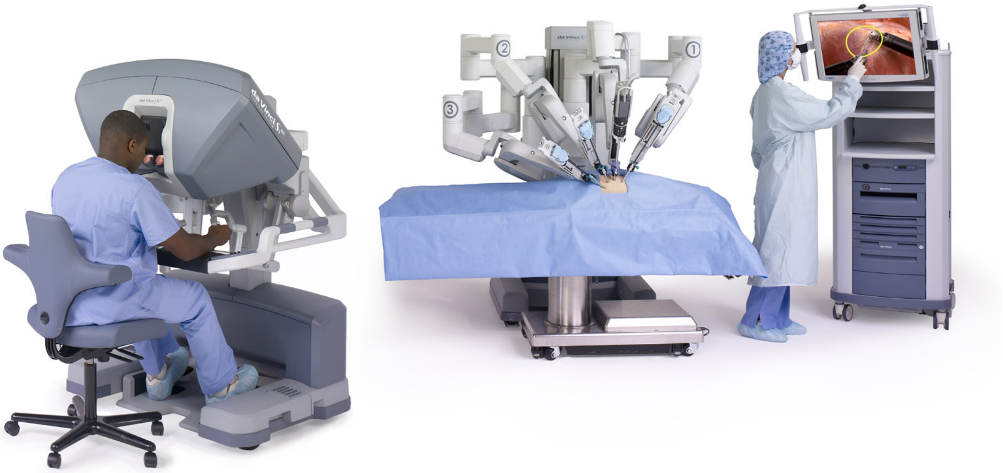


Figure 2.19: The DaVinci surgical robot.

Surgical robots are also used to exactly replicate a surgical plan, for instance by milling the surface of a bone to accommodate an endo-prosthesis. In some cases surgical robots are used to impose motion constraints on surgical instruments e.g., constrain drilling along a planned axis.

#### 2.4.4.2 PATIENT-SPECIFIC INSTRUMENTS

A *patient-specific template* (PST), is a guidance device that has a unique fit with the patient (typically by attaching to bone structures) and encapsulates the surgical plan, or parts of it, by incorporating holes and slits for constrained sawing and drilling (and other surgical actions) according to the surgical plan. PSTs are manufactured using rapid-prototyping technology in the majority of cases, but also CNC milling can be used to manufacture PSTs. PSTs require no specialized auxiliary hardware for navigation, but a surgeon can also not make adjustments to the planning during the procedure. This aspect can be problematic should the template not fit correctly which implies that traditional surgical alignment instruments should be present during surgery as a fallback strategy. Furthermore, these guides are used once, after which they are disposed of.

PSTs are particularly useful in surgery on rigid anatomical structures e.g., bones, and is therefore commonly seen in orthopaedic surgery, as shown in Section 2.5.3. PSTs are successfully employed to perform pedicle screw insertion (Goffin et al., 2001; Van Cleynenbreugel et al., 2002; Singare et al., 2009; Owen et al., 2007) and knee (Hafez et al., 2007; Radermacher et al., 1998; Rademacher, 1999), hip (Raaijmaakers et al., 2010; Audenaert et al., 2011) and shoulder replacement (Metzger and Vanasse, 2010) surgery. An example of a PST is shown in Figure 2.20 that illustrates a patient-specific device for guide-wire positioning in hip surface replacement (Raaijmaakers et al., 2010).

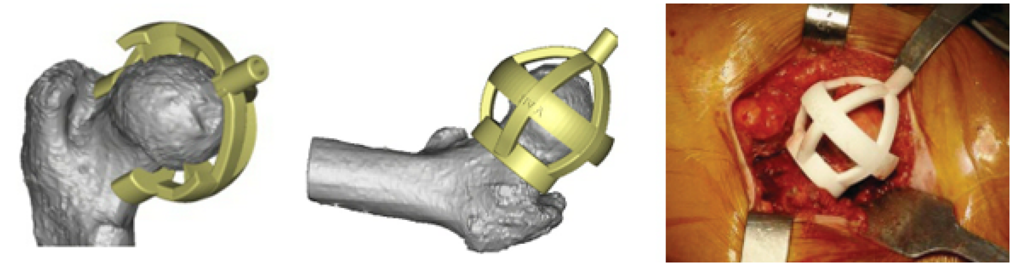


**Figure 2.20:** Drill guide for hip resurfacing surgery (Raaijmaakers et al., 2010)

### 2.4.4.3 CUSTOMIZABLE SURGICAL GUIDES

The *customizable surgical guide* (CSG), which is discussed in detail in Chapter 3, extends the idea of PSTs (§ 2.4.4.2). CSGs achieve a unique fit with the patient through patient-specific settings, in contrast to PSTs which permanently encode the unique fit with the patient with rapid-prototyping. CSGs also allows the surgeon to perform (constrained) surgical actions according to a plan.

A good example of a CSG is the HipSextant (Steppacher et al., 2011). During planning, the surgeon specifies three anatomical points and the desired hip cup orientation based on 3-D visualizations of the patient CT data, after which the planning software automatically calculates the adjustable parameters of the HipSextant. During surgery, the HipSextant fits on the bony anatomy of the patient. If for example the desired cup orientation is changed, the instrument can be adjusted to follow suit. Figure 2.21 shows the virtual and the real HipSextant device.

In Section 3.4.1 we introduce a pin-based CSG for knee replacement surgery and demonstrate how its settings can be optimized for a patient.

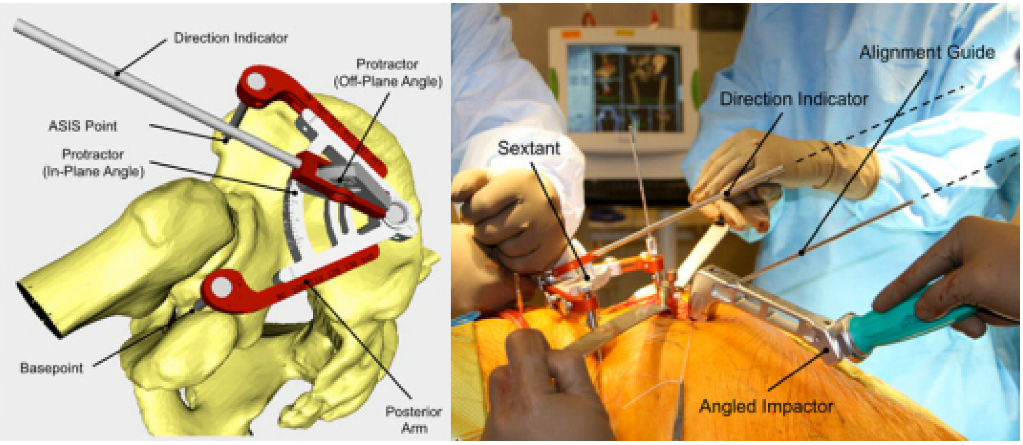


Figure 2.21: Adaptive instruments for placement of artificial cup in hip surgery. *Left:* Preoperative planning. *Right:* Intraoperative guidance (Steppacher et al., 2011)

## 2.5 APPLICATION AREAS

CAS is applied in several areas of surgery. We identify four areas of surgery where, judging by the amount of literature the infiltration is the highest: oral and maxillofacial surgery (§ 2.5.1), neuro surgery (§ 2.5.2), orthopaedic surgery (§ 2.5.3) and hepatic surgery (§ 2.5.4). In the following section we discuss the role of visualization in each type.

### 2.5.1 ORAL AND MAXILLOFACIAL SURGERY

Oral and maxillofacial surgery (OMS) refers to surgery performed on the craniomaxillofacial complex, consisting of the skull, mouth, jaws, neck, face and surrounding tissues such as nerves and muscles[5]. Severe facial trauma, tumors (benign or malignant), congenital defects in the facial region and aesthetics are the most prominent reasons for undergoing OMS, which primarily involves the reconstruction, cutting and relocation of bones in the facial area (orthognatic surgery), and treatment of soft tissues of the face (plastic and reconstructive surgery).

The main goal of OMS is to reconstruct deformities in the craniomaxillofacial complex. Deformities in this area are particularly visible and have a profound effect on the

[5] The brain and eyes are not part of OMS

well-being of the patient (Mollemans et al., 2007). A significant part of the planning and execution of OMS is dedicated to predicting the aesthetic outcome so that it can be optimized, and used in doctor-patient communication. However predicting the outcome is technically challenging as the interaction between bone and soft tissues such as adipose fat and muscles is complex.

Computer-assisted OMS is primarily applied to orthognatic and dental implant surgery (Ruppin et al., 2008; Galanis et al., 2007). For an in-depth review of computer-assisted OMS we refer the interested reader to (Hassfeld and Mühling, 2001).

#### 2.5.1.1 TASKS

The first tasks of the surgeon is to extend his anatomical knowledge with the patient-specific anatomy and pathology. Next, the deformity that needs to be corrected is investigated, as shown in Figure 2.22.

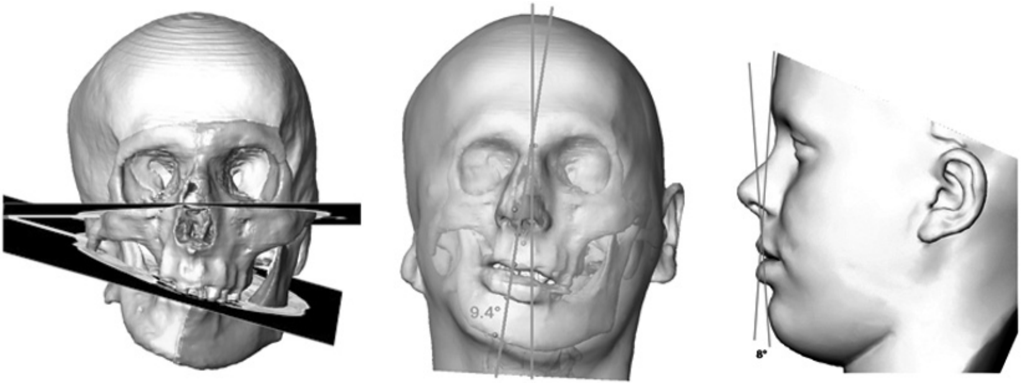


Figure 2.22: Visualization tools are used to study facial asymmetry and other deformities (Zachow et al., 2006)

Once the deformity is understood, a planning can be made that includes a surgical protocol that specifies how to cut and relocate bones, as shown in Figure 2.23, how to fixate them, and how to handle soft tissues. OMS also includes tumor resection, which is why sometimes also access (§ 2.2.3), resection (§ 2.2.4) and implant planning (§ 2.2.6) need to take place (Hassfeld and Mühling, 2001).

#### 2.5.1.2 VISUALIZATION

In Section 2.5.1 we already stressed the importance of accurate predictions of aesthetic outcome in OMS. *Outcome* simulation in OMS is used in the planning phase to predict the effect of surgical actions (e.g., bone re-positioning) and to inform the patient about the possible outcome of surgery. The core enabling technique here is soft tissue simulation. The most prominent soft tissue simulation methods used in OMS

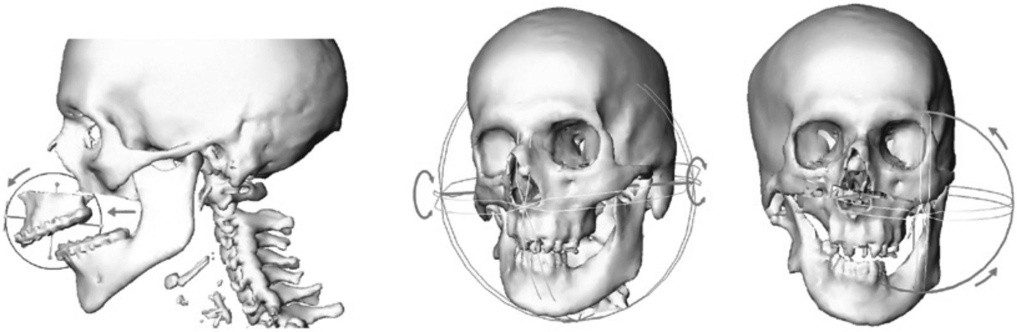


Figure 2.23: 3D widgets are used to virtually plan the relocation of bones (Zachow et al., 2006)

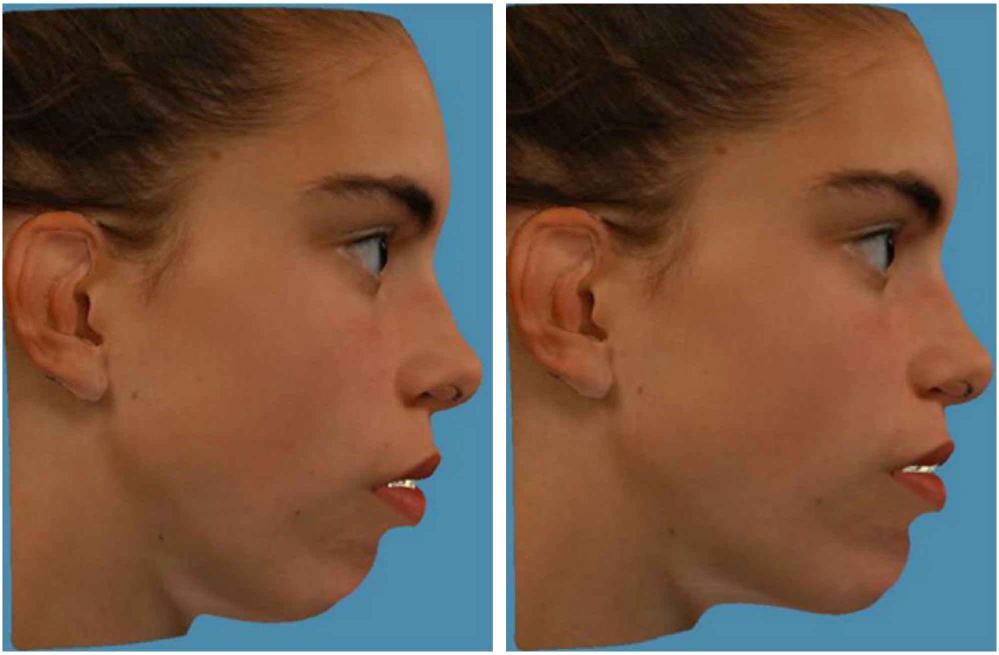
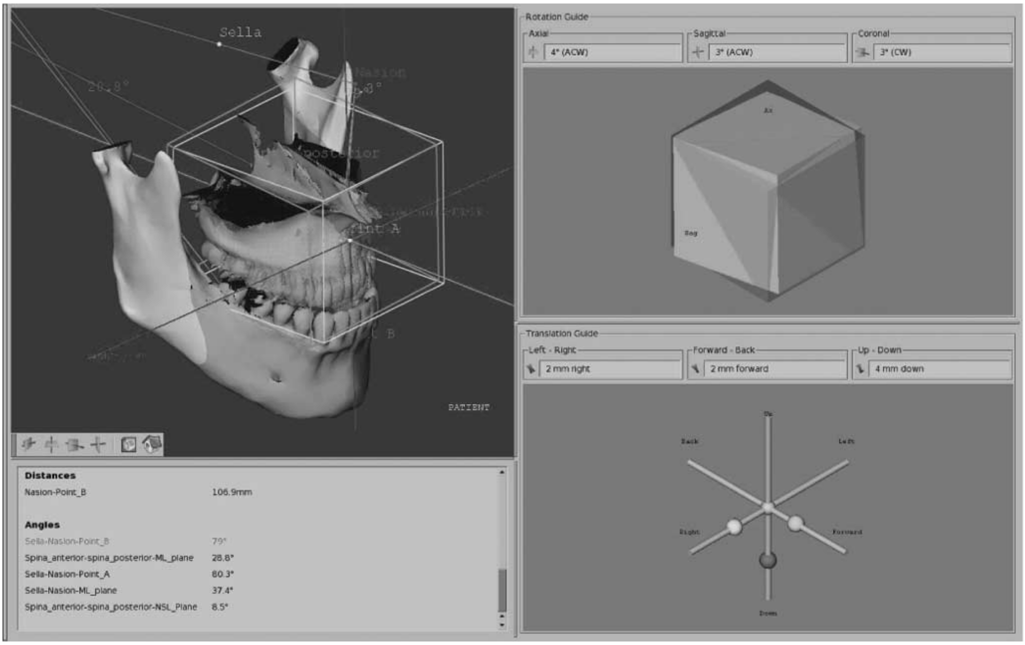


Figure 2.24: *Left:* Visualization of the textured preoperative face of the patient. *Right:* Predicted postoperative situation (Mollemans et al., 2007)

planning are the finite element method (FEM) (Bro-Nielsen, 1998), the mass-spring model (MSM), and the mass tensor model (MTM) (Cotin et al., 2000). Each method has it advantages/disadvantages, for OMS applications it is important that the results are accurate enough, while still allowing the surgeon to explore different scenarios within reasonable time. In a quantitative and qualitative comparison of linear and non-linear FEM, MSM and MTM methods, (Mollemans et al., 2007) found MTM and linear FEM to be the most accurate, with MTM being significantly faster than the other approaches. An example of outcome prediction is seen in Figure 2.24 which shows the preoperative and postoperative side views of the patient, enhanced with photographic information.

#### 2.5.1.3 GUIDANCE

Computer-assisted OMS can employ mechanical, electromagnetic, ultrasound or optical navigation. By doing so, the surgeon can follow the surgical plan more closely, by tracking instrument with respect to the patient, as shown in Figure 2.25. This requires an accurate registration, and in some cases a stereotactic frame or anatomical/artificial markers are used to enhance the registration.



Rotation Guide
Axial: 4° (ACW)
Sagittal: 3° (ACW)
Coronal: 3° (CW)

Translation Guide
Left - Right: 2 mm right
Forward - Back: 2 mm forward
Up - Down: 4 mm down

| Distances | |
|---|---|
| Nasion-Point_B | 106.9mm |
| **Angles** | |
| Sella-Nasion-Point_B | 79° |
| Spina_anterior-spina_posterior-ML_plane | 28.8° |
| Sella-Nasion-Point_A | 80.3° |
| Sella-Nasion-ML_plane | 37.4° |
| Spina_anterior-spina_posterior-NSL_Plane | 8.5° |

Figure 2.25: Example of image guided OMS. The visualization in the top-right corner shows two overlapping cubes. How much the cubes coincide defines how well the planned alignment is reproduced (Chapuis et al., 2007)

### 2.5.2 NEUROSURGERY

Computer-assisted neurological surgery, or neurosurgery, is applied in various areas of neurological surgery: tumor resection, epilepsy/pain treatment, motion disorders, biopsies (Hall and Truwit, 2008) and parkinsons disease treatment. One of the earliest domains of medicine to adopt CAS was neuro surgery (Germano, 1995; Grimson et al., 1996; Kikinis et al., 1996). Neurosurgery can be open, which requires a craniotomy (part of the skull is temporarily removed to gain access to the surgical target, as shown in Figure 2.27) or with MIS. A substantial amount of literature is available on the topic of computer-assisted neurosurgery.

The human brain gives us personality, lets us speak and move, and controls the biological processes in the human body. Damage to the brain can have a devastating effect on the functionality of the patient, in the worst case, a patient could even loose their life. Performing surgery on the brain is complex because all brain functionality is concentrated in a delicate and relatively small piece of soft tissue. Preoperative imaging (anatomical and functional) in combination with advanced visualization techniques have been able to reduce the risks associated with neurosurgery, and to improve the surgical outcome (Maciunas, 2006).

#### 2.5.2.1 TASKS

Computer-assisted neurosurgery starts by exploring the anatomy (§ 2.2.1) and pathology (§ 2.2.2) of the patient. Next, it is important to study how the treatment location (for tumor resection (§ 2.2.4) or implant placement (§ 2.2.6)) is embedded in the surrounding tissues, in order to plan an access trajectory (§ 2.2.3) that minimize risk to those structures, as shown in Figure 2.26.

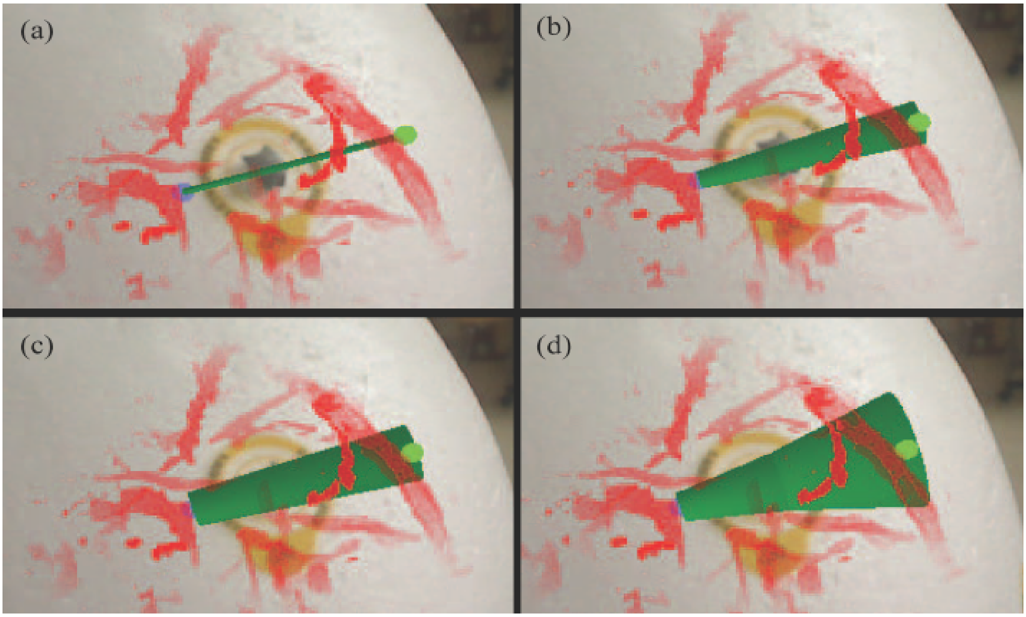
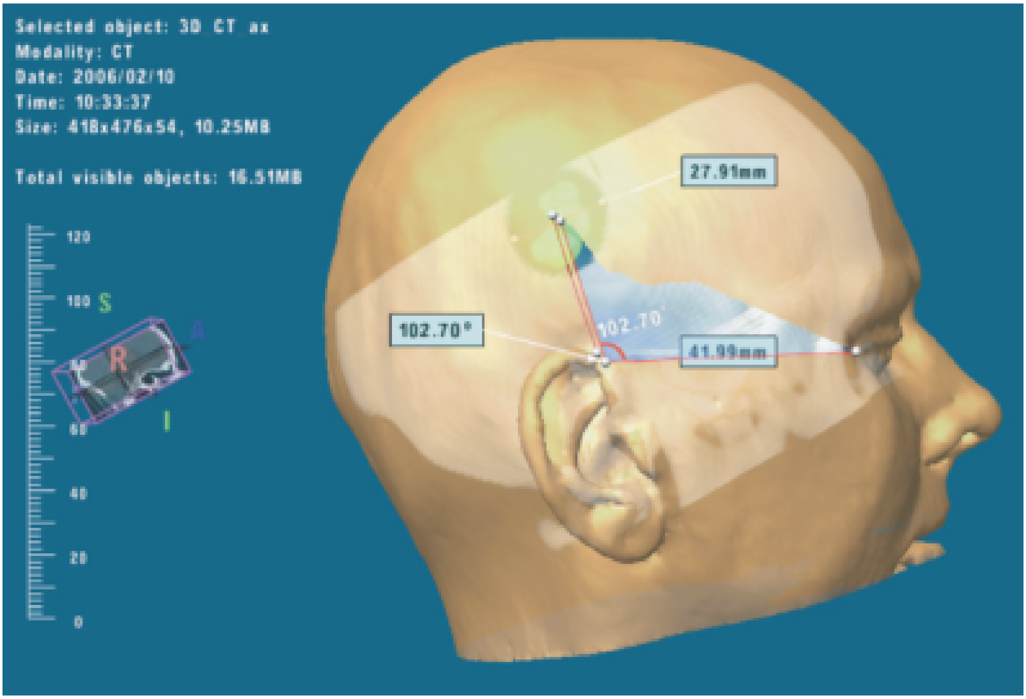


Figure 2.26: In this visualization, risks of multiple candidate trajectories and their associated risks are quantified in order to assist the neurosurgeon in selecting the safest path (Shamir et al., 2010)

#### 2.5.2.2 VISUALIZATION

One of the challenges in computer-assisted neurosurgery visualization is to combine images from different modalities in order to enhance the planning and guidance pipeline, as shown in Figure 2.28. Computer-assisted neurosurgery uses Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) to study patient anatomy and pathology. Computed Tomography Angiography and Magnetic Resonance Angiography are used to enhance blood vessels, and functional MRI (fMRI) is used to image functional areas of the brain that are at risk. Diffusion Tensor Imaging (DTI) is a special MR protocol for capturing fibrous in-vivo structures, such as brain white matter and muscles, and relies on the measurement of water self diffusion rates. Neuro surgeons are interested in what the neuro connections formed by white matter fibers, of a specific patient look like, because during surgery, they want so sever as few as possible (Filler, 2009). Positron Emission Tomography (PET) is a nuclear imaging technique that is used to locate tumors.



Selected object: 3D CT ax
Modality: CT
Date: 2006/02/10
Time: 10:33:37
Size: 418x476x54, 10.25MB

Total visible objects: 16.51MB

Figure 2.27: This figure illustrates the angle and distance measuring tools for the determining the appropriate craniotomy position (Stadie et al., 2008)

Visualization allows the surgeon to view the anatomical, pathological and functional data in its entirety, rather than individual slices. The advantage of doing so is that the surgeon has the freedom to estimate the risk of numerous alternative access paths, taking into account motor, speech and other functional areas without ever touching the patient, as shown in Figure 2.28. Non-rigid registration is an important challenge in achieving this goal (Archip et al., 2007).

Figure 2.28 is an example of how a neurosurgeon can plan the access to the target structure, while taking into account surrounding structures at risk, and Figure 2.27 shows a craniotomy planning visualization, where various measurement tools are employed, as shown in Section 2.3.1.

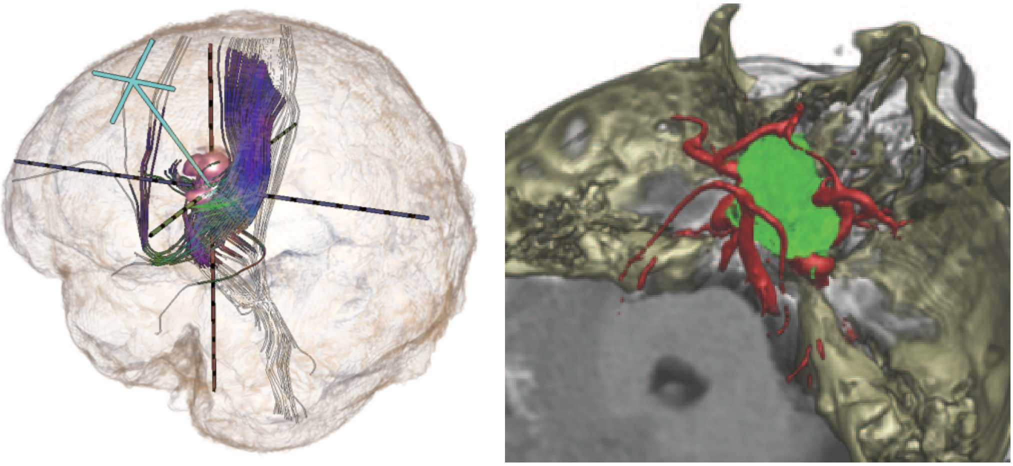


**Figure 2.28:** *Left:* Computer-assisted access planning for brain tumor removal, consisting of an access path (cyan) lesion and functional data (DTI) (Rieder et al., 2008) *Right:* Combined rendering of, CT, MRI and MRA for tumor resection planning (Beyer et al., 2007)

### 2.5.2.3 GUIDANCE

Computer-assisted neurosurgery uses *mental model* (§ 2.4.1), *mechanical* (§ 2.4.4) and *image* (§ 2.4.3) guidance to follow a surgical plan more accurately (Grunert et al., 2003).

In *mental model* based guidance, the surgeon builds up a mental image of all landmarks e.g., nerves, bony landmarks and vessels that are useful for carrying out the surgical procedure. During the procedure, he uses this patient-specific mental image to avoid structures at risk and to perform the procedure with greater accuracy. A good example is micro-neurosurgery, which is performed with a surgical microscope and with miniaturized instruments.

In *mechanical* guidance, a stereotactic frame in concert with preoperative imaging, is used locate small surgical treatment targets inside the brain (Kelly and Alker Jr., 1981). This method requires a straight trajectory to the target structure.

More common is *image* based guidance that combines pre and intraoperative imaging with tracked instruments to increase the accuracy and flexibility with which procedures can be carried out. Tracking is achieved with optical, electromagnetic and ultrasound tracking systems or via intraoperative imaging.

*Mechanical* and *image* based guidance have in common that their accuracy depends on the amount of brain-shift that has occurred between the time of image acquisition and surgical procedure. Brain-shift is a phenomenon that occurs because the brain is comprised of a variety of soft tissues which are subject to gravity, motion and surgical (re)actions (resection and swelling) that alter its shape during surgery (Roberts et al., 1998). This change in shape can be quite drastic, rendering any preoperative planning and risk assessment invalid. With intraoperative imaging the surgical plan can be altered to account for brain-shift (Gering et al., 1999).

### 2.5.3 ORTHOPAEDIC SURGERY

Orthopaedic surgery involves the treatment of musculo-skeletal trauma, degenerative diseases, infections, congenital disorders and tumours in or near bones. *computer-assisted orthopaedic surgery* (CAOS), is a branch of CAS which is tailored to orthopaedic surgery. There are four prominent applications of CAOS. In *joint replacement surgery*, damaged joints are (partially) replaced with artificial endoprostheses.

Computer-assisted *fracture fixation* is needed in complex trauma surgery where computer-assisted planning helps to virtually re-align the bones. Computer-assisted *spinal fixation*, a procedure in which two adjoining vertebrae are fused, helps to the reduce the risk of lesions to the spinal cord. Computer assistance is also used to plan bone *tumor resection* and subsequent endoprosthesis planning, as shown in Figure 2.29.

#### 2.5.3.1 TASKS

The CAOS pipeline starts with a thorough exploration of patient anatomy (§ 2.2.1) and pathology (§ 2.2.2). The patient-specific information obtained during this task is used throughout the planning phase. The tasks that follow depend on the type of CAOS. For example, in tumor resection planning, as shown in Figure 2.29 the following tasks are relevant: resection planning, access planning and implant planning. Based on preoperative imaging and other measurements, CAOS systems enable surgeons to visualize and measure parameters of the surgical site, for example to help decide on the size and type of an endo-prosthesis.

#### 2.5.3.2 VISUALIZATION

The role of visualization in CAOS is manifold. First is the accurate and faithful representation of bony structures, pathology and auxiliary representations such as virtual implant models, fixators, screws and instruments. CAOS systems typically use surface rendering as a visual representation modality (Fleute et al., 1999; Stindel et al., 2002; Atmani et al., 2007). Volume rendering is almost never used in CAOS, primarily

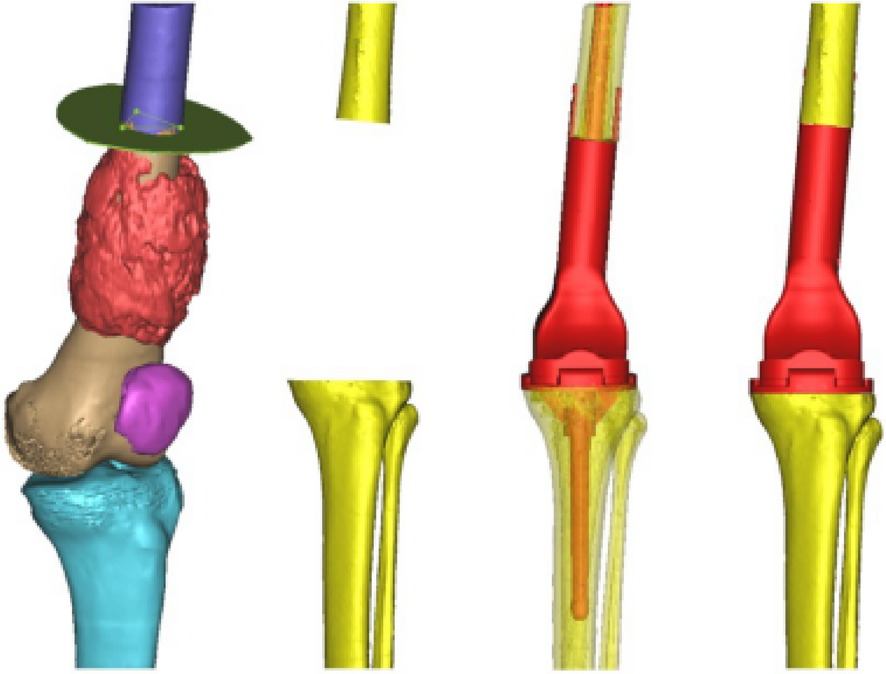


Figure 2.29: Tumor resection and endoprosthesis planning. **a)** Choosing the resection plan **b)** Bone stock after resection **c)** Verification of the endoprosthesis alignment **d)** Final implanted endoprosthesis (Subburaj et al., 2009)

because orthopaedic surgeons are used to a slice-based inspection of the anatomical data, as shown in Section 2.3.1. Furthermore, almost all outcome simulators (§ 2.3.3.2) accept only surface meshes as input. Only in special cases where inspection of 3D shape(s) and their arrangement is vital does it pay of to use 3D visualizations of the anatomy, for instance in *fracture fixation*, as shown in Figure 2.30.

CAOS systems typically rely on high resolution, preoperatively acquired CT data sets with high bone vs. soft tissue contrast. However, in some cases, real-time intraoperative imaging modalities such as fluoroscopy[6] suffices (Langlotz and Nolte, 2004), for instance in pedicle screw alignment. *Surgeon defined anatomy* (SDA) is another type of intraoperative imaging, in which anatomical structures such as bones are probed during surgery using a tracked instrument, as shown in Figure 2.31. This method can be further enhanced by deforming statistical shape models of the bones in question to fit the acquired surface landmarks (Fleute et al., 1999; Stindel et al., 2002).

Outcome simulation is being integrated more in the visualization pipeline. Applications range from the simulation of implant stability (Dick et al., 2008, 2009; Maurel

[6] A real-time x-ray based planar imaging modality

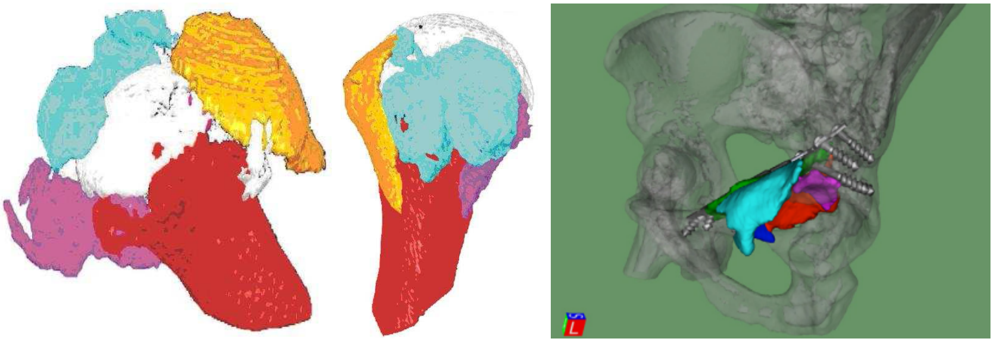


Figure 2.30: *Left:* Before and after virtual fragment fixation of a fractured humerus for trauma surgery (Harders et al., 2007) *Right:* Virtual pelvic fragment fixation for trauma surgery. Individual fragments are color-coded. By making the pelvis partially transparent, the direction and length of the fixation screws is easier noticeable (Cimerman and Kristan, 2007)

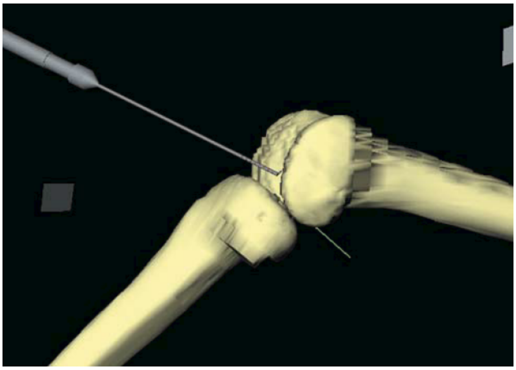


Figure 2.31: An example of surgeon defined anatomy. Intraoperative imaging is achieved by acquiring points on the surface of the bone, which are used to create a matching instance from an *active shape model* (ASM). This technique provides a very realistic, individualized representation of the operated anatomy without any conventional imaging modality (Langlotz and Nolte, 2004)

et al., 2005) to postoperative ROM simulation (Digioiaiii et al., 2000), impingement detection algorithms (Digioiaiii et al., 2000; Krekel et al., 2009) and 4D analysis of muscle contraction, as shown in Figure 2.32. The role of visualization is becoming more prominent since outcome simulation is becoming more of an iterative process[7].

#### 2.5.3.3 GUIDANCE

The guidance methods employed in CAOS strongly resemble those found in *Oral and Maxillofacial Surgery*, as shown in Section 2.5.1. Both types of surgery primarily deal

[7] State of the art outcome simulators are able to run fast, in some cases even in real-time, this is due to enhanced algorithms and increased computational resources that state of the art hardware offers

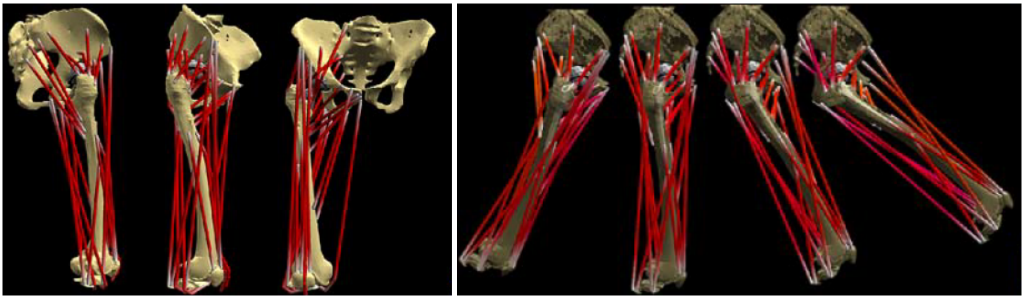
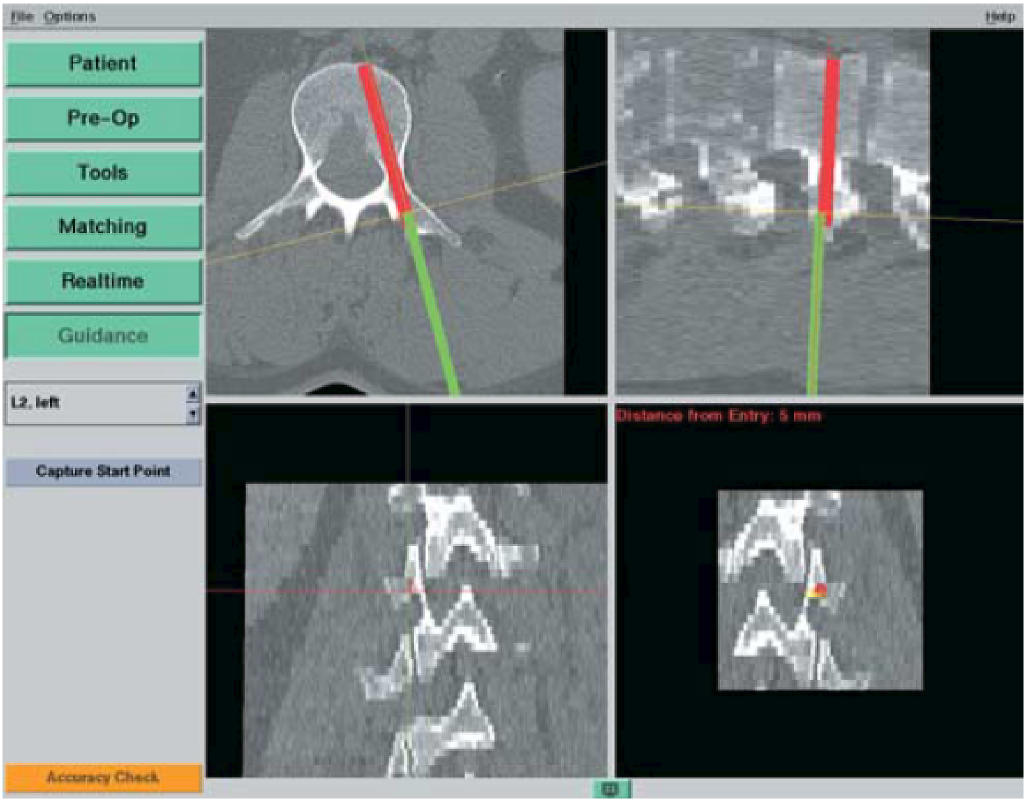


Figure 2.32: *Left:* 4D muscle model for analysis of muscle dynamics. *Right:* 4D analysis of muscle contraction during flexion and extension of the hip joint (Otake et al., 2005)

with bony surfaces. Figure 2.33 shows an example of pedicle screw placement with optical navigation.



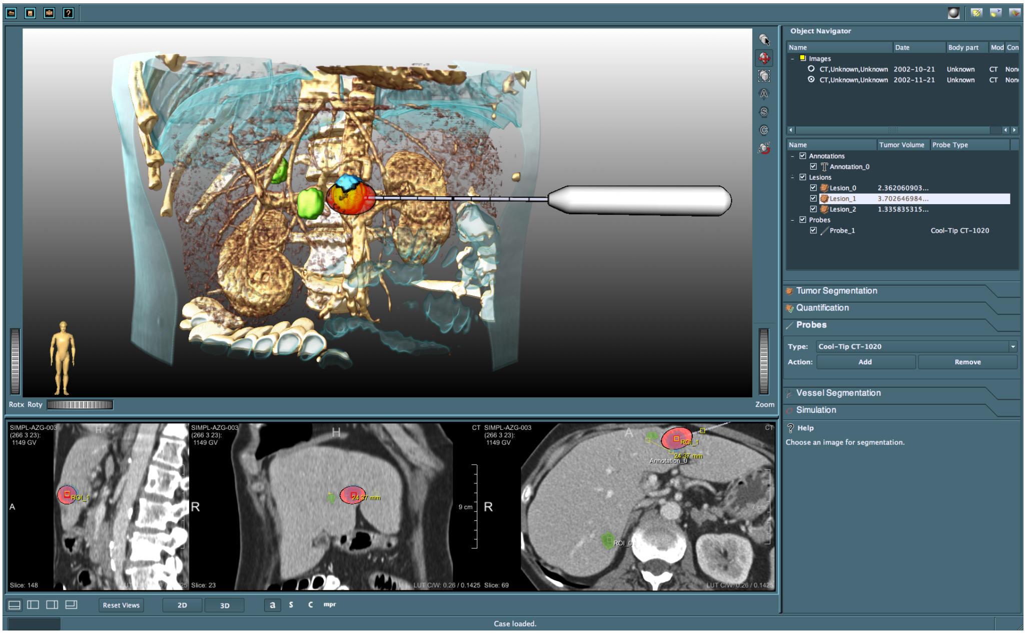
Distance from Entry: 5 mm

Figure 2.33: CT-based navigation of pedicle screw placement in vertebra L2. The preoperatively planned screw insertion is shown in red end the intraoperatively tracked screw placement instrument is shown in green (Langlotz and Nolte, 2004)

### 2.5.4 HEPATIC SURGERY

The most common liver diseases are primary hepatic carcinoma, hepatitis C, and liver cirrhosis. Depending on the characteristics and the severity of the disease, surgical treatment might be necessary.

Common strategies aim at removing diseased parts of the liver or at replacing them with a graft from a living donor or from donated specimens of the liver. Non-resectable tumors can be treated with *Radio Frequency Ablation* (Mundeleer et al., 2008; Li et al., 2009; Nicolau et al., 2005, 2009) or *Interstitial Laser Coagulation* (Heisterkamp et al., 1999; Hashimoto et al., 1991). These techniques are performed with *Minimally Invasive Surgery*, in which a small probe is inserted in the tumor and subsequently heated with either radio frequency or optical energy, as shown in Figure 2.34.



Object Navigator

| Name | Date | Body part | Mod | Con |
|---|---|---|---|---|
| – Images | | | | |
| ○ CT,Unknown,Unknown | 2002-10-21 | Unknown | CT | Non |
| ● CT,Unknown,Unknown | 2002-11-21 | Unknown | CT | Non |

| Name | Tumor Volume | Probe Type |
|---|---|---|
| – ☑ Annotations | | |
| ☑ Annotation_0 | | |
| – ☑ Lesions | | |
| ☑ Lesion_0 | 2.362060903... | |
| ☑ Lesion_1 | 3.702646984... | |
| ☑ Lesion_2 | 1.335835315... | |
| – ☑ Probes | | |
| ☑ Probe_1 | | Cool-Tip CT-1020 |

Type: Cool-Tip CT-1020

Action: Add Remove

Help

Choose an image for segmentation.

Case loaded.

**Figure 2.34:** Graphical user interface for radio frequency ablation surgery planning. *Top left:* Volume rendering combined with surface renderings of the tumors, visual representation of the RFA applicator and a visual representation of the area affected by the RFA procedure. *Bottom left:* Standard anatomical views extended with the RFA planning (Rieder et al., 2009)

Traditional liver surgery poses a number of challenges, in particular resection of parts of the liver. Liver tissue resection has a profound impact on liver blood flow. For this reason, radiologists and surgeons mainly work with a scheme from Couinaud that defines vascular territories. These territories mark regions of vascular structures that can be resected relatively safely without harming the blood flow of other territories. However, due to the varying shape and size of the liver from patient to patient, this scheme is not patient-specific.

3D imaging allows the surgeon to better understand the patient-specific liver anatomy and pathology. The computer-assisted extraction of pathological tissue and vessels allows the surgeon to perform measurements, study spatial relations and to investigate tumor infiltration.

#### 2.5.4.1 TASKS

As with all other applications, studying the patient-specific anatomy and pathology (§ 2.3.1.1) is the first task in the hepatic surgery pipeline. In this task, the embedding of the tumor in the healthy tissue is studied so that a resection plan (§ 2.2.4) can be devised and subsequently also an access planning (§ 2.2.3), in which structures at risk are avoided. The task of visualization in this context is to accurately and intuitively communicate the surgical plan so that at-risk structures can be avoided.

#### 2.5.4.2 VISUALIZATION

The primary objective of computer-assisted hepatic surgery planning (Bourquain et al., 2002; Shevchenko et al., 2010) is to provide an interactive environment in which the anatomical (liver and vessels), pathological (tumor(s) and necrotic tissue) and functional data (blood flow and vascular territories) can be displayed and interacted with in order to devise a patient-specific surgical plan.

Most visualizations for hepatic surgery planning aim at providing a complete image of the therapeutic object, including all intricate details. Figure 2.35 shows the first attempt at simplifying the visual depiction of the resection strategy in hepatic surgery. More recent research shows that it is useful to visually superimpose the risks of resection, on top of a simplified visualization of the liver(Hansen et al., 2008b), in an attempt to reduce the visual overhead (reconstructed liver + vascular structures) which can result in cognitive overload.

Liver surgery planning systems allow the surgeon to virtual perform a resection (Glombitza et al., 1999; Numminen et al., 2005; Feuerstein et al., 2008; Yamanaka et al., 2007; Chen et al., 2010; Reitinger et al., 2006; Sorantin et al., 2008). This method is non-invasive; the surgeon can evaluate as many resection approaches as possible, ensuring that all hazardous tissue is removed and as little healthy tissue as possible. The simplest form of resection plan is with planar resection geometry. More advanced methods include deformable cutting planes (Konrad-Verse et al., 2004; Hansen et al., 2008a), as shown in Figure 2.36, and cutting geometry defined by implicit functions (Preim et al., 2000) e.g., torus, cylinder and sphere.

The planning stage is also enhanced by incorporating soft tissue dynamics and interactive cutting, possibly in concert with haptic feedback. More recently, the work of (Courtecuisse et al., 2010) describes a generic simulator for soft tissue dynamics. Deformations are physically based and contact forces are articulated through force feedback. Their system also facilitates interactive cutting. Three systems exist for interactive liver surgery planning: *HepaVision2* (MeVis GmbH, Bremen), *LiverLive* (Navidez Ltd, Slovenia) and *OrgaNicer* (German Cancer Research Center, Heidelberg).

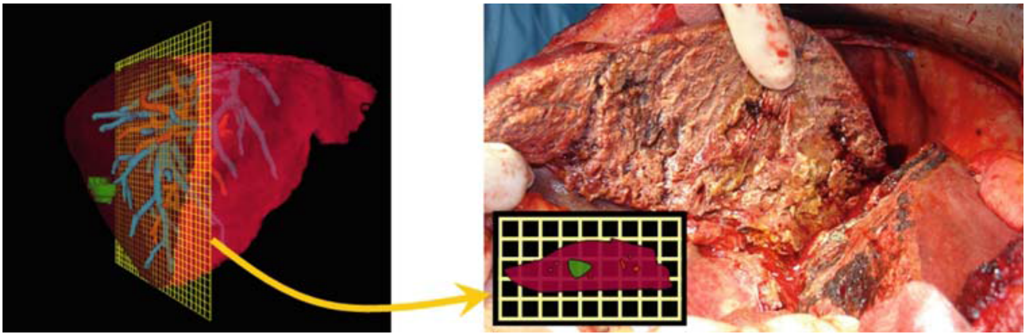


**Figure 2.35:** The resection map is used to transfer the planning to the operating theater using documentation-based guidance. *Left:* Preoperative planning of the resection plane. *Right:* A simplified 2D image of the resection helps the surgeon to perform the actual resection (Lamata et al., 2008)

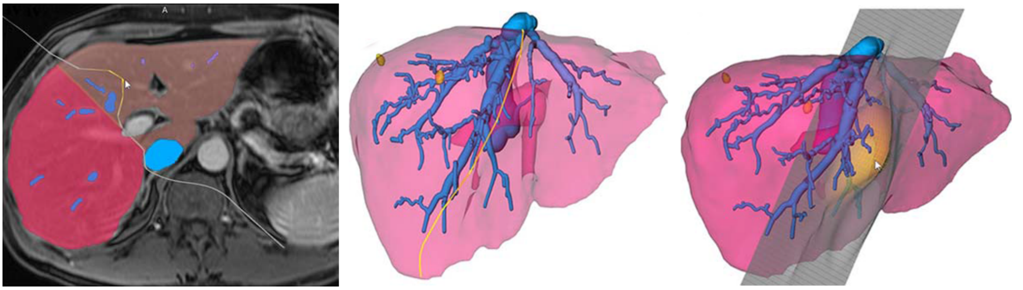


**Figure 2.36:** Deformable resection plane for hepatic surgery. *Left:* Planning by drawing into slices. *Middle:* Planning by drawing lines on the surface of the liver. *Right:* An interactive deformable cutting plane as a result of the two former planning methods (Hansen et al., 2008a)

### 2.5.4.3 GUIDANCE

Image-guided surgery techniques (§ 2.4.3) can be used for the guidance of hepatic surgery. By matching vessel centerlines extracted from *intraoperative ultrasound* to the preoperative vessel centerlines and taking into account deformation, and then combining this with optical instrument tracking, surgical actions can be shown within the context of the high resolution preoperative data and surgical plan (Lange et al., 2004; Dumpuri et al., 2010; Lee et al., 2010; Dagon et al., 2009). Another option is the use of laser range scanning technology to match the liver surface (Cash et al., 2007). Experimental augmented reality systems have been developed to provide guide minimally invasive (Feuerstein et al., 2008; Hansen et al., 2010), and open hepatic surgery (Hansen et al., 2010).

The resection plan can also be transferred to the operating theater using a 2D Resection Map (Lamata et al., 2008) or a simplified Risk Map (Hansen et al., 2010) for use as documentation-based guidance (§ 2.4.2) during surgery. In a similar vein, Tumor Maps have been proposed for the postoperative evaluation of tumor ablation results (Rieder et al., 2008).

Computer-assisted hepatic surgery faces similar technical challenges as computer-assisted neurosurgery (§ 2.5.2) where changes to the structure of the tissue of the brain (brain-shift) during a procedure seriously compromise the guidance accuracy of the surgical plan (which is based on preoperative images). Whereas the the skull is rigidly fixed in neurosurgery, the liver is part of a complex of other soft tissues, and also subject to the breathing motion of the patient. Compensating for the former with registration is still considered a very technically challenging problem, that hampers the use of computer navigation in clinical routine.

## 2.6 CONCLUSIONS

In this chapter, we analyzed the CAS literature where visualization plays a prominent role. Our discussion started with a description of CAS, and its surgical pipeline. We then analyzed and discussed relevant literature from a task-based perspective. Next, we discussed how medical data is visually represented in CAS, how medical professionals interact with the data, and how simulation is performed. Further, we studied how surgical plans are transferred to the operating theater. Finally, we identify and discuss four major application areas of CAS and its relevant visualization literature.

Our study shows, that visualization plays a key role in all aspects of the CAS pipeline. From accurate and faithful depiction of patient anatomy to the visual representation and interaction with simulation results. From this, it follows that without (medical) visualization research, CAS would not have matured as much as it has today.

Despite its universally recognized potential, the wide spread adoption in clinical routine still not a fact. Most CAS systems are costly and need to be maintained. Further, they require training and their use requires an adjustment to the standard surgical workflow e.g., fiducial markers, additional sterilization, and extra hardware in the operating theater. Given these factors, it is not easy to quantify the merits of CAS. For example, in orthopaedic surgery, surgeons are typically content with outcome of joint replacement surgery without computer assistance (Jaramaz et al., 2006). A new joint replacement CAS system should therefore offer significant benefits over the conventional way in order to legitimize the additional operating time and costs.

In this chapter, we briefly described the concept of customizable surgical guides, as shown in Section 2.4.4.3. Even though this concept seems promising, only a handful of publications were found. We estimate that this is due to a lack of proper planning tools. In the next section we will describe a planning system that is developed for generic CSGs.

# Customizable Surgical Guide Optimization[1]

In Section 2.4.4.3, we discussed how the planned alignment of joint prosthesis components can be transferred to the operating theater using a customizable surgical guide (CSG). The CSG is a mechanically adjustable device that, when configured correctly, provides a unique, stable and accurate connection (the fit) with a host bone. It incorporates one or more mechanical constraints such as drilling, sawing and milling, so that planned surgical actions can be reproduced intraoperatively.

The concept of a CSG is easily illustrated with a pinscreen toy[2], as shown in Figure 3.1. This toy encapsulates an impressed shape using an array of pins that can slide along their main axis. This is merely an example of what a CSG *could* look like, of course many alternative designs are possible.

The CSG relies on sparse (point) contact to achieve a fit with bone structures. For CSGs to be useful in a in a clinical context, manual configuration should take place in no more than five minutes. This implies that the number of contacts should be minimized, but of course not at the expense of the fit.

As the number of contacts decreases, the distribution of contact across the surface of the bone becomes increasingly more important. It turns out that manual CSG configuration leads to very poor and non-reproducible results. therefore, we propose to integrate an automatic CSG optimization algorithm in the planning of CSG-based surgery.

In this chapter, we describe a numerical optimization method for the automatic CSG configuration given an arbitrary bone reconstruction. In order to achieve this,

[1] T. Kroes, E. R. Valstar and E.Eisemann. Published in the International Journal of Computer Assisted Radiology and Surgery

[2] http://www.pinscreens.net/

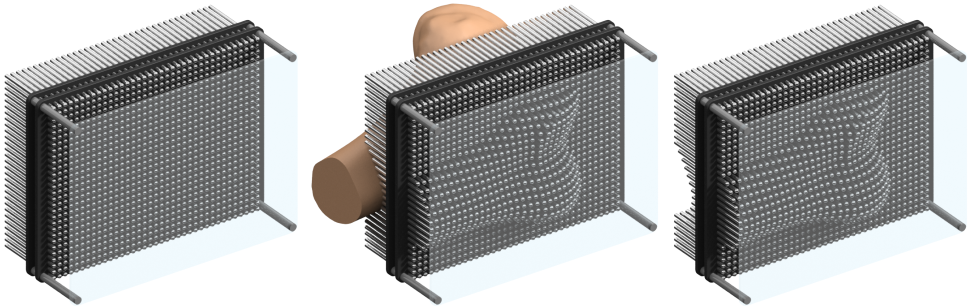


Figure 3.1: The CSG concept illustrated with a pinscreen toy and a model of the distal femur. *Left:* in its default state, all pins of the pinscreen are retracted. *Middle:* a femur model is impressed in the toy. *Right:* the femur model is removed and the shape of the femur is encapsulated in the pinscreen.

we introduce a method to quantify the fit of a CSG on the bone and use this metric to optimize the CSG configuration with a genetic optimization algorithm. We validate our framework in the context of knee arthroplasty, using a variety of knee models and a pin-based CSG design. We show that, given arbitrary CSG design and bone model, our optimization algorithm is able to produce CSG configurations with a good fit.

## 3.1 ABSTRACT

Computer-assisted orthopaedic surgery (CAOS) aims at minimizing invasiveness, postoperative pain, and morbidity with computer-assisted preoperative planning and intraoperative guidance techniques, of which camera-based navigation and patient-specific Templates (PST) are the most common. PSTs are one-time templates that guide the surgeon initially in cutting slits or drilling holes. This method can be extended to reusable and customizable surgical guides (CSG), which can be adapted to the patients bone. Determining the right set of CSG input parameters by hand is a challenging task, given the vast amount of input parameter combinations and the complex physical interaction between the PST/CSG and the bone.

This chapter introduces a novel algorithm to solve the problem of choosing the right set of input parameters. Our approach predicts how well a CSG instance is able to reproduce the planned alignment based on a physical simulation and uses a genetic optimization algorithm to determine optimal configurations. We validate our technique with a prototype of a pin-based CSG and nine rapid prototyped distal femora.

The proposed optimization technique has been compared to manual optimization by experts, as well as participants with domain experience. Using the optimization technique, the alignment errors remained within practical boundaries of $1.2$ mm translation and $0.9°$ rotation error. In all cases, the proposed method outperformed manual optimization.

Manually optimizing CSG parameters turns out to be a counterintuitive task. Even after training, subjects with and without anatomical background fail in choosing appropriate CSG configurations. Our optimization algorithm ensures that the CSG is configured correctly, and we could demonstrate that the intended alignment of the CSG is accurately reproduced on all tested bone geometries.

## 3.2 INTRODUCTION

Osteoarthritis and rheumatoid arthritis lead to irreversible damage to joints. These conditions impact the patients mobility and lead to severe pain. An orthopaedic surgeon can replace the joint in order to reduce these symptoms. During joint replacement surgery, the shape of the bone has to be altered (by sawing and drilling) in such a way that the orthopaedic implant can be securely installed into the planned position and orientation. There are many factors, such as blood loss, aseptic loosening, and operating time, which can have a negative impact on the patient's treatment. Among these factors is mal-alignment, which has an important effect on the stability of the implant and in some cases also the functioning of the joint, e.g., range of motion (Bäthis et al., 2004; Lotke and Ecker, 1977). In this work, we will focus on this particular aspect. With the conventional array of surgical instruments, implant alignment is a challenging task, because anatomical references, used for implant alignment, are obscured by layers of tissue, such as muscles and fat.

Alignment accuracy can be improved using CAOS systems that provide planning routines and active/passive guidance during joint replacement procedures (Jaramaz et al., 2006; Kowal et al., 2007; Matziolis et al., 2007; Sparmann et al., 2003; Stöckl et al., 2004).

However, most CAOS systems tend to increase operating time and add complexity to the surgical procedure. They have a steep learning curve, and the accuracy depends on the quality of the input information, e.g., reconstructed bone, quality of marker tracking, and registration (Sikorski and Chauhan, 2003). At the same time, this type of navigation requires auxiliary hardware, which needs to be sterilized. Furthermore, a recent meta-study shows that the increased accuracy of implant alignment does not lead to improved postoperative function recovery (Xie et al., 2012).

On the contrary, PSTs are surgical guides that fit uniquely on a patient and are manufactured using rapid prototyping technology. They encode the complete planning in the template and provide guidance during pedicle screw insertion (Goffin et al., 2001; Van Cleynenbreugel et al., 2002; Singare et al., 2009; Owen et al., 2007), knee (Hafez et al., 2007; Radermacher et al., 1998; Rademacher, 1999), hip (Raaijmaakers et al., 2010; Audenaert et al., 2011) and shoulder replacement (Metzger and Vanasse, 2010). No specialized auxiliary hardware is needed for navigation, but a surgeon can also not make adjustments to the planning during the procedure. This aspect can be problematic should the template not fit correctly due to manufacturing issues, poor handling of the template and/or poor 3D reconstruction of the bone, on which the template planning is based. These templates can only be used once, after which they are disposed of.

CSGs attempt to mitigate the problems associated with existing CAOS approaches, such as mal-alignment (Klatt et al., 2008). The CSG is a mechanically adjustable surgical instrument that fits uniquely onto a patient and is designed to provide guidance (e.g., holes for drilling and slits for cutting) to the surgeon during surgery. In contrast to PSTs, this type of device is reusable, but needs to rely on a manual configuration step. While this feature even makes it possible to apply changes to the surgical plan during an actual procedure, it is often a complex task and very difficult to perform correctly by hand.

The objective of this chapter is to investigate how to automate the CSG configuration process for an arbitrary CSG. We illustrate our method by applying it to knee replacement surgery. Via a semiautomated planning step, the CSG becomes patient-specific and ensures that the planned alignment can be accurately reproduced and the device snaps into the intended position and orientation, as shown in Figure 3.2. To this extent, we created a novel and generic computer-assisted planning method that predicts the CSG trajectory to the bone and its stability and guides the configuration process. The method is designed to allow users to indicate particular regions on the bone to be avoided (for instance due to bone spurs). We validate our optimization method via a simulation, as well as a real-world setting with a pin-based CSG applied to a rapid prototyped bone model.

The remainder of this chapter is structured as follows: after the discussion of related work, we briefly describe the involved material and the exemplary CSG design used throughout this chapter. We then present our algorithmic solution to configure the device for a specific patient. Finally, we present the results of our approach using rapid prototyping and illustrate its usefulness in the context of joint replacement surgery, before concluding.

## 3.3 RELATED WORK

In the field of CAOS, only few CSG-oriented publications exist, which we mention here. A CSG for hip replacement surgery has been introduced in (Steppacher et al., 2011) in order to improve acetabular cup positioning. A novel method to transfer a computer-assisted knee replacement surgery to the patient, using an adjustable pin-grid-based jig, is described in (Haselbacher et al., 2012). The results of a pilot study conducted on the distal femur show a relatively high axial translation error, which might relate to the fact that the pin configurations were manually configured, an issue we address in this work.

In (Weinstein, 2006) a drill guide for dental implants is described. In this approach, a set of actuated pins is used to register the instrument to the bone and to reproduce the planned implant direction. In contrast to our method, pins with sensors are used to obtain a shape-based registration (Simon et al., 1995; Simon and Kanade, 1997; Simon and Lavallée, 1998), whereas in our solution the pin position and layout are fixed and the insertion depth determined a priori. In (Yaniv and Joskowicz, 2005) a robot-assisted drill guide is described that uses a special registration process that allows surgeons to drill holes along a predefined axis. Additionally, an analytical method for calculating the docking robustness of PSTs in 2D has been developed (Mattheijer et al., 2013). Another method analyzes patient CT scans and identifies bone surface regions where the contact adds the most to stability (Radermacher et al., 2000). This input could be integrated in our approach.

## 3.4 MATERIALS AND METHODS

### 3.4.1 PIN-BASED CSG

The pin-based CSG studied in this chapter is inspired by (Haselbacher et al., 2012) and uses a sparse point-surface contact set (a selected number of strategically placed adjustable pins) to achieve a stable configuration between the device and the bone. The pin-based CSG consists of a regular grid of holes, through which pins can be inserted, as shown in Figure 3.2. To give a more precise impression of how our method should be integrated in the surgical pipeline, we give an overview for a CSG-assisted total knee replacement in Figure 3.3.

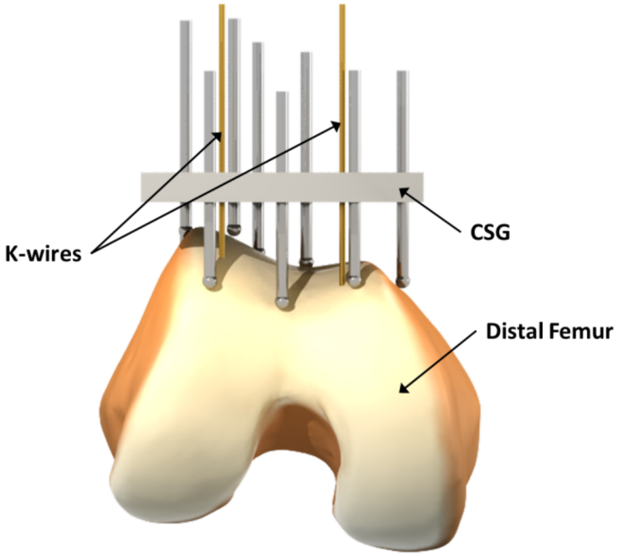


Figure 3.2: The Pin-based version of the CSG applied to the distal femur. The surgical plan is transferred to the operating theatre by encapsulating the shape of the bone in the guide using a collection of strategically distributed pins (which collide with the surface of the bone). The CSG has predefined holes for the k-wires that are compatible with standard instrumentation for performing the principal bone cut.

Since the CSG works on the basis of sparse point-surface contact, it is of paramount importance to configure the device appropriately. First, the pin depth should be adjusted in such a way that all pins touch the bone surface when the device is in its intended position. Further, their number should stay reasonable for a clinical setting, which implies that they need to be strategically distributed. Given these constraints, our algorithm derives an optimal set of active pins $csg$, a CSG configuration, via a simulation and optimization procedure, as shown in Figure 3.2.

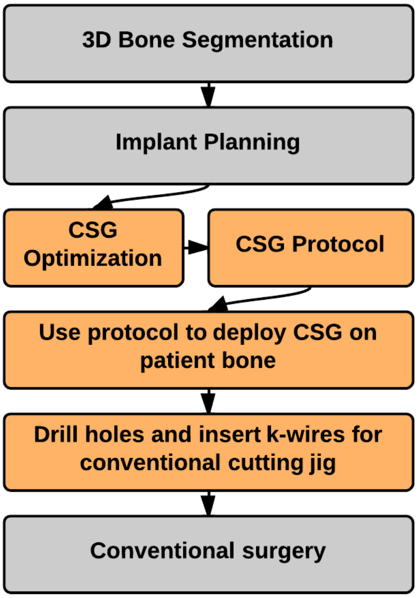

3D Bone Segmentation
Implant Planning
CSG Optimization
CSG Protocol
Use protocol to deploy CSG on patient bone
Drill holes and insert k-wires for conventional cutting jig
Conventional surgery

**Figure 3.3:** CSG pipeline for knee replacement surgery. The orange steps are specific to the CSG. The CSG takes as input the planned implant alignment and uses it to make the CSG patient-specific and to optimize its configuration. When the CSG is optimized, its configuration protocol is used intraoperatively to adjust the CSG and to dock it on the patients bone. Next, holes are drilled for k-wires. Once the k-wires are inserted, a cutting block can be attached to the k-wires and conventional surgery can take over.

In our pilot study, we fabricated a prototype of the pin-based CSG, which consists of a square plate ($width$=90 mm, $height$=10 mm, $depth$=90 mm) with 11 x 11 holes ($radius$=2 mm), through which pins ($length$=100 mm, $radius_{tip}$=2.5 mm) can be inserted and fixated, as shown in Figure 3.2. The prototype of the pin-based CSG merely serves as a tool to validate our optimization method and is not directly intended for clinical use. The pin-based CSG can contain 121 pins in total; however, it seems impractical to adjust all pins. Setting a single pin takes at least 10s, and the manual configuration process is increasingly tedious and cumbersome with a larger number. Additionally, sometimes it is important to avoid placing pins, which would lead to unwanted contacts with certain regions of the bone, e.g., those designated inaccessible by the surgeon, as shown in Figure 3.4. We will refer to these situations as full, respectively, limited exposure.

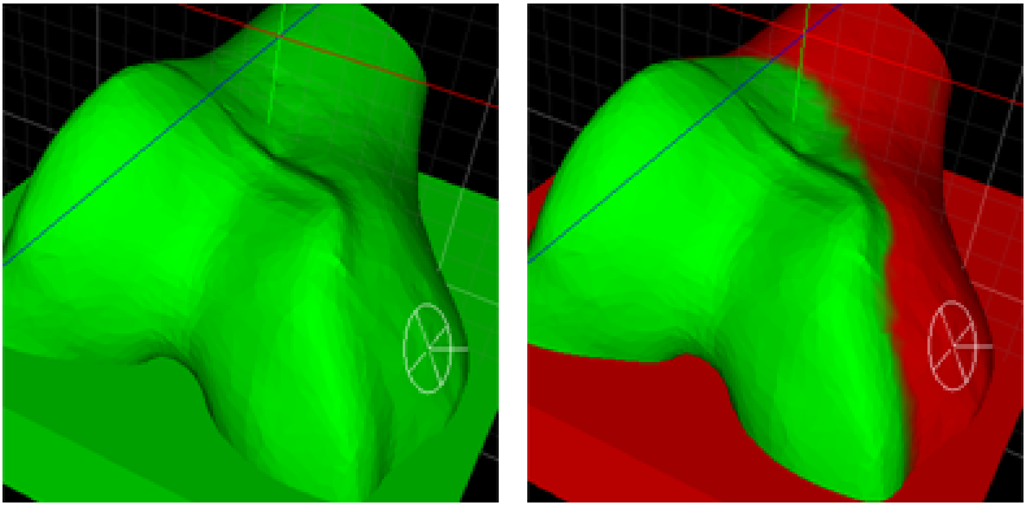


Figure 3.4: *Left:* Full exposure, pins can be deployed anywhere on the bone/cartilage. *Right:* Limited exposure. The orthopaedic surgeon paints the areas on the bone that are deemed accessible during surgery, thus limiting where pins can be deployed.

Given the limit on the number of pins, the amount of possible distributions, and the complexity of the physical interactions between the CSG and the bone, it is challenging to configure the CSG to ensure a very low rotational and translational error after application to the bone. Using our algorithm, this configuration step can be automated, leading to a small number of strategically positioned pins, ensuring stability and accuracy.

### 3.4.2 CSG OPTIMIZATION

The core of our optimization method is the derivation of the CSG configuration, which we will describe in detail in this section. First, we define the CSG objective function to measure the device's deviation from its intended location, while considering an uncertainty in the CSG placement process. We then explain how this objective function is minimized with the help of a genetic algorithm in order to optimize the configuration of the CSG. For convenience, Table 3.1 contains an overview of all the variables used in this section.

#### 3.4.2.1 CSG OBJECTIVE FUNCTION

In order to steer the optimization method toward a suitable CSG configuration, an appropriate CSG objective function is key. Ultimately, it should be an indicator of

| | |
|---|---|
| $s$ | Bone surface |
| $M_d$ | Maximum deviation over all pins in a CSG |
| $T$ | pin-number optimization threshold applied to $M_d$ |
| $n_{csg}$ | CSG number of pins |
| $n_{max}$ | CSG maximum number of pins |
| $d$ | Docking movement (origin and direction) |
| $E_{csg}$ | CSG error |
| $E_{d,csg}$ | CSG alignment error after docking |
| $n$ | Population size |
| $c_e$ | Elite CSG percentage |
| $c_c$ | Crossover probability |
| $c_m$ | Mutation probability |
| $c_n$ | New random CSG probability |
| $i$ | Stop if not improving after $i$ iterations |

Table 3.1: Optimization variables

how well the device snaps into its intended position and how stable it is. Hence, the baseline of our objective function is a measurement of the alignment error (global drift and orientation deviation) when the device reached an equilibrium state on the bone. Nonetheless, as angles and translational movement are not comparable, we opt for an objective function which allows us to bound both.

Assuming for the moment only a single direction-origin pair $d$ defining a translational movement toward the bone surface, we then define for a given CSG configuration ($csg$) the CSG error for $E_{d,csg}$ as the maximum deviation $M_d$ over all pins. In other words, we compute $M_d$, as the maximum Euclidean distance between the intended and actual pin location, as shown in Figure 3.5. Given $M_d$ we can derive a bound on global drift and orientation deviation and vice versa. In our work, we impose a maximally acceptable drift of 1.5 mm, which implies a rotational error of less than $1^\circ$, as shown in Figure 3.16 and Figure 3.17. The surgeon can also modify this value prior to surgery.

One important observation is that the equilibrium state of the CSG depends on the bone surface $s$ and a surgeon would not be able to move a device perfectly along a single direction. Consequently, several docking movements $d$, in form of a starting position and direction, should be tested. In practice, we restrict $d$ to a truncated cone, as shown in Figure 3.6. The directions and origins inside the truncated cone are tested; the final objective function is then $E_{csg} = max_{d \in cone} E_{d,csg}$. In practice, we use 40 directions because the maximum drift parameter changed only marginally (drift $<$ 0.05 mm) hereafter and the computational overhead of adding more directions does not pay off in this case.

To determine the CSG equilibrium state, we employ a physical simulation that predicts how the device will behave. During the simulation, we subject the CSG to exter-

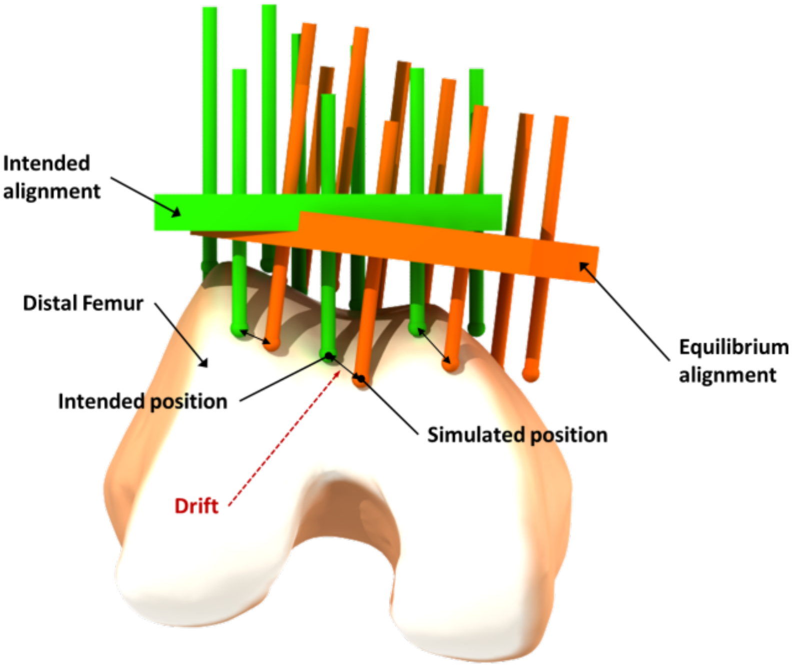


Figure 3.5: The drift value for a single pin is defined as the Euclidean distance between the intended pin position and the simulated pin position at equilibrium.

nal forces to mimic the real behavior of the docking process. We observed that apart from the principal pressing force along $d$, the user will exert moments and parallel forces on the CSG in an attempt to assess its stability using the haptic feedback that it provides (if the CSG wanders under these external forces, it is not securely docked in the right position). Taking the pressing force into account is useful because the morphology of the host bone might make particular pressing directions more suitable. For instance, in the case of the distal femur, we observed that when applying the pressing force under a slight angle, the CSG behavior improves, as shown in Figure 3.7.

In most cases, the CSG will reach an equilibrium state in which the CSG error can be determined. However, in some cases, the CSG will simply fall off and the physical simulation will be aborted prematurely. Here, we consider the error to be infinite, indicating that it is not useful.

The objective function also allows us to take several constraints into account. First, the truncated cone defining possible values for $d$ can be manually modified by the surgeon to adjust the angle of approach and the cone angle. For instance, a right-handed surgeon might never place the device from the left, due to limited exposure or the way that the patella is exposed. Such adjustments can be performed via a simple interface showing the virtual bone and the cone. The cone angle range is limited to 10° to 30° in order to prevent unreasonable docking directions (e.g., from below the surface). In practice, these constraints are easily fulfilled. Additionally, we provide standard settings to add an automatic bias of a 5° inclination for left/right-handedness of the surgeons, but refrained from using it in our study to avoid such prior knowledge.

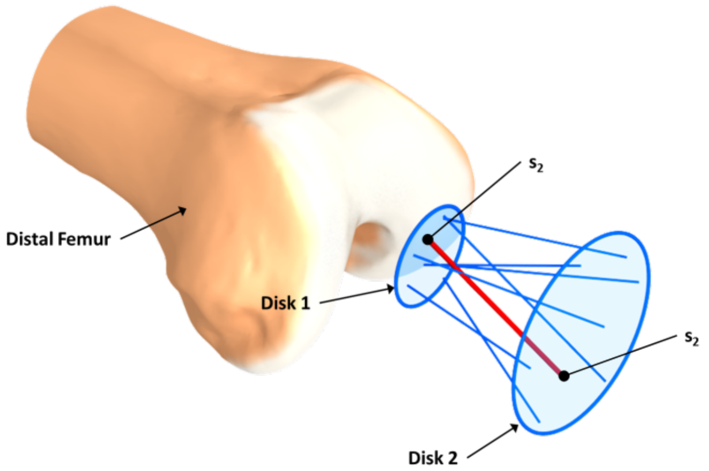


**Figure 3.6:** Docking directions are generated inside a truncated cone by picking a random point on disk one and two, these two points ($s1$ and $s2$) are then connected and form the docking direction $d$. The default radius for Disk 2 is 5 mm and the cone angle is 15°. The cone angle represents the placement uncertainty, and does not dependent on the size of the patient. However, this parameter can be changed by the user prior to optimization.

#### 3.4.2.2 GENETIC OPTIMIZATION

Given the sparse contact between the CSG and the bone, it is critical that the CSG configuration is tailored in such a way that it optimizes the fit and warrants stability and accuracy. For a given patient, we rely on an algorithm that uses a genetic optimization method driven by the previously defined objective function, which will be explained in this section. This solution allows us to handle the very large input parameter space (with around $2^{11\times 11}$ possible pin configurations), in which, given the current software and hardware resources, it would be impossible to evaluate all configurations iteratively. Although we refer to our pin-based CSG, most of this optimization strategy can, with minor modifications, be applied to other types of CSGs as well e.g., for hip replacement.

Genetic algorithms are inspired by natural evolution, in which fit individuals are more likely to survive (Holland, 1975). Unfit individuals are removed by a selection process. The remaining population develops into new individuals via inheritance,

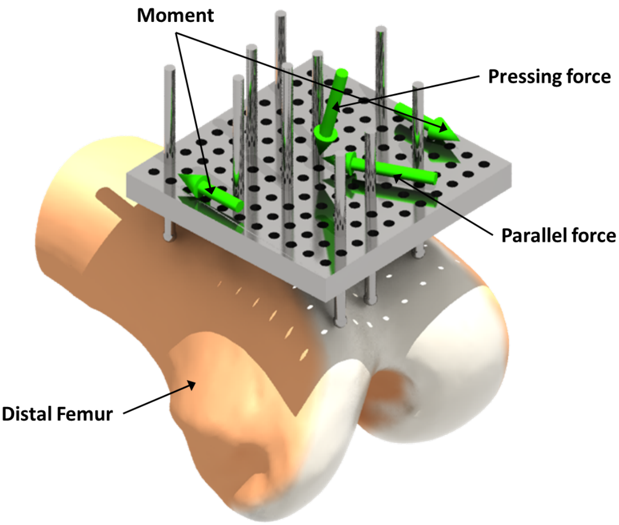


Figure 3.7: Visualization of the CSG and the external forces applied during the physical simulation. The moment magnitude varies periodically with a sine function, the parallel force rotates around the center of the CSG.

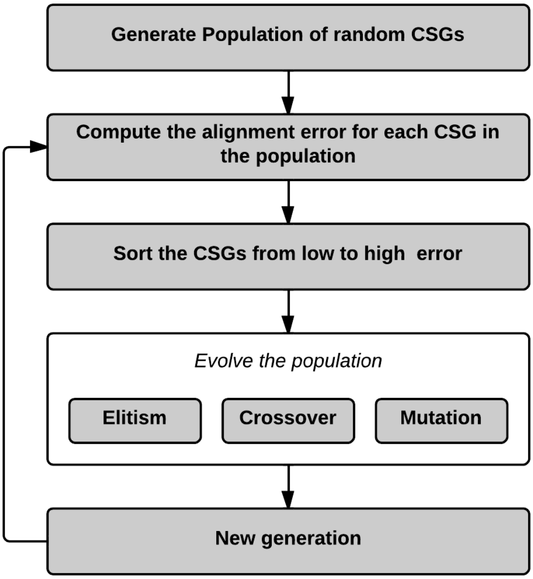

Generate Population of random CSGs
Compute the alignment error for each CSG in the population
Sort the CSGs from low to high error
Evolve the population
Elitism
Crossover
Mutation
New generation

Figure 3.8: Overview of the genetic algorithm used in the CSG optimization algorithm.

crossover, and mutation. By iterating the selection and evolution steps, the individuals are likely to approach the local minima of the objective function. Figure 3.8 gives a schematic overview of the genetic optimization algorithm used in our approach.

In our context, individuals correspond to different CSG configurations, as shown in Figure 3.3. In our case, each configuration $csg$ consists of a set of active pins in the CSG, as shown in Figure 3.9. Initially, the CSG population consists of random active-pin distributions, which are established via a Poisson distribution to ensure a minimum distance between the pins and to avoid clumping, which leads to individuals with high alignment error that are unlikely to survive the genetic optimization. The pin insertion depth is determined automatically by moving the pins downward from the intended rest pose of the CSG until they collide with the surface of the bone.

Initially, our set consists of ten pins, which is a reasonable number to be configured manually. Introducing additional pins seems overly conservative, as in all test cases, ten led to solutions that respected the imposed accuracy constraints for practical use ($< 1.5$ mm and $< 1°$). In fact, our algorithm always found solutions with even less pins while maintaining stability and accuracy.

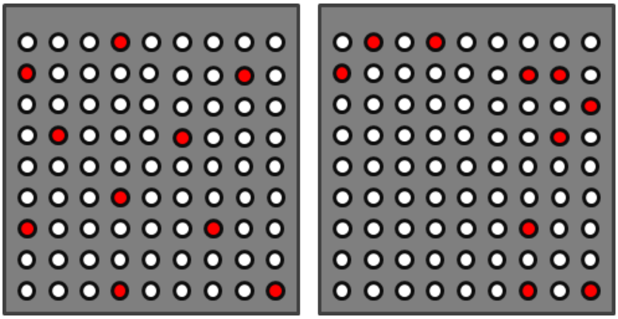


Figure 3.9: *Left:* Example of a CSG pin configuration using a Poisson distribution. *Right:* Pin distribution as a result of random sampling, which leads to clumping of pins (exaggerated case). Although this pin distribution might work, there is a high probability that it will have a high alignment error, since there are no pins in the lower left corner.

To evolve the set of current individuals, we apply elitism, crossover, and mutation. Elitism keeps the best individuals (elite) in the population to maintain their good properties. For crossovers, properties of two randomly chosen individuals are exchanged, as shown in Figure 3.10. Mutation means copying elite individuals and applying a slight configuration change. Precisely, a randomly chosen active pin is moved to a new location, as shown in Figure 3.11. In order to reduce the probability of getting stuck in a local extremum, random CSGs are added to the population with a small probability.

Finally, we introduce a special mutation step with the goal of converging toward a minimal pin set; if an individuals error falls below a threshold $T$, as shown in (§ 3.4.2.1),

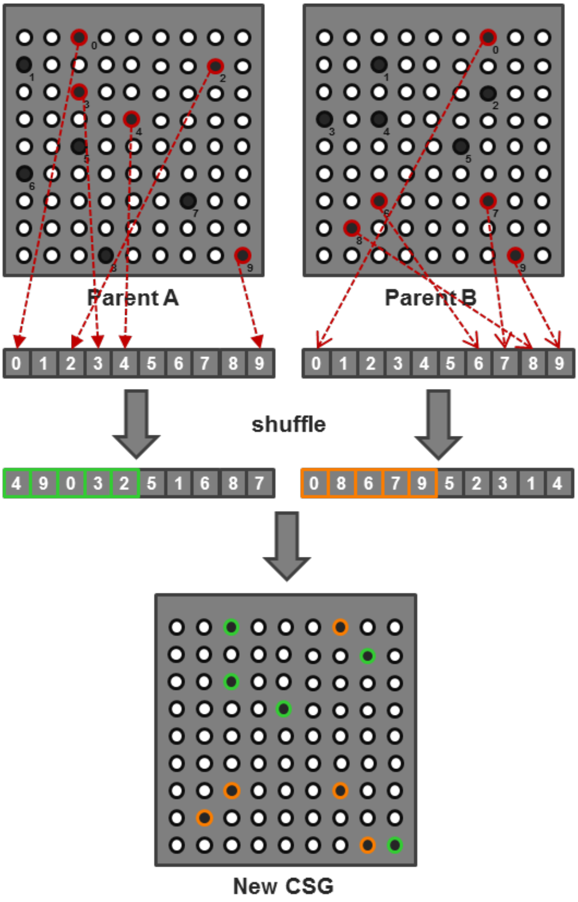


**Figure 3.10:** In the crossover stage, the configuration of two random CSGs (parent A and B) are combined to spawn a new CSG. A new CSG is formed by combining the pin ids from two shuffled pin id lists.

a second copy with one randomly removed pin is added to the population, as shown in Figure 3.11.

In theory, in an ideal case, only six pins might remain, which is the required number for a static equilibrium (Mattheijer et al., 2013). However, finding such a perfect configuration is particularly challenging and might not even be possible for all bone morphologies. It turns out that in practice, a minimum of eight well-distributed pins is required to reach a stable device placement, as shown in Table 3.3. The outline of our optimization strategy reads as follows:

1. Generate a CSG population of $n$ individuals;

2. For each individual, evaluate the objective function § 3.4.2.1;

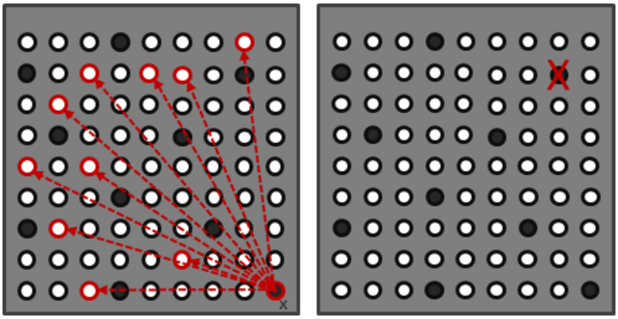


**Figure 3.11:** *Left:* Example of the single pin mutation strategy. Pin x is deactivated and an arbitrary other inactive pin is activated thereby creating a new CSG which is added to the population. *Right:* Example of pin removal mutation. A new CSG is created by making a copy of the elite individual and removing a randomly selected active pin. The newly created CSG is added to the population.

3. Sort the population based on alignment error in an ascending manner;
4. $c_e$ percentage of individuals with the lowest error are propagated to the next generation without any modification (elitism);
5. For each individual whose error is below the given accuracy threshold, we add a copy with one removed pin to the population;
6. Complete the population to $n$ by performing cross-overs, mutations, and by inserting random individuals
   a $c_c n$ CSGs are created using crossover, where the parents are chosen proportionally to their error (individuals with low error are more likely to be chosen than individuals with high error);
   b The remaining fraction of $(1-c_c)n$ CSGs are used for mutation $(c_m n)$, and new individuals $c_n n$
7. If the best solution has been the same over $i$ iterations, we stop the algorithm; if it changed, we restart at step 2.

The following parameter set works well in practice (Slight variations do not significantly impact the quality of the outcome):

### 3.4.2.3 IMPLEMENTATION AND PERFORMANCE

The optimization method described in this chapter is implemented in C++, Python[3] and OpenGL[4], using the open source Bullet Physics[5] Simulation API. Our optimization

[3] https://www.python.org/
[4] http://www.opengl.org/
[5] http://bulletphysics.org

$$\begin{array}{lcl} n & = & 50 \\ c_e & = & 6\% \\ c_c & = & 0.5 \\ c_n & = & 0.1 \\ c_m & = & 0.4 \\ i & = & 50 \\ T & = & 0.5\text{mm} \end{array}$$

framework provides a complete interface for exploring all aspects of the optimization process, meaning that the end user can see how CSGs evolve via the genetic optimization. For all CSGs, the user can inspect the configuration and interaction with the bone from different directions. The maximum pin drift associated with the directions are temporally visualized via a disk, as shown in Figure 3.12. Furthermore, the system allows the user to make small changes to the optimized CSG, in order to investigate the impact of a change on stability.

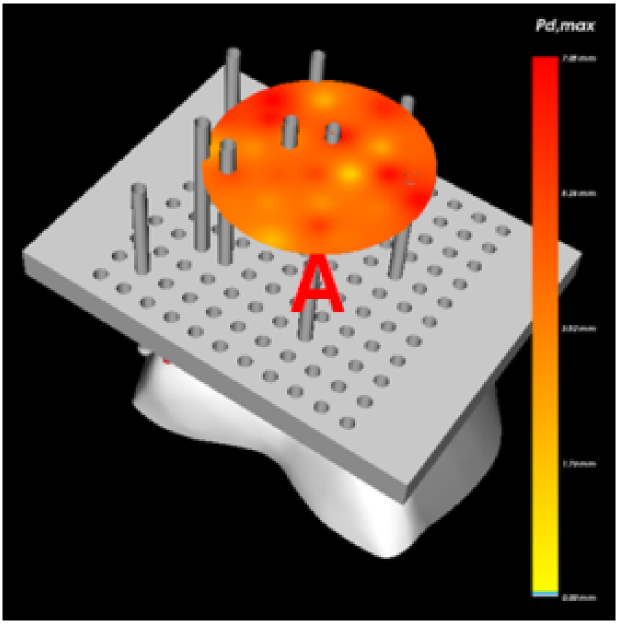


Figure 3.12: 3D viewer for inspecting the guide animation.

Table 3.2 shows the timings of the optimization routines for various CSGs applied to the distal femur model. Although the timings are considerable, there is still room for improvement, as our primary focus was the development of the optimization technique itself, and not particularly its performance. Since the physical simulation is entirely independent, it is well suited for a multi-threaded environment, resulting in a roughly linear speedup in the number of cores of the system. Further, using graph-

| Exposure | No. Generations | Time |
|---|---|---|
| Full | 49 | 1:24:43 |
| Full | 26 | 0:39:47 |
| Full | 26 | 0:22:25 |
| Limited | 26 | 0:37:04 |
| Limited | 45 | 0:48:06 |
| Limited | 16 | 0:20:30 |
| ∗Limited | 189 | 1:29:25 |
| ∗Limited | 200 | 2:56:18 |
| ∗Limited | 192 | 5:01:23 |

**Table 3.2:** List of computer-optimized CSGs and the time it took to run the genetic optimization. **The number of pins is minimized*

ics hardware for the physical simulation (e.g., for collision detection) might result in a significant speedup, as evidenced by recent graphics engines, such as Optix (Parker et al., 2010).

### 3.4.3 EXPERIMENTS

The goal of our experiments is threefold. First we want to determine the accuracy and reliability of our optimization method compared to the manual method, taking into account full and limited surgical exposure. Second, we want to see whether our optimization method works with varying bone geometries. Third, we want to verify whether our pin minimization method leads to accurate and consistent results. In the next sections, we describe our experimental setup (§ 3.4.3.1), which CSG-bone combinations were tested, (§ 3.4.3.2) and how we performed the measurements (§ 3.4.3.3).

#### 3.4.3.1 SETUP

The experimental setup as shown in Figure 3.13 comprises a prototype of the pin-based CSG, a 3D printed distal femur, and a 3D point digitizer (Microscribe [6]). The bone model is scaled 1.5 times to minimize any potential errors due to the limited resolution of the fused deposition modeling printing technology (0.17 mm) and errors in 3D point digitization. To evaluate the configuration of a CSG, we measured its precise location and orientation after placing it on the bone, as shown in Section 3.4.3.3.

#### 3.4.3.2 CSG CONFIGURATIONS

We tested the prototype of the pin-based CSG on nine 3D printed distal femora, as shown in Figure 3.14. Two are based on actual patient data, and the remaining seven are generated by an Active Shape Model (ASM), which was built from a training set of 62 distal femora as described in (Baka et al., 2014). Shapes were extracted from the

[6] `http://www.3d-microscribe.com/`

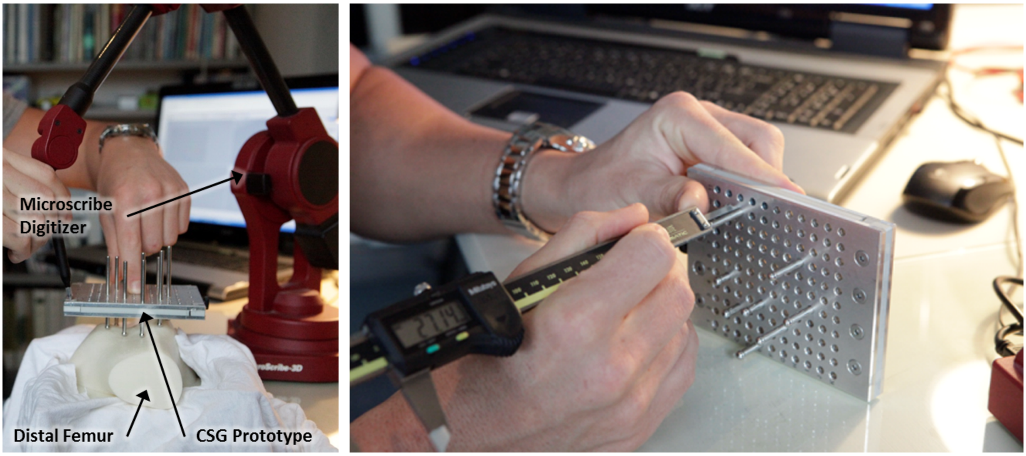


**Figure 3.13:** Photographs of the experiment setup. *Left:* The 3D print of the distal femur has been draped with a cloth to mimic a real operating scenario. After the CSG has been placed on the 3D print of the distal femur, four points on the CSG prototype are digitized using the 3D point digitizer in order to derive a transformation matrix and subsequently translational and rotational errors. *Right:* CSG prototype is manually configured using a digital caliper.

ASM by varying the first two modii of variation. We created a mean femur and six extremes of the first two modii of variation, as shown in Table 3.3.

The manually configured CSGs from Table 3.3 were generated by 9 participants (age 24-62). Participants were divided into three groups: a) three untrained participants without special a priori knowledge of human anatomy b) four medical visualization students with prior anatomical knowledge, but no surgical experience (although one even has a background as a radiology assistant), and c) two expert orthopaedic surgeons (approximately 15 and 30 years of surgical experience). The concept of the pin-based CSG was explained to the participants, stressing the importance of alignment reproducibility and stability of the CSG when docked onto the bone. They were asked to create two pin configurations (based on full and limited exposure) that would optimize the placement of the CSG in its equilibrium state (the smallest translational and rotational error with respect to the planned alignment). To facilitate this task, participants were given the option to use our computer program to set active pins using a mouse and to see the corresponding CSG device in the intended equilibrium state with all active pins in contact with the bone, as shown in Figure 3.15. The experiments were performed under no time pressure; each participant could use as much time as wanted and had as many attempts as needed to setup a configuration. Up to ten pins were allowed to be placed on the device, despite the possibility to use less, all participants used all pins. The experiments started with a quick demonstration of an ad hoc configuration and a short explanation of the simple computer program to set and investigate the pin combination. Participants took between one and three minutes to

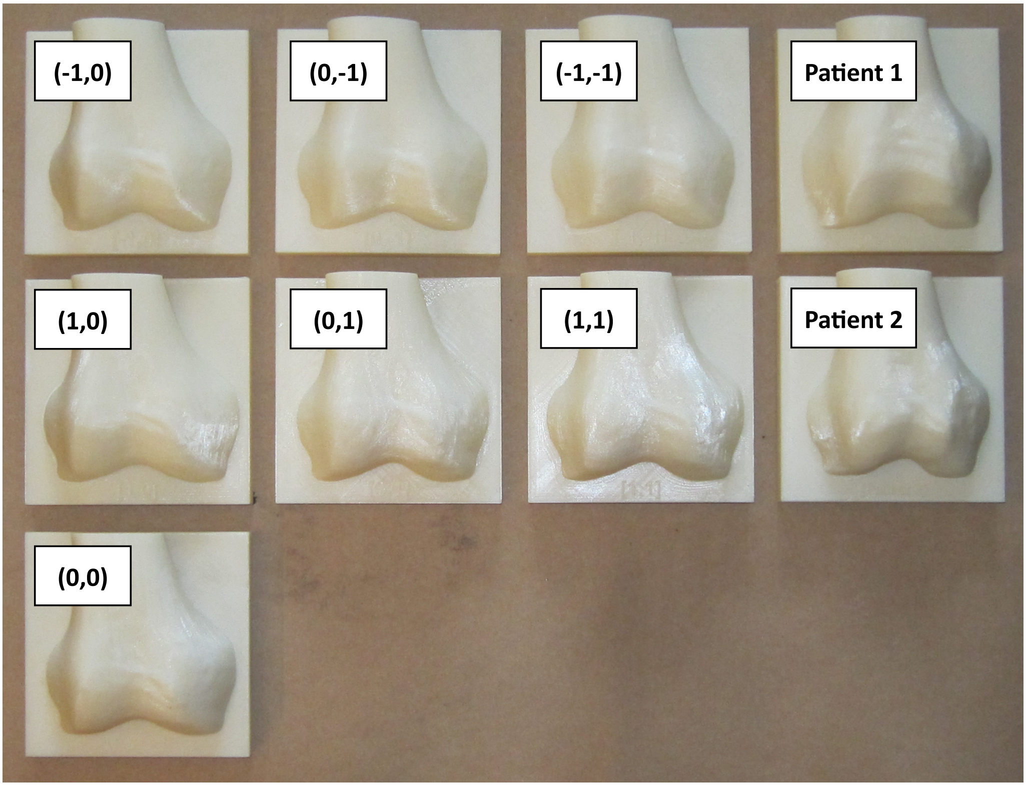


Figure 3.14: 3D printed distal femora that were used in the experiments. Seven femora are derived from an active shape model and two from actual patient data.

create a pin configuration.

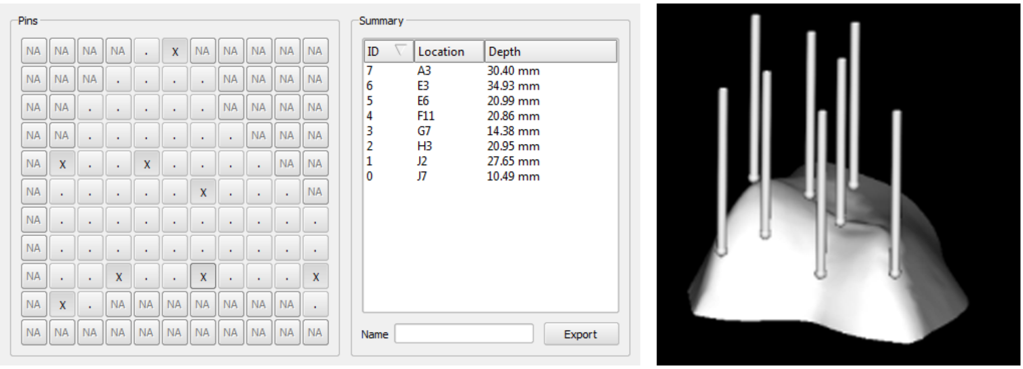
#### 3.4.3.3 MEASUREMENT METHOD

For each CSG in Table 3.3, we used a digital caliper ($\pm 0.01$ mm) to adjust the pin depth to carefully reproduce each CSG configuration, as shown in Figure 3.13. Next, it was deployed ten times on the 3D printed distal femur. In order to measure how much the CSG deviates from the planned position and orientation, two point-paired registrations are performed by digitizing reference points (known in the virtual and the real world) on the CSG and the 3D printed distal femur using the 3D point digitizer, as shown in Figure 3.13.

While in theory, three reference points are sufficient for point-paired registration, for practical reasons and to increase accuracy, we obtained four reference points on the CSG (located at the corners of the device). Given the resulting registrations, the homogeneous matrices describing the position and orientation can be computed. From this transformation matrix, we derive the distance between the intended and the ac-

| Bone | | | Exposure | Type | No. CSGs |
|---|---|---|---|---|---|
| ASM | 0 | 0 | Full | Manual | 9 |
| | 0 | 0 | Full | Optimized | 3 |
| | 0 | 0 | Limited | Manual | 9 |
| | 0 | 0 | Limited | Optimized | 3 |
| ASM | -1 | 0 | Limited | Optimized | 2 |
| ASM | 1 | 0 | Limited | Optimized | 2 |
| ASM | 0 | -1 | Limited | Optimized | 2 |
| ASM | 0 | 1 | Limited | Optimized | 2 |
| ASM | -1 | -1 | Limited | Optimized | 2 |
| ASM | 1 | 1 | Limited | Optimized | 2 |
| ASM | 0 | 0 | Limited | $*$Optimized | 3 |
| Patient 1 | | | Limited | Optimized | 2 |
| Patient 2 | | | Limited | Optimized | 2 |

**Table 3.3:** Overview of the CSG-bone combinations that were tested. CSGs were tested on bones from actual patient data and bones extracted from an ASM, the input modii of variation are mentioned in the second, and third column. **Pin count was also optimized (8, 8 and 9 pins respectively)*



| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| NA | NA | NA | NA | . | X | NA | NA | NA | NA | NA |
| NA | NA | NA | . | . | . | . | NA | NA | NA | NA |
| NA | NA | . | . | . | . | . | NA | NA | NA | NA |
| NA | NA | . | . | . | . | . | . | NA | NA | NA |
| NA | X | . | . | X | . | . | . | . | NA | NA |
| NA | . | . | . | . | . | X | . | . | . | NA |
| NA | . | . | . | . | . | . | . | . | . | . |
| NA | . | . | . | . | . | . | . | . | . | . |
| NA | . | . | X | . | . | X | . | . | . | X |
| NA | X | . | NA | NA | NA | NA | NA | NA | NA | . |
| NA | NA | NA | NA | NA | NA | NA | NA | NA | NA | NA |

| ID | Location | Depth |
|---|---|---|
| 7 | A3 | 30.40 mm |
| 6 | E3 | 34.93 mm |
| 5 | E6 | 20.99 mm |
| 4 | F11 | 20.86 mm |
| 3 | G7 | 14.38 mm |
| 2 | H3 | 20.95 mm |
| 1 | J2 | 27.65 mm |
| 0 | J7 | 10.49 mm |

**Figure 3.15:** Interface for manually creating a pin-based CSG configuration. *Left:* User interface for choosing a pin configuration, in this case, the user can only pick a limited amount of pins because the exposure is limited. *Right:* Visualization of the pins on the surface of the cartilage/bone in the planned alignment.

tual location, and the angle between the intended and actual orientation vectors to verify the accuracy of the alignment.

## 3.5 RESULTS

The results from the experiments are depicted in Figure 3.16, Figure 3.17 and Figure 3.18. In these scatter plots, each marker represents a single CSG configuration that is tested on a 3D printed distal femur, the position of the marker denotes the maximum translation and rotation error that were measured during the experiments.

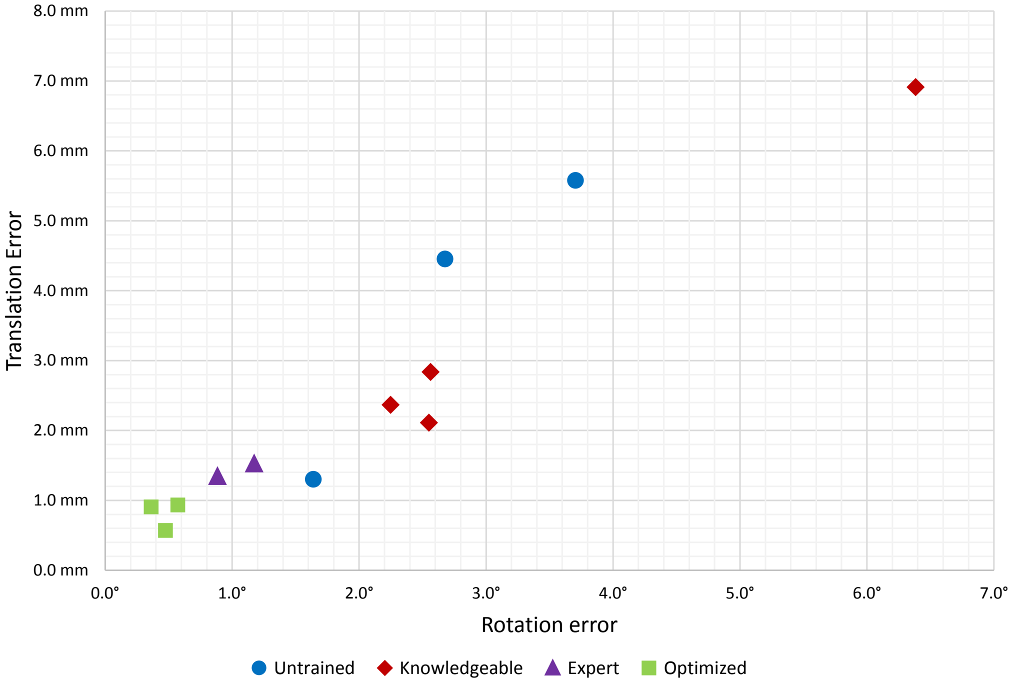


**Figure 3.16:** Alignment errors as a result of placing manually configured as well as computer-optimized CSGs (full exposure) on the mean distal femur from our ASM.

Figure 3.16 and Figure 3.17 indicate that the optimized CSG configurations always outperform manually configured CSGs. The optimization process will always ensure that the deviation threshold is respected — here $1.5$ mm and $< 1°$. The computer-optimized CSGs were also successfully placed in each trial, indicating that there is sufficient haptic feedback and stability to warrant a proper alignment. In contrast, there is a significant spread in alignment error among manually configured CSGs.

The surgeon-defined CSGs are superior to those of the novices for the full exposure, which might be due to the experts' substantial knowledge about human anatomy and morphology of the femur, and these full exposure CSGs could actually be considered acceptable. Nonetheless, there is no guarantee that such manual definitions will perform well, especially considering the spread of the various samples.

The situation actually changes drastically, when investigating the limited exposure scenario, which can also be considered more realistic. Here, the configuration process is more complex because certain surface regions need to be avoided. All manually defined devices perform significantly worse, including the surgeons, and the difference to our optimized CSGs becomes very obvious–see the cluster of optimized guides (all within $1.2$ mm translational error and $0.9°$ rotational error) versus the surgeon-defined CSGs, which now belong to the worst performing CSGs.

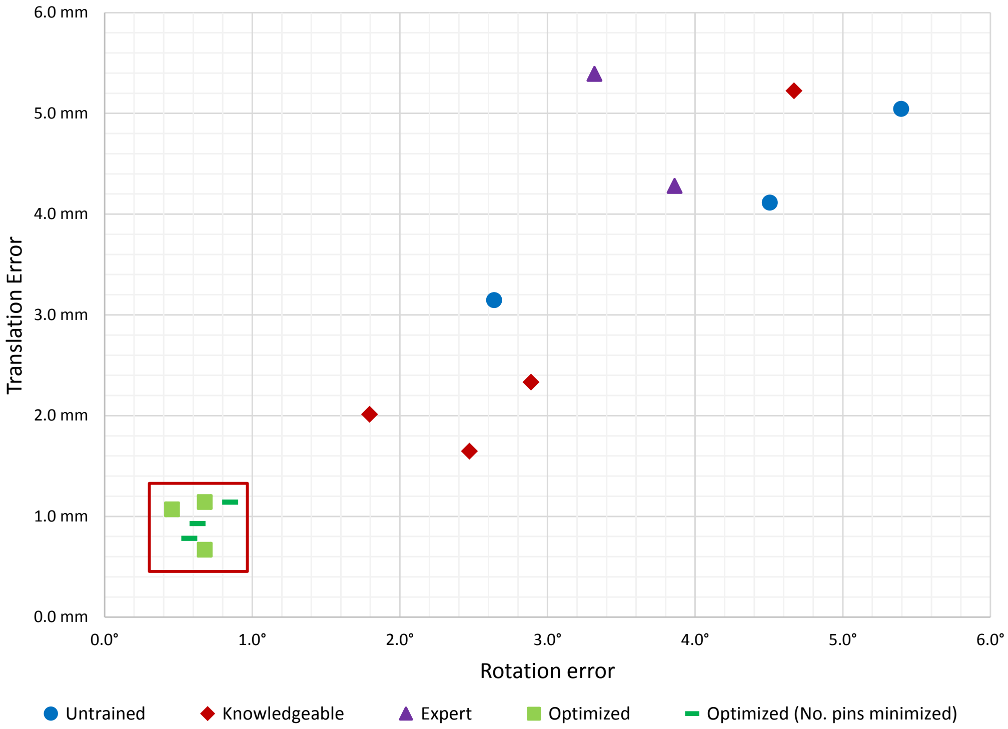


**Figure 3.17:** Alignment errors as a result of placing manually configured as well as computer-optimized CSGs (limited exposure) on the mean distal femur from our ASM. Three additional CSGs have been tested with a minimized number of pins.

The results from the experiments clearly indicate that manual CSG configuration is a delicate and complex task with often poor results, while the optimization framework consistently leads to reproducible and reliable configurations, even in cases where the number of pins is minimized, as shown in Figure 3.17.

In an additional verification step, we asked three participants to redo the user study three times, but none managed to improve their manual results significantly ($< 1.2$ mm and $< 0.9°$ compared to the optimized). This fact further underlines that configuring CSGs is not intuitive, even after experimenting for a considerable amount of time.

Finally, Figure 3.18 shows the errors of 19 computer-optimized CSGs on a variety of bone shapes. The deviation threshold is respected by all samples, which strongly suggests that our optimization method is robust to varying input geometry as well.

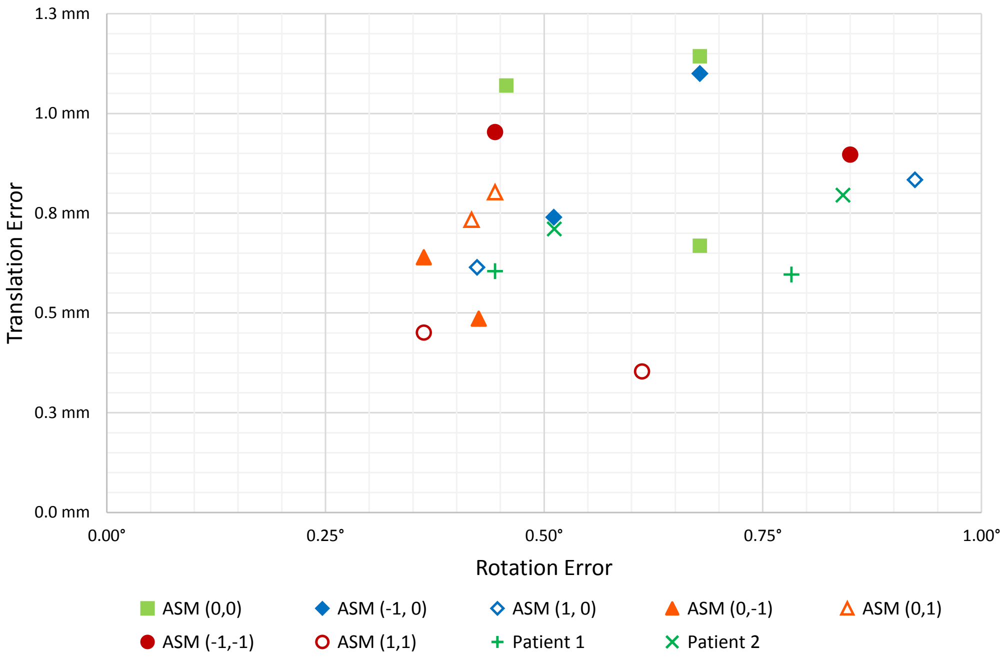


**Figure 3.18:** Alignment errors as a result of placing computer-optimized CSGs (limited exposure) on nine distal femora, two based on real patient data and seven based on shapes derived from our ASM.

## 3.6 DISCUSSION AND CONCLUSIONS

Alignment of prosthetic implants in joint replacement surgery has a significant impact on the survival of orthopaedic implants (Lotke and Ecker, 1977; Bäthis et al., 2004). Besides other factors, especially mal-alignment can lead to aseptic loosening, premature failure, and impaired range of motion (van Strien et al., 2009). Computer navigation (CT-based and CT-free) and patient-specific templating have improved the accuracy of alignment and reduced the chance of outliers (Mason et al., 2007; Ng et al., 2012).

Our study focusses on CSGs, and in particular on the computer-assisted definition of guide input parameters to warrant a reliable alignment during surgery. No current studies exist that describe such a method for CSG optimization. Predicting the actual alignment of CSGs is difficult and requires knowledge about the physical interactions between the CSG and the host bone. Non-surgeons, as well as experienced orthopaedic surgeons, struggle with the task. Not only is the alignment accuracy of human-defined CSGs low, but there is also a significant spread between subjects. In our method, we currently do not address the effect of imaging modality choices (CT/MRI) (White et al., 2008), their effect on bone reconstruction and subsequently alignment accuracy. The topic of uncertainty with respect to bone reconstruction and manual guide configuration accuracy is considered a topic on its own and could lead to a follow-up study, involving also a clinical pilot study.

In summary, we introduced a novel computer-assisted method to configure CSGs and predict their reliability during surgery. It will help surgeons to follow the planned alignment more closely, and ultimately lead to an improved surgical outcome. For the patient, this means less postoperative pain, improved function, and longevity of the joint. Although our method is validated on the knee joint, it is certainly not restricted to this application, since our optimization procedure is more general and can handle any arbitrary types of bone geometry. Furthermore, our simulation framework supports different guides as input, which makes it also interesting for PST designers. The pin-based CSG we used in this chapter can be applied to other joints as well, taking into account the specific joint anatomy and accessibility. The design of new CSGs, and exploring variations of the current one are interesting areas of future work. These aspects illustrate several of the advantages and the generality of our approach, which makes it widely applicable.

# Contact Visualization for Customizable Surgical Guides[1]

In the previous chapter, we described an optimization method for the automatic configuration of customizable surgical guides. Though the optimization itself is fully-automatic, some surgeon interaction is required before and after optimization. Prior to optimization, the surgeon has to mark regions on the surface of the cartilage/bone that are deemed safe for docking. In order to facilitate this process, accurate and faithful depiction of pathology is vital. After optimization, the surgeon can pick a preferred CSG configuration from a list of $n$ CSG configurations that meet the reproducibility criteria. They can further make small modifications to the chosen CSG configuration to assess what the effect is on the reproducibility using an interactive and iterative simulation process. The resulting CSG configuration is then used during surgery to provide *mechanical guidance* § 2.4.4. The visualizations, assuming that they provide a good visual depiction of the CSG-bone connection, can also provide additional *implicit guidance* § 2.4.1 in the form of a rudimentary validation step, either through the recollection of visual memory or by observing the visualization during surgery.

During planning and execution of CSG-based surgery, it is important to provide the surgeon with insightful and meaningful visual cues that clearly illustrate how a CSG connects to the bone. Lighting effects can contribute toward this goal, but not all techniques work equally well, as shown in Figure 4.1. The images in the left column show that diffuse lighting behaves poorly when it comes to visualizing contact. It is impossible to discern which of the two images depicts contact. The images in

[1] T. Kroes, D. Schut and E. Eisemann. To appear as a book chapter in GPU Pro 6: Advanced Rendering Techniques

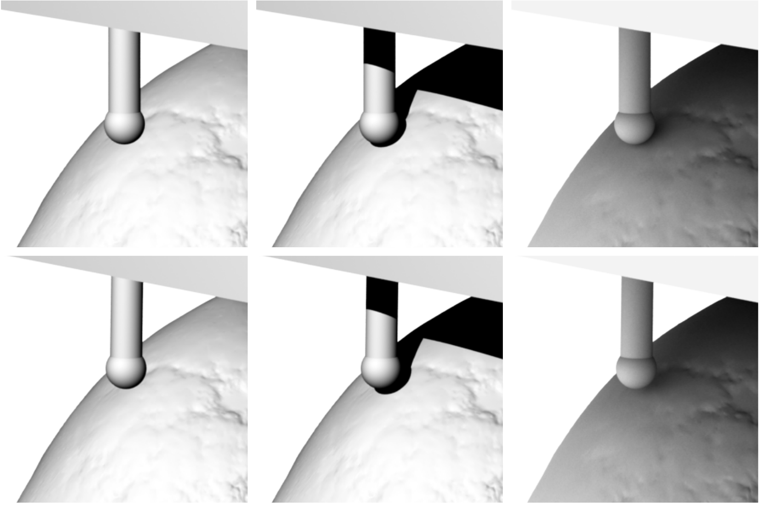


Figure 4.1: Three methods to depict CSG-bone contact. *Top row:* In the intended alignment, the CSG pins precisely touch the surface of the bone. *Bottom row:* If the CSG is incorrectly docked on the bone, one or more of the pins do not touch the bone. *Left column:* Diffuse lighting. *Middle column:* Diffuse lighting + shadows. *Right column:* Ambient occlusion.

the middle column show the effect of adding drop shadows. While the visualization of contact improves, its usefulness is, however, scene, light and camera dependent. First, drop shadows are only effective when other objects in the scene do not cast overlapping shadows. Second, the alignment of the camera and light source can compromise the interpretability of the contact visualization. For example, when the camera and light direction (nearly) coincide, the interpretation of contact becomes prohibitively difficult, as shown in Figure 4.2.

The images in the last column are generated with ambient occlusion. This method varies the light intensity at each shading point depending on how exposed the point is to the environment light. This method is effective at depicting local contact shadows, however it is less effective at communicating the global shape, as shown in the left image in Figure 4.3. In order to harness the power of ambient occlusion while still being able to explore the global shape of the bone, the environment color can be varied (e.g., changing color by lattitude), as shown in the right image in Figure 4.3.

Ambient occlusion is a computationally expensive algorithm. The previous ambient occlusion images were generated by an off-line ambient occlusion program, and its performance is far from real-time. In this chapter, we describe a near real-time ambient occlusion algorithm for volumetric data, in order to enhance the visual depiction of CSG-bone contact.

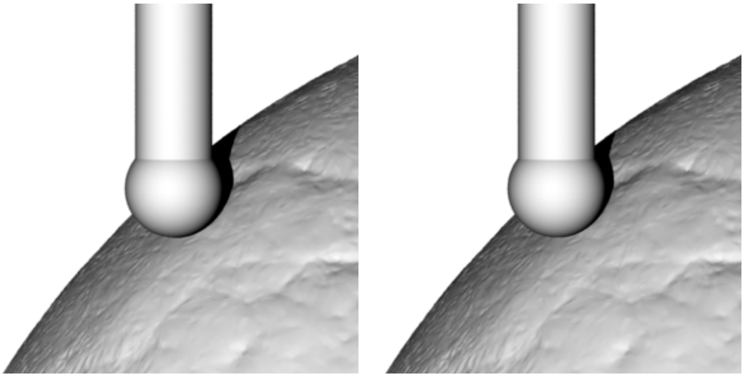


**Figure 4.2:** An example where drop shadows fail to effectively depict contact. In this scene, the camera light and light direction nearly coincide. As a result, it becomes virtually impossible to see which of the pins has contact with the surface of the bone. *Left:* Intended alignment, the pin touches the surface of the bone. *Right:* Incorrectly docked pin, which hovers over the surface of the bone.

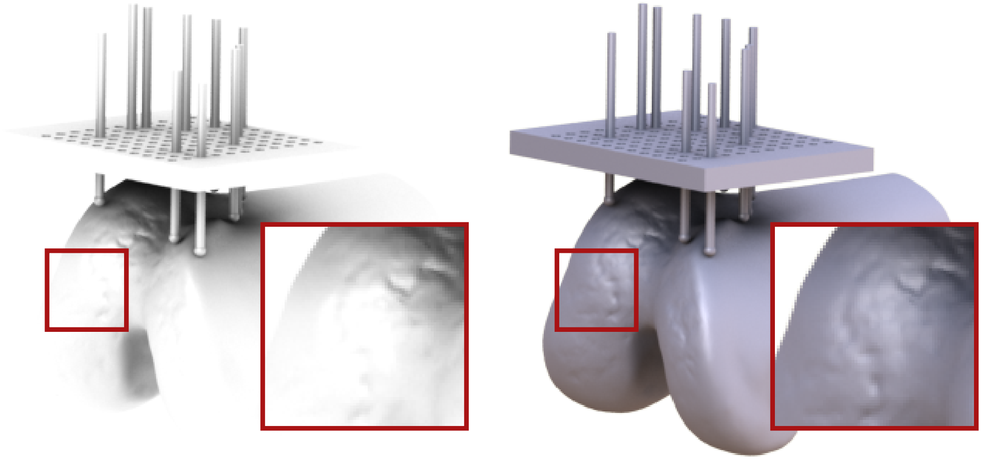


**Figure 4.3:** *Left:* Ambient occlusion with a uniform environment color, it is difficult to inspect the global shape. *Right:* Ambient occlusion where the environment color varies with latitude, this way the global shape can be inspected more easily.

## 4.1 INTRODUCTION

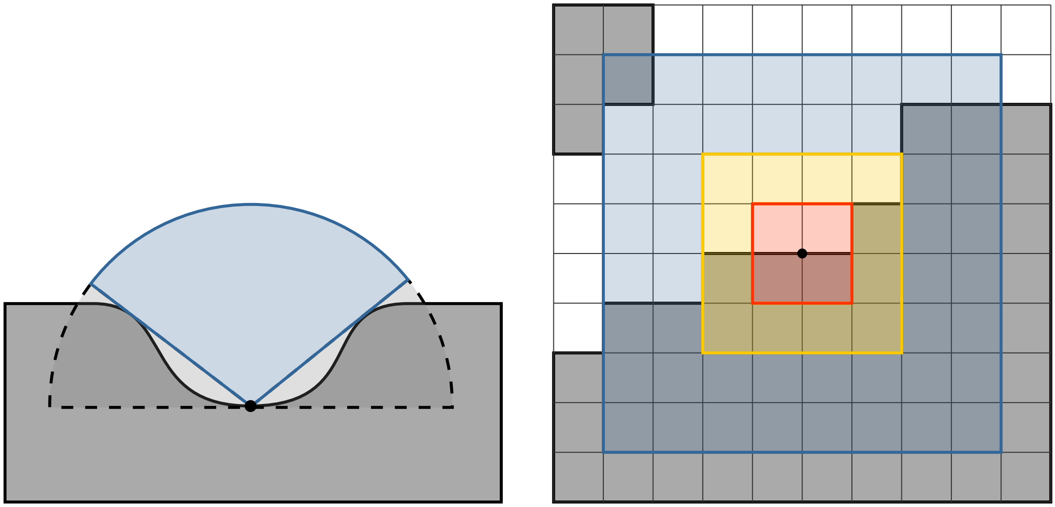
Ambient occlusion (Zhukov et al., 1998) is a compelling approach to improve depth and shape perception (Lindemann and Ropinski, 2011; Langer and Bülthoff, 1999), give the illusion of global illumination, and is an efficient way to approximate low-frequency outdoor lighting. In principle, ambient occlusion computes the light accessibility of a point, e.g., it measures how much a point is exposed to its surrounding environment.

An efficient and often used version of ambient occlusion is Screen Space Ambient Occlusion (Kajalin, 2009). It exploits the depth buffer to compute the visibility. This method is very appealing, since the computational overhead of this method is minimal. However it cannot be applied to direct volume rendering (DVR) because voxels are typically semi-transparent, as defined via a transfer function. Therefore, a depth buffer would be ambiguous and is not useful in this context.

The first method to compute ambient occlusion in direct DVR, called *Vicinity Shading*, was developed by Steward (Stewart, 2003). This method computes the ambient occlusion in each voxel by taking into account how much the neighboring voxels obscure it. Illumination is stored in an additional volume, which needs to be recomputed after scene modification. Similarly, (Hernell et al., 2010) compute ambient occlusion by raytracing inside a small neighborhood around the voxel. In (Kroes et al., 2012) the entire volume is taken into account.

Our approach tries to avoid costly ray tracing and casts the problem into a filtering process. In this sense, it also relates to (Penner and Mitchell, 2008), who use statistical information about the neighborhood of the voxels to estimate ambient occlusion, as well as the method by Ropinski et al. which is similar and also adds color bleeding (Ropinski et al., 2008b). Furthermore, our approach relates to (Crassin et al., 2010) who propose to use filtering for shadow and out-of-focus computations.

Our smooth probabilistic ambient occlusion (SPAO) is a novel and easyily implementable solution for ambient occlusion in DVR. Rather than applying costly ray casting to determine the accessibility of a voxel, this technique employs a probabilistic heuristic in concert with 3D image filtering. In this way, ambient occlusion can be efficiently approximated and allows the interactive modification of the transfer function, which is critical in many applications, for instance medical and scientific DVR. Furthermore, our method allows various quality trade-offs regarding memory, performance, and visual quality. Only very few texture look-ups are needed in comparison to ray-casting solutions and the interpretation as a filtering process ensures a noise-free smooth appearance.



(a) The hemisphere around a point which determines the ambient occlusion. The blue part is unoccluded.

(b) Hollow cubes around a point that are used to approximate the obscurance.

**Figure 4.4:** Volumetric obscurance techniques

## 4.2 METHOD

There are various definitions for ambient occlusion. Here, we define it as the part of a point that is accessible from the outside world. A 2D example is given in Figure 4.4(a) and illustrates the ambient occlusion computation.

More formally, the ambient-occlusion value $\mathcal{A}(p,n)$ is given by the integral of the visibility function over the hemisphere $\Omega$ centered around a point $p$ in the direction of the normal $n$ of that point. The visibility function $V(p,\omega)$ returns 0 if the ray from Point $p$ in direction $\omega$ is blocked and 1 if it is unblocked, or even intermediate values, if the transfer function is defined accordingly. In consequence, we obtain:

$$\mathcal{A}(p,n) = \frac{1}{\pi}\int_{\Omega(n)} V(p,\omega)d\omega$$

It is also possible to integrate the visibility function over the whole sphere around a point, hereby $\Omega$ is a full sphere, instead of a hemisphere, which is called obscurance, and produces similar results. Calculating obscurance instead of ambient occlusion has the advantage that it does not require a normal. However, this definition will lead to parts of the volume that are located behind the point to intervene in the computation. This can be a disadvantage for standard scenes, as the result might become too dark, but in the context of DVR, it is sometimes even preferable, as it will unveil information below the surface, which is often a desired property.

Both, ambient occlusion and obscurance, only depend on the geometry of the volume. Therefore they can be stored in an additional volume that is then used to modulate the original volume's illumination. The occlusion values can be calculated directly from the opacity of the original volume. Nonetheless, the values have to be recomputed when the original volume changes, for example when the user changes the transfer function. This latter step can be very costly and make it impossible to interact with transfer functions, while maintaining visual fidelity. Our approach is fast to compute and enables a user to quickly perform such modifications without having to wait a long time for the result.

Initially our solution will build upon obscurance, but in § 4.2.2, we will extend our algorithm to approach ambient occlusion by making use of the normals to reduce the influence of the part of the volume behind the surface.

### 4.2.1 OVERVIEW

To approximate obscurance at a certain point in the volume we avoid ray casting and instead provide an approximation, which is based on the probability of the rays to be blocked by the volume. Instead of solving $\mathcal{A}(p, n)$ and its integral entirely, we consider a limited region around $p$, formed by boxes of increasing size. The volume between successive boxes forms a shell, as shown in Figure 4.5, for which the probability of a random ray to be blocked is deduced. The various probabilities are then combined heuristically to yield the final occlusion information.

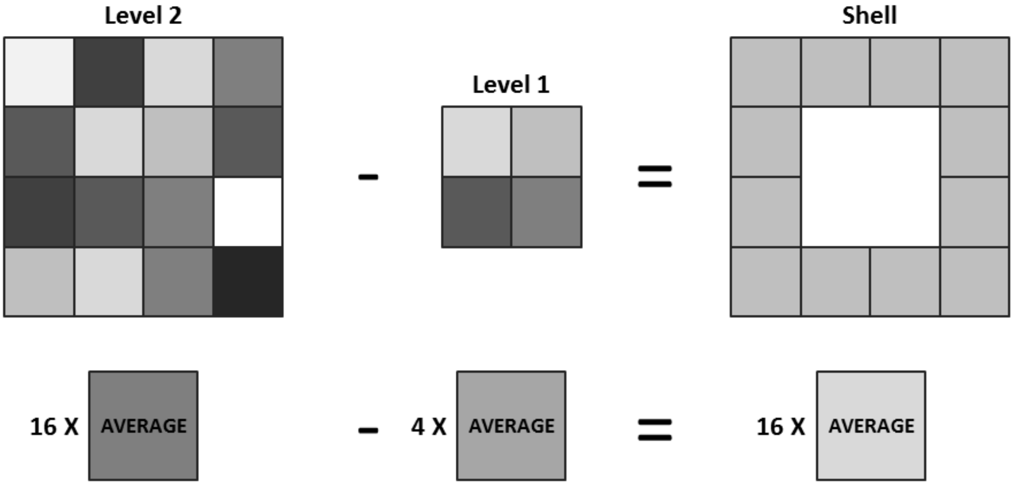


Figure 4.5: In this 2D illustration, the shell on the right is a one voxel thick hull that is formed by subtracting the average opacity from level 1 in the middle from level 2 on the left.

Specifically, if a shell is represented by a sphere with a one-voxel-wide boundary $S$, containing the volume data after being transformed by the transfer function, solving for $\mathcal{A}$ boils down to an averaging process. Hence, solving the integral for all rays in all directions, will result in the probability of a ray to be stopped and corresponds exactly to the average of all voxel values in the shell $Average_S(p)$.

If we now decompose the volume into a set of shells around a point, we can compute the probability of the rays to be stopped for each shell, but still need to combine all the blocking contributions together. In order to do so, we make use of a heuristic. By assuming a statistical independence between the value distributions in the various shells, the probability of rays originating at $p$ to be blocked by a set of $n$ englobing shells $\{S_i\}_{i=1}^n$ ordered from small to large is given by:

$$\prod_{i=1}^{n}(1 - Average_{S_i}(p)).$$

To understand this formula, it helps considering only two layers $\{S_1, S_2\}$ . A random ray from $p$ traverses $S_1$ with probability $(1 - Average_{S_1}(p))$. If this ray passed $S_1$, it is again, potentially, stopped by $S_2$, this time with probability $(1 - Average_{S_2}(p))$, yielding a total probability of $(1 - Average_{S_1}(p))(1 - Average_{S_2}(p))$. In the following, we will describe an efficient and GPU-friendly approach to compute an approximation of this solution.

### 4.2.2 APPROXIMATING OBSCURANCE FOR CUBE SHELLS

By using shells in stead of spheres, the computation of $Average_{S_i}$ becomes much more efficient. The cubes are chosen to be of increasing size and centered at each point $p$ of the volume. The shells are then defined by hollowing out these cubes by subtracting the next-smaller cube from its successor. In practice, these cubes will never have to be explicitly constructed, but it is helpful to think of them for illustrative purposes. The process is illustrated in Figure 4.4(b).

Following the previously described algorithm, we need to deduce $Average_{S_i}$, which in our new situation corresponds to the average of all voxel values between two successive cubes. We assume for now that we have a quick way of determining the average inside of a complete cube, then we could rapidly determine $Average_{S_i}$.

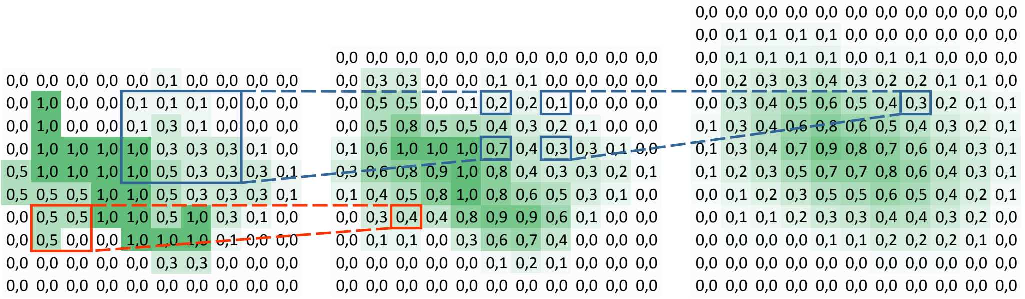
To explain the computation, we will illustrate the steps for an $Average_S$ of a shell $S$ defined two cubes $C_1$ and $C_2$, with voxel-value averages $A_1$ and $A_2$ and number of voxels $S_1$, $S_2$ ($S_1 < S_2$), respectively:

$$Average_S = (S_2 A_2 - S_1 A_1)/(S_2 - S_1)$$

In other words, we subtract from the total voxel sum of one cube $S_2 A_2$, the total voxel sum of the next-smaller one ($S_2 A_2$) and normalize the result by the number of voxels in the shell between both.

Please notice, that the above formula can be rewritten to: $Average_S = 1/(1 - S_1/S_2)(A_2 - (S_1/S_2)A1)$ - in consequence, only the average and the relative change in size ($S_1/S_2$) is needed to deduce $Average(S)$, which facilitates computations in practice. Imagine, each cube is obtained by doubling the length of each edge of the predecessor. Then the ratio would be $1:8$, resulting in $Average_S = 8/7(A_2 - 1/8A_1)$.

### 4.2.3 FAST CUBE AVERAGES



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,1 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 1,0 | 0,0 | 0,0 | 0,1 | 0,1 | 0,1 | 0,0 | 0,0 | 0,0 |
| 0,0 | 1,0 | 0,0 | 0,0 | 0,1 | 0,3 | 0,1 | 0,0 | 0,0 | 0,0 |
| 0,0 | 1,0 | 1,0 | 1,0 | 1,0 | 0,3 | 0,3 | 0,3 | 0,1 | 0,0 |
| 0,5 | 1,0 | 1,0 | 1,0 | 1,0 | 0,5 | 0,3 | 0,3 | 0,3 | 0,1 |
| 0,5 | 0,5 | 0,5 | 1,0 | 1,0 | 0,5 | 0,3 | 0,3 | 0,3 | 0,1 |
| 0,0 | 0,5 | 0,5 | 1,0 | 1,0 | 0,5 | 1,0 | 0,3 | 0,1 | 0,0 |
| 0,0 | 0,5 | 0,0 | 0,0 | 1,0 | 1,0 | 1,0 | 0,1 | 0,0 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,3 | 0,3 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,3 | 0,3 | 0,0 | 0,0 | 0,1 | 0,1 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,5 | 0,5 | 0,0 | 0,1 | 0,2 | 0,2 | 0,1 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,5 | 0,8 | 0,5 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0,0 | 0,0 |
| 0,1 | 0,6 | 1,0 | 1,0 | 1,0 | 0,7 | 0,4 | 0,3 | 0,3 | 0,1 | 0,0 |
| 0,3 | 0,6 | 0,8 | 0,9 | 1,0 | 0,8 | 0,4 | 0,3 | 0,3 | 0,2 | 0,1 |
| 0,1 | 0,4 | 0,5 | 0,8 | 1,0 | 0,8 | 0,6 | 0,5 | 0,3 | 0,1 | 0,0 |
| 0,0 | 0,3 | 0,4 | 0,4 | 0,8 | 0,9 | 0,9 | 0,6 | 0,1 | 0,0 | 0,0 |
| 0,0 | 0,1 | 0,1 | 0,0 | 0,3 | 0,6 | 0,7 | 0,4 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,1 | 0,2 | 0,1 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,1 | 0,1 | 0,1 | 0,1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,1 | 0,1 | 0,1 | 0,1 | 0,0 | 0,1 | 0,1 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,2 | 0,3 | 0,3 | 0,4 | 0,3 | 0,2 | 0,2 | 0,1 | 0,1 | 0,0 |
| 0,0 | 0,3 | 0,4 | 0,5 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0,1 |
| 0,1 | 0,3 | 0,4 | 0,6 | 0,8 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 |
| 0,1 | 0,3 | 0,4 | 0,7 | 0,9 | 0,8 | 0,7 | 0,6 | 0,4 | 0,3 | 0,1 |
| 0,1 | 0,2 | 0,3 | 0,5 | 0,7 | 0,7 | 0,8 | 0,6 | 0,4 | 0,3 | 0,1 |
| 0,0 | 0,1 | 0,2 | 0,3 | 0,5 | 0,5 | 0,6 | 0,5 | 0,4 | 0,2 | 0,1 |
| 0,0 | 0,1 | 0,1 | 0,2 | 0,3 | 0,3 | 0,4 | 0,4 | 0,3 | 0,2 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,1 | 0,1 | 0,2 | 0,2 | 0,2 | 0,1 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |
| 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 |

**Figure 4.6:** 2D Example of how N-buffers are calculated. A dataset is shown on the left, with the first two n buffer levels next to it. In each level the average of four values of the previous level is combined into one value.

In the previous section, we assumed that the average inside of a cube can quickly be determined. Here, we will propose two possible solutions to this problem, based on the observation that, for a given cube size, the averages are actually given by box-filtering the volume.

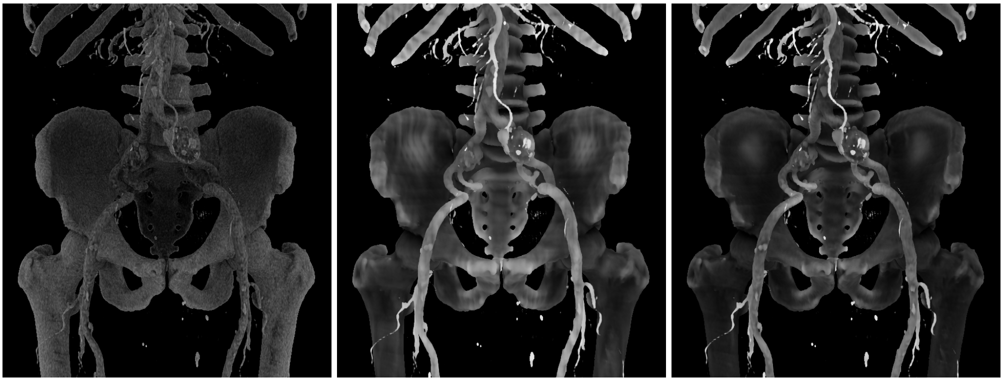
Determining averages of various kernel sizes is a common problem in computer graphics in the context of texture mapping. These techniques translate to corresponding operations in a 3D volume. The most common such approximation is mipmapping, but we will also present the use of N-Buffers (Décoret, 2005).

As mipmaps are rather standard, we will only focus on N-buffers here. Similar to mipmaps, they consist of multiple levels $l$, each representing the average values of the original volume inside cubes of width $2^l$. Unlike mipmaps, the resolution of N-buffers is not reduced in each level. In consequence, it is possible to retrieve the exact filled part of a cube at every position in the volume, whereas for a mipmap linear interpolation can provide only an approximation based on the eight closest points voxels, which reduces the quality, as shown in Figure 4.7.

The N-Buffer construction is efficient, as each new level can be computed from the previous using only 8 lookups. A 2D example of the calculation is shown in Figure 4.6. Nonetheless, N-Buffers result in higher memory consumption, so it can be useful to apply a few mipmap levels before processing the rest using N-Buffers.

### 4.2.4 APPROXIMATING AMBIENT OCCLUSION

We explained earlier that ambient occlusion in comparison with obscurance can provide cues that are closer to realistic lighting because voxels behind the point of interest



(a) Raytacing (256 rays/voxel) (b) Mipmap filtering (c) N-buffer filtering

Figure 4.7: Volumetric obscurance technique comparison.

are not taken into account. To reduce this effect, we can offset the lookup operations in the direction of the normal. When choosing the offset carefully, the increase in size of the cubes and the offset can be correlated to obtain shells that correspond now to hemispheres. This goal can be achieved by multiplying the normal vector by half the size of the box. An example with a shorter vector is illustrated in Figure 4.8.

However, in DVR a normal is not always clearly defined, e.g., inside a homogeneous semi-transparent volume, e.g., a jelly pudding. Similarly, between two different semi-transparent voxels it might be less clear how to define a normal at the interface between opaque and transparent materials. In consequence, we propose to scale the cube offset based on how strong the gradient is. Interestingly, while most techniques derive normals from the normalized gradient via central differences, we can use this gradient to determine if the normal is pronounced. Furthermore, by removing the normalization operation and instead normalizing the voxel values themselves to a range [0,1], the gradient itself corresponds to a appropriately-scaled normal. Additionally, we allow the user to specify a global scale to either pronounce or reduce the impact of this ambient-occlusion approximation, as shown in Figure 4.9).

## 4.3 RESULTS

The method presented in this chapter has been implemented in a CUDA-based standalone software program for direct DVR. The program and its source code are available under the original BSD license. It is shipped with sample data sets. The transfer function and, thus, the visual representation can be changed on-the-fly. Also, the user can select from three different methods of ambient occlusion computation: mipmaps, N-buffers, and ray tracing. Our program makes use of CUDA 3.0 texture objects and will not support lower CUDA versions.

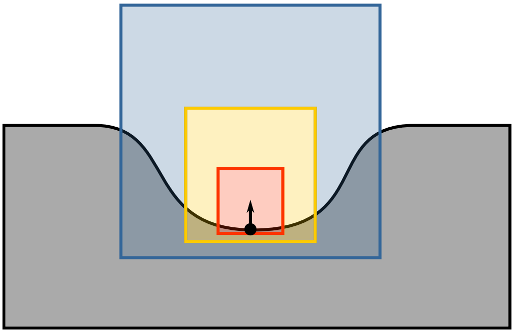
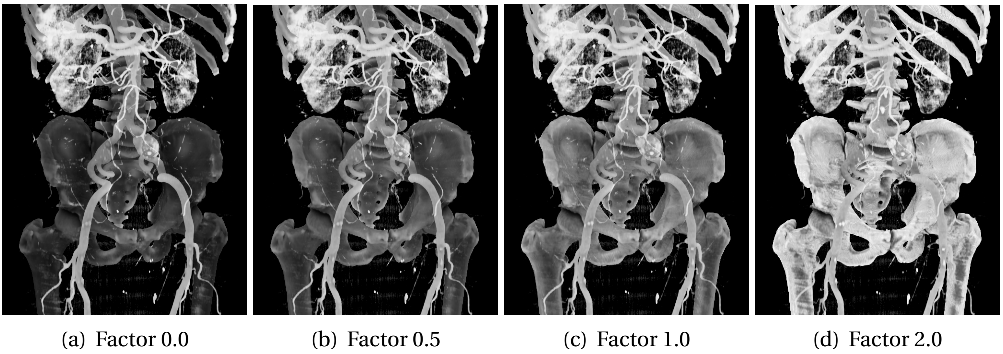


Figure 4.8: The lookups of the cubes from a point with a normal of length 0.75 in the upwards direction.



(a) Factor 0.0

(b) Factor 0.5

(c) Factor 1.0

(d) Factor 2.0

Figure 4.9: Effect of the normal factor

We tested the performance of our technique using the publicly available Macoessix data set from the Osirix Website [2], as shown in . All tests were peformed on an Intel(r) Xeon(r) W3530 (2.80 GHz) workstation with 12 GB RAM and a GeForce GTX TITAN Graphics Card with 4 GB of RAM. N-buffers are slightly more costly than Mipmaps, but both are orders of magnitude faster than a volumetric ambient-occlusion ray tracer. The latter takes more than four minutes, as shown in Table 4.1.

Figure 4.3 and Figure 4.3 show the results of our approach on the Backpack and Manix data sets.

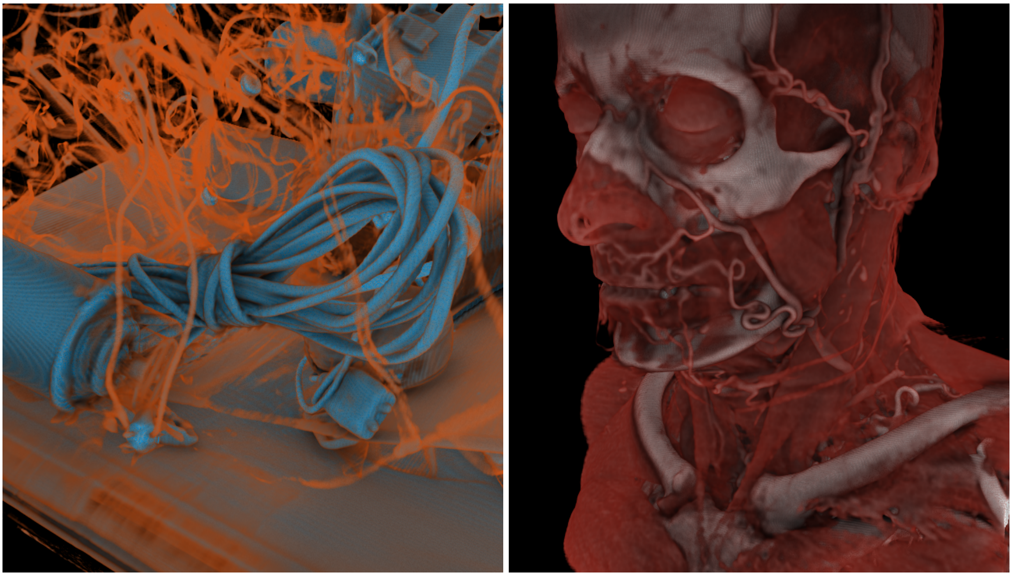
## 4.4 CONCLUSIONS

This chapter presents a novel approach to compute ambient occlusion for direct DVR. We demonstrate that, by considering the ambient occlusion computation as a filtering process, we can significantly improve efficiency and make it usable in a real-time DVR

[2] http://www.osirix-viewer.com/datasets/

| | N-buffers | Mipmaps | Raytaced, 512 rays |
|---|---|---|---|
| Level 0 | 30.93 ms | 33.00 ms | - |
| Level 1 | 33.99 ms | 4.58 ms | - |
| Level 2 | 40.13 ms | 0.66 ms | - |
| Level 3 | 41.16 ms | 0.17 ms | - |
| Level 4 | 42.69 ms | 0.14 ms | - |
| Level 5 | 38.09 ms | 0.13 ms | - |
| Level 6 | 41.91 ms | 0.12 ms | - |
| Levels Total | 268.90 ms | 38.8 ms | - |
| AO Computation | 63.24 ms | 110.39 ms | 425.36 sec |
| Total | 332.14 ms | 149.19 ms | 425.36 sec |

**Table 4.1:** Performance measurements for the Macoessix data set (512 x 512 x 512) for N-buffers and mipmaps based SPAO. For each technique we show the time it takes to compute the individual levels and to combine them into an ambient occlusion volume.



(a) Backpack data set

(b) Manix data set

**Figure 4.10:** SPAO applied to the Backpack (512 x 512 x 461) and Manix (512 x 512 x 460) data sets.

application. Such an approach is useful for medical visualization applications were transfer functions are very often subject to change.

Our approach is very simple and efficient and it leads to a very good quality/performance tradeoff. Nonetheless, we also experimented with more complex combinations of the shells, especially, as the assumption of independence of the occlusion probabilities is usually not true in most data sets. In practice, it turns out that our solution still works well and any increase in complexity also led to a significant performance impact. Nonetheless, this topic remains interesting future work. Further, we would like to investigate to approximate physically-plausible light transport, such as global illumination, with our filtering technique, which could further enhance the volume depiction.

# Remote Visualization for Collaboration and Doctor-patient Communication[1]

The CSG planning method, described in Chapter 3, requires significant computing power, and only runs efficiently on dedicated hardware with special computing capabilities. Deploying the software on consumer (tablet) PCs, laptops, or mobile devices is not feasible. First, these devices lack the required computing power and/or storage capabilities. Second, the CSG optimization process occupies all computing power which implies that, during its operation, the device cannot be used for other purposes. Third, patient data cannot leave the hospital due to privacy issues.

However, as patients are becoming more involved in their treatment plan, it is not uncommon that they will ask a radiologist for a copy of their medical data. Some hospitals even provide a copy of the medical data per default. When a patient receives the tomographic data, they typically have to explore it in a slice-by-slice manner, which is the de facto standard in medicine. However, for non-experts, these slices are difficult to interpret, because they do not have the anatomical/pathological background and they are typically more interested in the three dimensional shape. In this case, realistic direct volume rendering (DVR) can help to convey shape, depth, and size. However, this type of rendering is computationally expensive, and requires powerful hardware. Ideally, the tomographic data can be explored from a distance with remote visualization, irrespective of the computing power of the device.

Remote visualization systems are either model-based or image-based. In model-based systems, data sets, or parts of it, are exchanged between server and client, which

[1] T. Kroes, F. H. Post, C. P. Botha, M. Neven and M. Eisemann. Published in the Public Library of Science (Kroes et al., 2012).

is an issue if the data is privacy sensitive. The size of the volume data sets also prohibits use on mobile devices due to download time and network pressure. Image-based systems, such as ParaViewWeb (Jourdain et al., 2010) solve the privacy issues by only sending graphical depictions of the data back to the client.

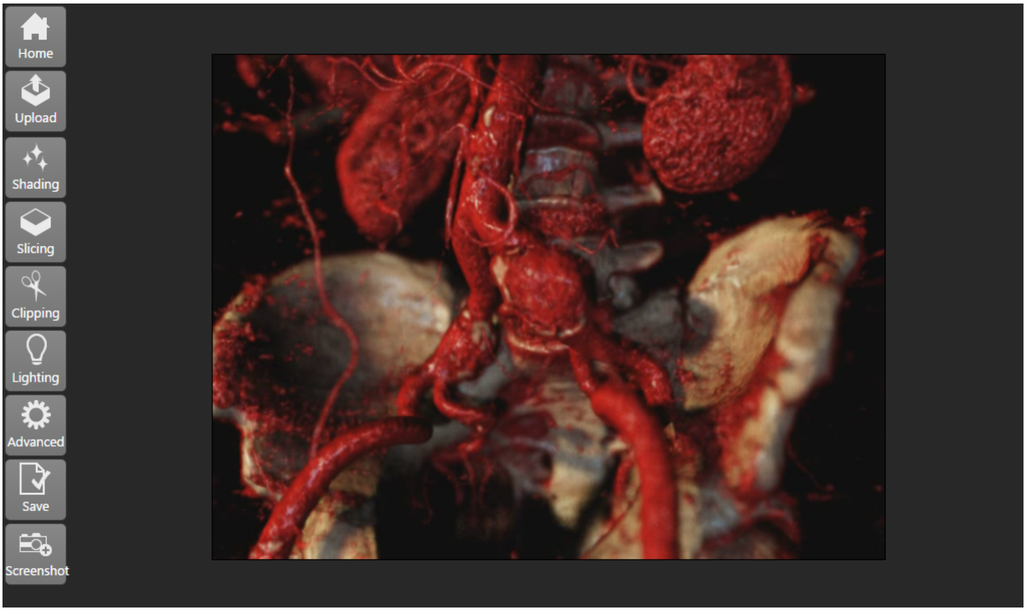
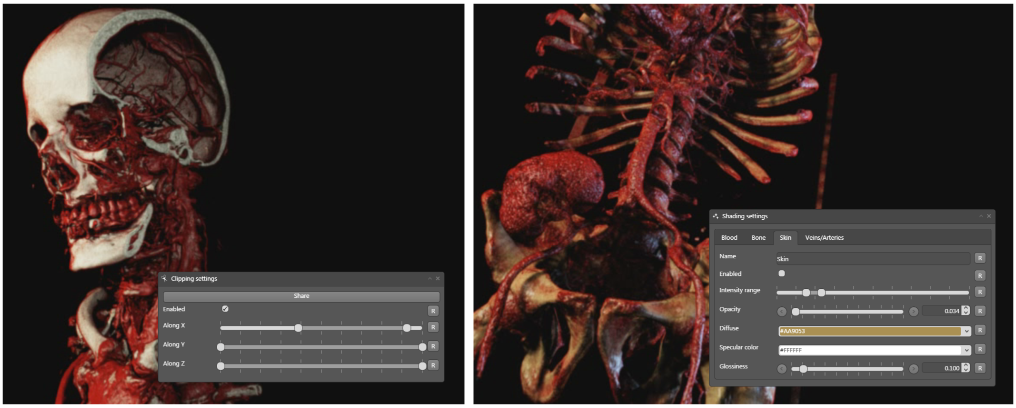


Figure 5.1: Screenshot of the remote visualization framework. The buttons on the left open windows that allow a user to manipulate the visualization. The center contains the video stream, as computed by the visualization server. The only prerequisites for this visualization are a state-of-the-art browser and an internet connection.

In an attempt to improve doctor-patient communication and patient self health management, we developed an image-based remote visualization prototype for exploration of tomographic data, as shown in Figure 5.1. In our implementation, doctors and patients upload DICOM[2] files, similar to the commercially available VisuApps tool for online volume visualization. When DICOM files are uploaded, users connect to the dedicated visualization server with a state-of-the-art browser. This server renders the images, which are sent back to the client in the form of a MPEG-4 encoded video stream. The user can modify all scene parameters through a JavaScript enabled HTML5 page, as shown in Figure 5.2.

Our remote visualization framework was built with generality in mind, in theory any visualization system can be integrated. This means that our system can also be used to enhance CSG-based joint replacement surgery. If the planning software from Chapter 3 is integrated in our remote visualization framework, the planning becomes

[2] Digital Imaging and Communications in Medicine



Enabled ☑

Along X

Along Y

Along Z

Name: Skin

Enabled □

Intensity range

Opacity: 0.034

Diffuse: #AA9053

Specular color: #FFFFFF

Glossiness: 0.100

**Figure 5.2:** Interaction with the remote visualization system. *Left:* By adjusting the clipping of the medical volumetric data set the inner parts are revealed. *Right:* Adjusting the shading parameters (transfer function) of the volume rendering.

accessible to a wide range of devices, irrespective of the location. This implies that the planning can even be queried and modified in the operating room, and the fact that the framework accepts multiple concurrent connections makes online collaboration possible.

The remainder of this chapter describes Exposure Render, an interactive direct volume rendering tool that incorporates physically-based lighting. This tool has been developed with doctor-patient communication in mind. Exposure Render integrates a host of effects e.g., shadows and reflections, which are known to contribute to the understanding of shape, depth and spatial relationships.

## 5.1 ABSTRACT

The field of volume visualization has undergone rapid development during the past years, both due to advances in suitable computing hardware and due to the increasing availability of large volume datasets. Recent work has focused on increasing the visual realism in Direct Volume Rendering (DVR) by integrating a number of visually plausible but often effect-specific rendering techniques, for instance modeling of light occlusion and depth of field. Besides yielding more attractive renderings, especially the more realistic lighting has a positive effect on perceptual tasks. Although these new rendering techniques yield impressive results, they exhibit limitations in terms of their flexibility and their performance. Monte Carlo ray tracing (MCRT), coupled with physically based light transport, is the de-facto standard for synthesizing highly realistic images in the graphics domain, although usually not from volumetric data. Due to the stochastic sampling of MCRT algorithms, numerous effects can be achieved in a relatively straight-forward fashion. For this reason, we have developed a practical framework that applies MCRT techniques also to direct volume rendering (DVR). With this work, we demonstrate that a host of realistic effects, including physically-based lighting, can be simulated in a generic and flexible fashion, leading to interactive DVR with improved realism. In the hope that this improved approach to DVR will see more use in practice, we have made available our framework under a permissive open source license.

## 5.2 INTRODUCTION

Realistic illumination in volume visualization plays a central role in 3D shape perception. For example, in (Lindemann and Ropinski, 2011) a user study is performed, in which the effectiveness of seven state of the art DVR techniques is measured, clearly showed that global illumination models help in assessing depth and size in images. Furthermore, in (Ropinski et al., 2010) a realistic lighting model for volume rendering is developed, and the work demonstrates that by using realistic lighting, observers use less time and are more accurate at assessing depth in a volume rendering.

Recent years have seen a great deal of research toward enhancing interactive direct volume rendering (DVR) approaches with more realistic illumination, for example ambient occlusion (Zhukov et al., 1998), shadows (Behrens and Ratering, 1998; Hadwiger et al., 2006), realistic scattering (Rezk-Salama, 2007; Ropinski et al., 2008b) and global illumination (Beason et al., 2006). However, research up to now has focused on fast approximations of illumination that could be integrated with GPU-based volume renderers, both texture- and raycasting-based, as the physically-based modeling of illumination was considered to be prohibitively expensive.

In contrast to many of the existing approximations, Monte Carlo ray tracing (MCRT), combined with physically based light transport, is able to simulate real-world light interaction without compromising accuracy of light transport computations, thus resulting in more realistic images. Monte Carlo rendering algorithms are capable of dealing with complex lighting, material and camera configurations. It has been demonstrated that MCRT, with suitable modifications addressing hardware peculiarities, can be performed on the the GPU (van Antwerpen, 2011; Novák et al., 2010). However, to the best of our knowledge, MCRT has not yet been applied to the complete interactive DVR pipeline in order to achieve photo-realism. (Rezk-Salama, 2007) comes the closest, but is based on explicitly using isosurfaces in the rendered volume data. This distinction is discussed in more detail in § 5.3.

In this work, we apply MCRT to the interactive rendering of volumetric datasets, sampling the whole domain and taking into account the full gamut of volume densities. In order to combine surface and volumetric scattering, we introduce *hybrid scattering*.

Our DVR framework is able to generate high quality images at interactive speeds. It builds up images progressively, where a recognizable rendering appears within a fraction of a second and image quality increases rapidly. The rendering can be interacted with from the very start. All scene parameters, e.g., transfer function, camera and lighting, can be modified interactively. Based on our experiments we conclude that stochastic MC based simulation of light transport is an attractive solution to the problem of photo-realistic rendering in interactive DVR. Stochastic MC based simulation of light transport is particularly interesting because it enables the integration of various physically based effects into a unified approach without significant effort,

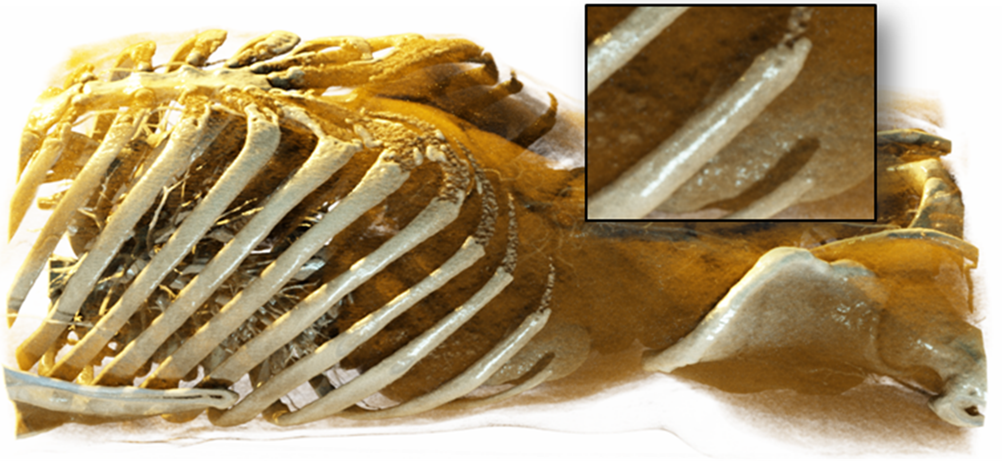


Figure 5.3: Image rendered with hybrid scattering. Note how detail is preserved through specular highlights, while still fully supporting volumetric scattering.

whereas other solutions restrict the number of lights, the shape of lights, the camera model, and so forth. Furthermore, due to its sampling approach, problems with aliasing and stepping artifacts are dealt with easily. The DVR framework is able to cope with complex lighting on the fly, and the increased quality of the images help to convey shape and detail. With this work, our contributions are the following:

- We demonstrate that MCRT is an appealing approach to DVR, allowing a host of effects and flexible lighting schemes, and that it can be run efficiently on commodity graphics hardware using CUDA, with the necessary performance optimizations.

- We present a re-usable GPU-based interactive direct volume renderer that integrates stochastic ray-traced lighting, thus enabling physically-based volumetric shadows, any number of arbitrarily positioned, shaped and textured area lights and finally the modeling of a real-world camera, including its lens and aperture. Our complete implementation is available under a permissive open source license, hopefully stimulating collaboration and allowing others to reproduce and further improve our work.

In 2009, Banks and Beason reported that the market penetration of physically-based illumination in scientific visualization was nearly zero in 2008, despite many indications in research that it has perceptual advantages (Banks and Beason, 2009). One of the reasons they cited, was purely that the scientists generating the visualizations did not have easy access to global illumination implementations in their standard workflow. By making our work available as a reusable and permissively licensed implementation, we hope to contribute to the uptake of physically-based illumination in interactive direct volume rendering.

The rest of the paper is structured as follows. In § 5.3 we survey related work on volumetric shadows, ambient occlusion and physically-based light transport. In § 5.4, we document our technique and in § 5.5 we analyze the performance and present some example renderings. Finally, we summarize our findings and point out directions for future research in § 5.6.

## 5.3 RELATED WORK

Global illumination, and especially shadows, are compelling ways of conveying depth and shape in 3D visualization in general (Wanger et al., 1992) and in volume rendering specifically. Some of the first improvements of the more straightforward light transport approximations, such as that presented by Max (Max, 1995), were made with the introduction of shadows in volume rendering. Behrens and Ratering introduced shadows in volume rendering by pre-computing a shadow volume, for a given relative light source position, that could then be rendered using a standard texture-based volume rendering algorithm (Behrens and Ratering, 1998). (Kniss et al., 2002, 2003) presented a volumetric lighting model that integrated half angle slicing, a texture-based volume rendering technique to calculate volumetric shadows, a lookup table-based phase function implementation and an approximation of multiple forward scattering based on aggregating light from previous slices.

(Hadwiger et al., 2006) adapted deep shadow maps (Lokovic and Veach, 2000), a technique for computing semi-transparent volumetric shadows, to raycasting on the GPU. (Ropinski et al., 2008a) presented an alternative implementation of deep shadow maps for GPU raycasting that supported caching when the light source configuration was kept constant, and compared it to normal shadow maps (non-transparent shadows) and shadow rays. In all cases, volume rendering realism was greatly improved with the integration of shadows. However, the mentioned examples were all limited to modeling a single point light source. The DVR framework in this chapter does not pose restrictions on the lighting configuration, e.g., the number of lights, their shape and finally their texturing.

Ambient occlusion (AO), introduced by (Zhukov et al., 1998) with the term *obscurances*, is an effective and usually inexpensive technique for approximating global illumination. In their survey, Méndez-Feliu and Sbert point out the difference between obscurances and ambient occlusion: Whilst the latter represents the degree of openness of a point, the former also takes into account diffuse indirect lighting, yielding more physically correct lighting and for example color bleeding effects (Méndez-Feliu and Sbert, 2009). However, the two terms are often used interchangeably. The vicinity shading method introduced by Stewart (Stewart, 2003) was the first to incorporate ambient occlusion into volume rendering, with a method called vicinity shading. Their method uses neighboring voxels and their obscurances to compute local illumination, which results in darkened crevices and depressions. Their method requires preprocessing for every scene modification and requires an additional buffer to store the results of the illumination. Penner and Mitchell (Penner and Mitchell, 2008) used histograms to classify the obscurance around a voxel. The method by (Ropinski et al., 2008c) used local histogram clustering for the pre-computation of occlusion information. It is important to note that AO and even obscurance do not take into account a specific light position, but are both based on the geometric occlusion of a sample point, and hence approximate the light that could conceivably reach that point. In the DVR framework presented in this chapter, the entire volume domain is taken into account for shadow computations.

(Ritschel, 2007) combine a form of ambient occlusion represented as spherical harmonics (SH), which they call the visibility function, with a DVR approach, relating the emission at each point to both its density and the interaction between the incoming light, from a single source, and the direction-dependent visibility function. (Lindemann and Ropinski, 2010) extend the work of (Ritschel, 2007) with a SH representation of the incident direct and indirect lighting that integrates chromatic attenuation as well as a local approximation of subsurface scattering. They claim to support arbitrary area light sources, but from the paper it is not clear how these are defined. In the work of (Kronander et al., 2011) lighting, as well as visibility are encoded in the spherical harmonics. Their method supports directional, point and environment lights. In contrast to our DVR framework method, this type of rendering requires extra storage, sensitive to the number of SH coefficients used.

(Schott et al., 2009) introduced Directional Occlusion, inspired by the AO algorithm. Their method is limited to a light source that has to coincide with the camera. This restriction was later partially removed by allowing the user to place a light within a hemisphere, oriented toward the camera, with the introduction of a multi directional occlusion model by (Šoltészová et al., 2010). Ropinski et al. enhanced their GPU raycasting framework with simulated scattering and shadowing (Ropinski et al., 2010). The illumination volume was generated with slice-based front-to-back chromaticity accumulation, and in a second pass back-to-front scattering accumulation, and could be generated on the fly when the transfer function or light position was updated. All of these methods yield impressive results, do not require significant pre-computation, and run at interactive frame rates. However, all are limited to modelling a single point light source.

The previously discussed papers yield results that have proven to aid in the perception of shape and depth. The methods render a fairly good approximation of real light interaction with volumes or iso-surfaces. Another class of volume renderers takes a more physically based approach. They typically solve the light transport equation for a volume in a pre-processing step and store the result in an additional buffer. (Wyman et al., 2006) introduced such a method, in which the direct lighting, shadows and diffuse inter-reflection are captured in an illumination buffer. This buffer is then used to texture iso-surfaces. This method was further developed by (Beason et al., 2006), introducing translucency and caustics, at the cost of static lighting. Both of these approaches focus on rendering isosurfaces and are not able to do full volumetric rendering.

(Csébfalvi and Szirmay-Kalos, 2003) were the first to apply Monte Carlo integration to volume rendering, with the aim to find a solution to the problem of data sets not fitting on graphics hardware memory. In a pre-processing step, this method generates a point cloud of random samples according to the volume's probability density function. During progressive rendering, from this point cloud, new samples are generated with importance sampling, which are projected back on the image plane and subsequently the pixel intensity is determined by Monte Carlo integration. This method does include gradient vector based lighting effects, for comparison purposes, but does not focus on photo-realistic rendering and ignores occlusion. This work is extended to support real-time modification of the transfer function (Csébfalvi and Szirmay-Kalos, 2003).

(Rezk-Salama, 2007) presented a GPU framework for Monte Carlo rendering of volumetric data sets. Their work comes closest to ours in the sense that they employ stochastic sampling techniques. However, they render a number of layers using an isosurface in the volume as basis for calculating later scattering. The first pass calculates local illumination on the selected isosurface, the second pass is an AO pass and the final layer, usually rendered with a single pass, models scattering. This is done by starting a transmissive ray at the isosurface, scattered within a Phong lobe, and reflecting this ray from the second isosurface it hits until it exits the volume. The three layers are composited to form the final image. Our DVR framework renders the volume in a unified way, and deliberately does not treat isosurfaces differently.

(Schlegel et al., 2011) have developed several rendering optimizations based on raycasting. As a result they are able to render ambient occlusion, volumetric shadows and color bleeding in real-time. This method includes soft shadows, AO and color bleeding.

(Schott et al., 2011) have applied depth-of-field to DVR using a slice based approach. This method employs incremental filtering to blur both iso surfaces and transparent sections. This method creates realistic depth of field effects, at the expense of relatively low refresh rates. Furthermore this method can only be integrated in a slice based renderer such as (Schott et al., 2009; Šoltészová et al., 2010). Our DVR framework integrates depth-of-field in a unified way using stochastic sampling.

The previous approaches to photo-realistic rendering are often motivated by computational limitations of CPU, GPU and other dedicated hardware. For this reason, most of the work presents compelling, yet approximated, simulation of light transport. We expect that, with more powerful graphics hardware, these approximations will eventually be superseded by physically-based light transport modeling. To our knowledge, the work presented here is the first to demonstrate that interactive, brute force, progressive stochastic rendering for photo-realism in DVR is possible. Our method does not depend on pre-computed quantities or additional volumes and the memory footprint is insensitive to lighting, material, camera and transfer function configurations.

## 5.4 METHOD

At the highest level, our DVR framework continuously estimates the light contribution arriving at the camera film. As this estimate is an approximation, several estimates are needed to form a high quality image. As soon as the transfer function, lighting or camera position are changed, the Monte Carlo integrated estimate buffer is cleared and the MC algorithm starts from the beginning. Effectively, the image on screen initially appears briefly at a lower quality, but is progressively refined as more frame estimates are integrated. Interaction is instantaneous, as new frame estimates are generated rapidly. In the following subsections, we describe the DVR framework in more detail.

### 5.4.1 RENDERING PIPELINE

Figure 5.4 depicts the rendering pipeline, which is executed numerous times during progressive refinement. Rendering starts with the stochastic ray-caster, which is discussed in detail in Section 5.4.2. This process yields a High Dynamic Range (HDR) MC estimate of the light arriving at the film plane. This estimate is filtered with a separable Gaussian kernel, with standard deviation $1.5$ and window $5 \times 5$ pixels in order to reduce anti-aliasing. Next, Monte Carlo integration is performed by computing the cumulative moving average. The HDR MC estimate is then tone mapped to compress its dynamic range and subsequently gamma corrected. The Low Dynamic Range (LDR) image is then an-isotropically filtered to reduce noise, which is discussed in Section 5.4.3.

The following sections focus on how we implemented the stochastic ray-casting and the interactivity optimizations.

### 5.4.2 STOCHASTIC RAY CASTING

In contrast to standard ray-casting, in which camera rays originate from the camera origin and are cast through pixel centers on the screen, our method implements a thin lens camera model (Barsky et al., 2003), as shown in Figure 5.6. This model, which incorporates a finite aperture, models a real-world camera. The aperture size controls the depth of field, e.g., the range of distances in which objects appear to be sharp. As a result we are able to easily incorporate depth-of-field effects, which can be used

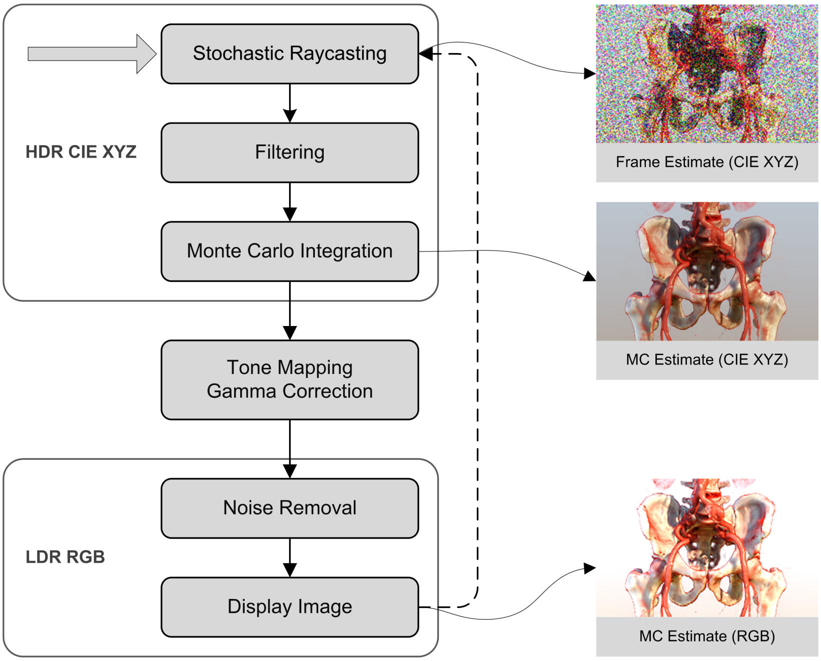


Figure 5.4: A high level overview of our rendering pipeline.

effectively in photo realistic DVR to draw attention to particular regions in an image. Figure 5.5 shows an example of this phenomenon.

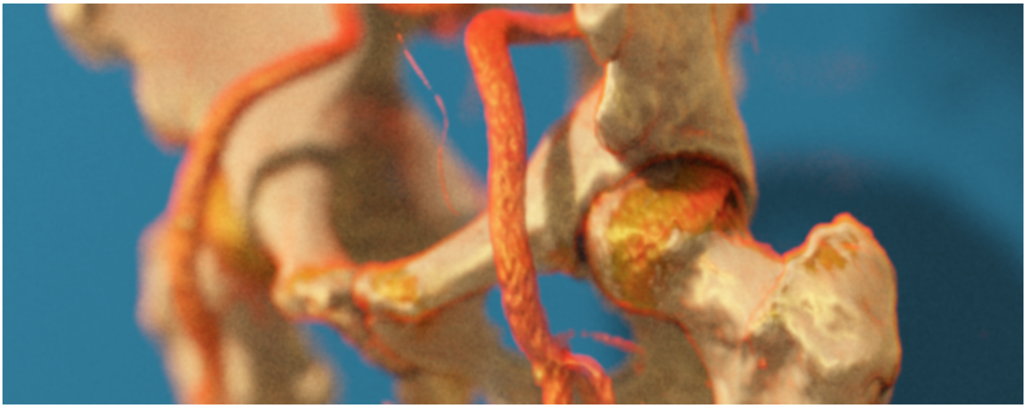


Figure 5.5: Depth-of-field rendering in Exposure Render.

In our implementation, camera rays are constructed by sampling a point on the lens and sampling a perturbed point within the finite area of a pixel on the film plane with stratified sampling. The camera parameters are stored in CUDA constant memory for fast access during ray generation. The rays are intersected with the volume's bounding box, yielding a parametric range $s$. Rays that do not intersect are discarded.

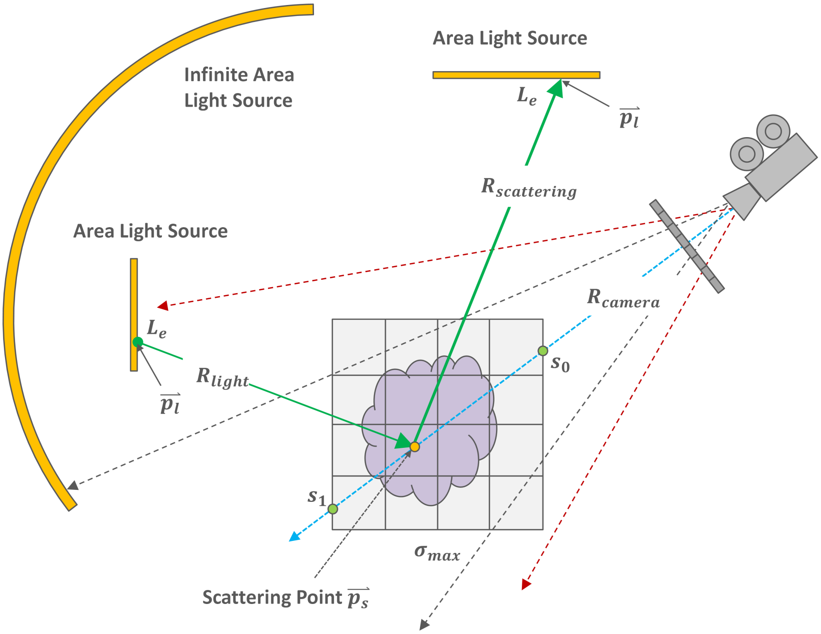


Figure 5.6: High level overview of stochastic ray-casting in our DVR framework . For every ray that is traversed, one scattering point is stochastically determined and the light contribution is computed using two additional rays.

Rays are treated differently in our DVR framework compared to other ray casters, which generally propagate through the volume with Ray Marching (RM). When shadow calculations are required at every sample position along the ray, RM becomes prohibitively slow. To address this problem, we use Woodcock tracking (Woodcock et al., 1965), which propagates through the volume with random length steps and yields a single scattering point $\vec{p_s}$. The Woodcock Tracking method determines where a scattering event occurs within the volume by: 1) generating a path length $s$ using the maximum extinction coefficient $\sigma_{max}$ in the volume, 2) accepting the tentative collision point with probability $\sigma_t(\vec{p}(s))/\sigma_{max}$ and 3) repeating this step until a collision is accepted, where $\sigma_t(\vec{p}(s))$ is the extinction coefficient at point $\vec{p}(s)$ in the volume.

We ultimately implemented Woodcock tracking, because it only yields a single scattering point within the volume, resulting in far fewer shadow computations per estimate. Furthermore, in contrast to ray marching, Woodcock tracking is unbiased. The performance of Woodcock tracking has been further optimized in (Szirmay-Kalos et al., 2011), but this extension has not been implemented yet in our framework, as this method requires an extinction volume to be recomputed for every transfer function change.

#### 5.4.2.1 DIRECT LIGHTING

Sampling direct lighting is not as straight-forward as with standard ray casting as our method also includes arbitrarily shaped lights and physically based phase and surface scattering functions, as shown in Figure 5.6. In order to reduce the sample variance in the Monte Carlo estimator, importance sampling is applied to the phase and surface scattering function for sampling ray directions, and the two samples, obtained by sampling lights and scattering functions, are combined with Multiple Importance Sampling (MIS) as discussed in the work by Veach (Veach, 1997).

In our implementation, the single scattering contribution at scattering point $\vec{p_s}$ is computed by:

1. Computing the contribution of light that flows along light ray $R_{\text{light}}$ which is formed by connecting the scattering point $\vec{p_s}$ with a stochastically sampled point on a stochastically chosen light source

2. Sampling the scattering function, yielding the scattering ray $R_{\text{scattering}}$. The intersection with a light is used to compute the contribution of light that flows along this ray.

3. Combining the two contributions using MIS by applying the power heuristic.

For the computation of light attenuation we also apply Woodcock tracking. In this case, the location of the scattering point $\vec{p_s'}$, yielded by Woodcock tracking, determines if a light ray is blocked or unblocked. If $\vec{p_s'}$ is beyond $\vec{p_s}$, it is unblocked, and vice versa. If the ray is unblocked, the light's exitant radiance $L_e$ is added. The lights are stored in CUDA constant memory for fast access.

#### 5.4.2.2 HYBRID SCATTERING

The challenge associated with applying MCRT to volumetric data is that there is no explicit boundary surface present, as with for instance surface meshes. MCRT algorithms that deal with participating media typically are unaware of shading normals, e.g., they use phase functions for computing volumetric scattering.

We present hybrid scattering, a technique that switches stochastically between surface and volumetric scattering based on the local gradient. Hybrid scattering aims to integrate surface and volumetric based scattering into stochastic volume rendering. As a result, volume renderings show detail through specular highlights in areas of well-defined gradient vectors, at the same time including volumetric scattering in areas of ill-defined gradient vectors. As a result the images yielded with this method are not only more vivid, but are also less susceptible to sample variance. The method switches probabilistically between surface and volumetric scattering based on the local voxel gradient magnitude, which is computed on the fly, and an additional weighting parameter controlling the amount of BRDF and volumetric scattering respectively. The gradient vector, used for BRDF calculations is computed on the fly with finite differences.

The method mixes surface and volumetric scattering by stochastically choosing whether a BRDF or phase function is used with the following probability:

$$P_{brdf} = \alpha_{\vec{x}} \cdot (1 - e^{-25 \cdot g^3 \cdot |\bigtriangledown \tau_n(\vec{x})|}) \tag{5.1}$$

$$P_{phase} = 1 - P_{brdf} \tag{5.2}$$

Where $\alpha_{\vec{x}}$ is the opacity at sample position $\vec{x}$, determined by the opacity transfer function. $|\bigtriangledown \tau_n(\vec{x})|$ is the normalized gradient magnitude at $\vec{x}$ and $g$ represents the *gradient factor*, a value ranging from [0-1] which controls the amount of surface/volumetric scattering. Hybrid scattering is integrated into our DVR framework by drawing a random number $\xi$ from a uniform distribution and comparing it to the BRDF probability $P_{brdf}$:

$$f(\vec{p_s}) = \begin{cases} f_r(x, \vec{\omega_o}, \vec{\omega_i}) & \xi < P_{brdf} \\ p(\omega_o \rightarrow \omega_i) & \text{otherwise} \end{cases} \tag{5.3}$$

In our implementation we use an isotropic phase function for volumetric scattering, but other anisotropic phase functions such as Schlick's phase function (Blasi et al., 1993), could be used as well. For surface scattering we implemented the Fresnel blend shading model as described in (Ashikmin et al., 2000). It models a diffuse surface with a glossy surface overlaid in a physically plausible way, and is reciprocal and energy conserving, unlike the conventional phong shading in standard ray casters. The fresnel blend BRDF blends between diffuse and glossy reflection by incorporating Schlick's approximation to Fresnel reflectance. The diffuse part is a straightforward Lambert shader, and the glossy part of the Fresnel blend is defined by a microfacet Blinn shader, developed by Blinn (Blinn, 1977). Furthermore, the BRDF is controlled by relatively simple parameters (e.g., diffuse refection, specular reflection and the Blinn exponent, which controls the size of the highlights). We implemented a one dimensional transfer function with the following channels: opacity, emission, diffuse and specular color, specular roughness and index of reflection. Each channel of the transfer function is stored in CUDA texture memory.

Though hybrid scattering is primarily designed to mix surface and volumetric scattering it can also be applied to data sets with lower signal to noise ratio by adjusting the gradient factor.

### 5.4.3 INTERACTIVITY

MCRT algorithms estimate the incident light on the camera film plane by sampling $N$ light transport paths and computing the estimate with:

$$I_i = \tfrac{1}{N} \sum_{j=1}^{N} \frac{f_i(C_j)}{p(X_j)}$$

The quality of this estimate is measured with variance, which is the square root of the expected error. As the number of samples $N$ increases, the expected error decreases with $O(\sqrt{N})$. The first few iterations of the MCRT algorithm yield high sample variance estimates, which results in images with high frequency noise, as shown in Figure 5.7. This noise is an undesirable side effect of MCRT and should be removed as soon as possible, in order to warrant interactivity. In the case of interactive volume rendering it is important that the image quality is high also during interaction. That is, the quality should be such that the user can still discern shape and important details. However, since every interaction triggers a completely restart of the MC algorithm unacceptable high frequency noise is induced.

In order to solve this problem we apply an anisotropic noise reduction filter to the final LDR MC estimate at every iteration, where the amount of noise reduction is dictated by the mean running sample variance of the LDR MC estimate. Previous work, aimed at suppressing noise from MC simulation of light transport has shown impressive results (McCool, 1999; Suykens and Willems, 2000; Xu and Pattanaik, 2005; Kontkanen et al., 2006). However, the performance of these algorithms is not real-time and thus not suitable for our purpose.

For our experiments, we implemented the *K Nearest Neighbor* (KNN) noise reduction filter. The KNN filter is a complex Gaussian blur filter in which the pixel weights are determined by the color similarity of neighboring pixels (Buades et al., 2006). The filter runs efficiently on CUDA enabled hardware, and is easy controlled. In our implementation we use a window size of $7 \times 7$ pixels. The *lerpC* parameter of the KNN filter blends between reconstructed and original pixels, and is in our implementation controlled by the mean running sample variance. As the sample variance decreases, so does the influence of the KNN filter. The mean sample variance of the LDR estimate is computed with a numerically stable algorithm, as described in (Knuth, 1997). All other parameters of the filter remain constant during rendering, for more details concerning the KNN filter configuration we refer to our open source implementation.

Figure 5.8 depicts the development of the LDR MC estimate over time with our noise reduction strategy applied. Note that especially in the first iteration, the noise is greatly reduced, at the expense of slightly increased blurring.

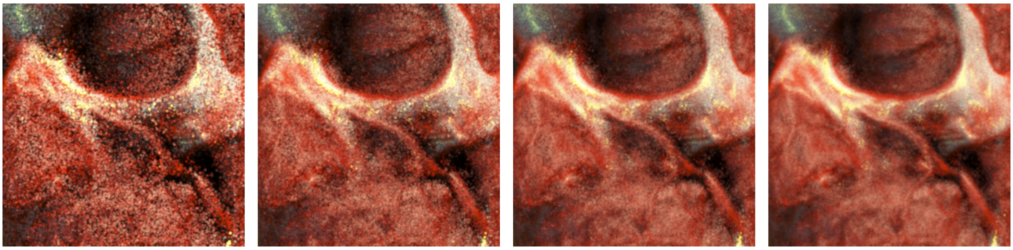


Figure 5.7: Image progression without noise reduction. The initial iterations show high sample variance.

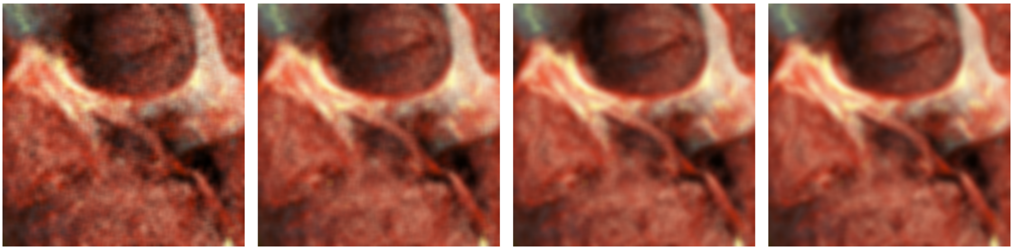


Figure 5.8: The same data set and configuration as in Figure 5.7, rendered with a noise reduction filter. By applying the noise reduction during the initial iterations, the objectionable noise at the startup of the MC algorithm is reduced to a great extent, at the expense of slightly increased blurring during the initial iterations. The influence of noise filtering, and thus the amount of blurring, reduces based on the error in the running estimate.

## 5.5 RESULTS

In this section, we show example renderings generated by our framework, briefly discuss its implementation, and present a number of performance benchmarks.

Example renderings are shown in Figure 5.9, Figure 5.3, Figure 5.5, Figure 5.10 and Figure 5.11. More example renderings, the implementation itself and demo movies can be found on the Exposure Render website (http://code.google.com/p/exposure-render/).

### 5.5.1 IMPLEMENTATION

The techniques discussed in this work are implemented Exposure Render, a complete interactive DVR framework, as shown in Figure 5.12, which has been made publicly available through a Google Code project: http://code.google.com/p/exposure-render/. The program runs on CUDA enabled graphics hardware, is written in C/C++, contains a full graphical user interface using Qt, and is shipped with sample volume

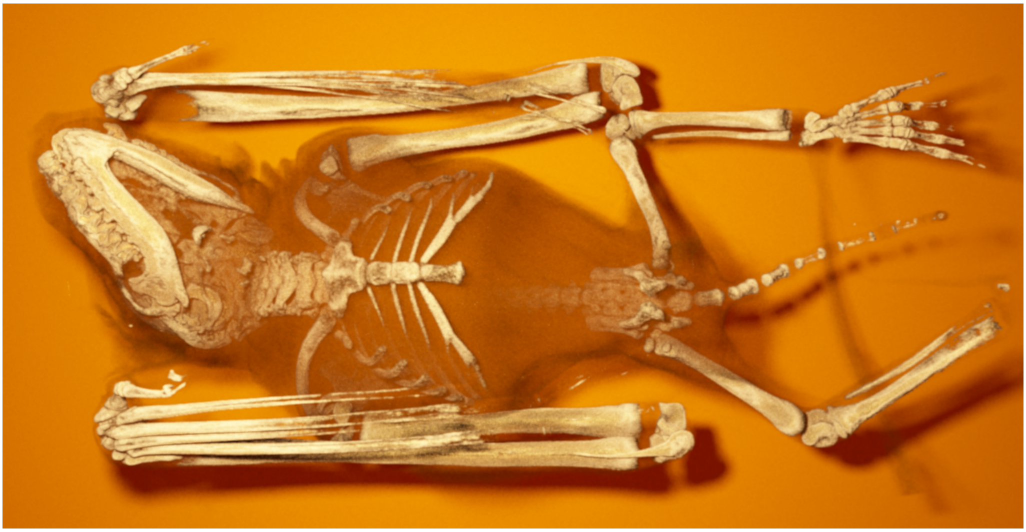


Figure 5.9: Volume rendering of a bat. Note the soft shadows on the plane behind the CT data caused by a planar area light.

data, transfer function, lighting and camera presets. At the time of this writing, the software has been downloaded more than four thousand times and it has been reported on by a number of popular graphics community websites.

### 5.5.2 PERFORMANCE

All experiments were performed on a PC with an NVIDIA GeForce GTX 470, with 877 MB of graphics memory, and an Intel ®Core$^{\text{TM}}$i7 CPU 920 with 12GB of RAM. In order to determine the performance of our method we have subjected our renderer to several benchmarks. First, we document the average number of iterations per second for various datasets and viewpoints, as shown in Table 5.1. We also investigate the convergence characteristics of the DVR framework in second part of our benchmark.

We measured the average number of estimates that our renderer generates per second for a film resolution of $800 \times 600$ pixels for five typical data sets from three different camera angles. Table 5.1 shows these results.

Unlike frame-based DVR, our method renders progressively. Convergence is therefore a good performance metric. We benchmark the convergence for five different data sets, varying aperture size and increasing number of lights. This gives a good indication of how robust our DVR framework is with respect to various effects.

We measured convergence by calculating the normalized root mean squared (NRMS), the error between the running MC estimate and a fully converged MC estimate. Figure 5.13 shows the results of these tests. From this, it can be seen that the image con-

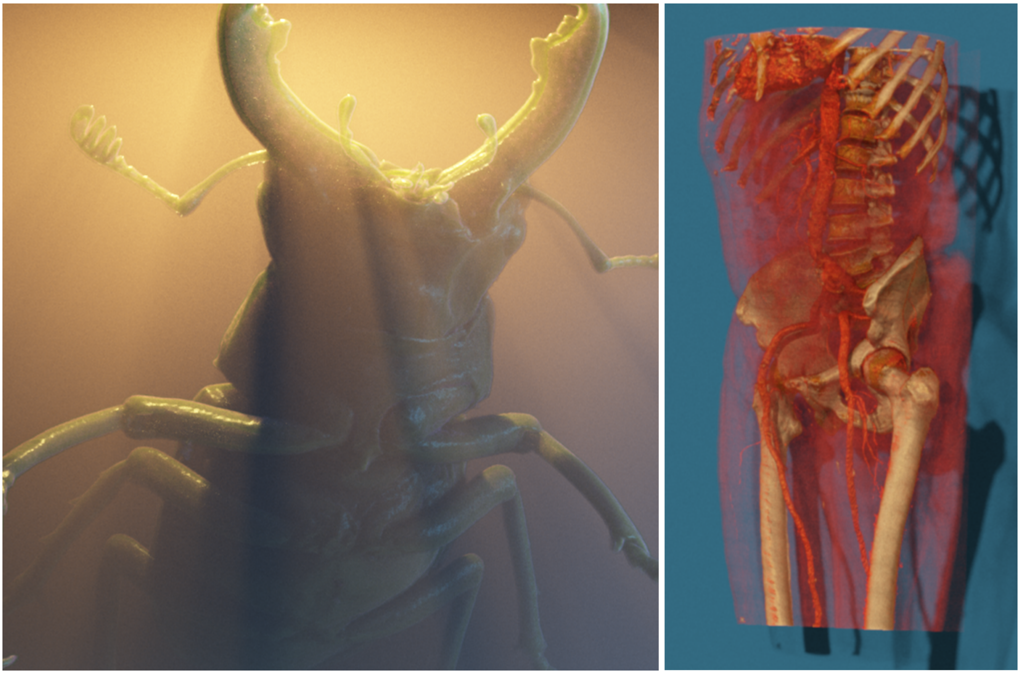


**Figure 5.10:** *Left:* Our method deals well with thin participating media. *Right:* Integration of physically based surface and volumetric scattering functions.

| Data set | Size | Front | Left | Top |
|---|---|---|---|---|
| Mecanix | 512 x 743 x 512 | 35.2 | 34.1 | 36.3 |
| Manix | 512 x 460 x 512 | 32.7 | 39.7 | 42.4 |
| Engine | 256 x 256 x 128 | 65.5 | 53.7 | 65.7 |
| Bonsai | 256 x 256 x 257 | 57.7 | 66.6 | 68.1 |
| Artifix | 512 x 347 x 512 | 47.3 | 40.3 | 50.8 |

**Table 5.1:** Performance measurements expressed in the number of frame estimates per second for five data sets, rendered at $800 \times 600$ pixels from a front, left and top view respectively. Scenes are lit with an environment light and two additional area lights. Al data sets are encoded in a (16 bit unsigned short) format

verges quite rapidly and that 10% NRMS is reached in a fraction of a second for all datasets.

Based on these benchmarks, we see that the renderer converges quite rapidly and is robust to different datasets, varying depth-of-field and increasing number of light sources. Importantly, the visualization remains interactive due to the progressive updates, yielding a high-quality volume rendering.

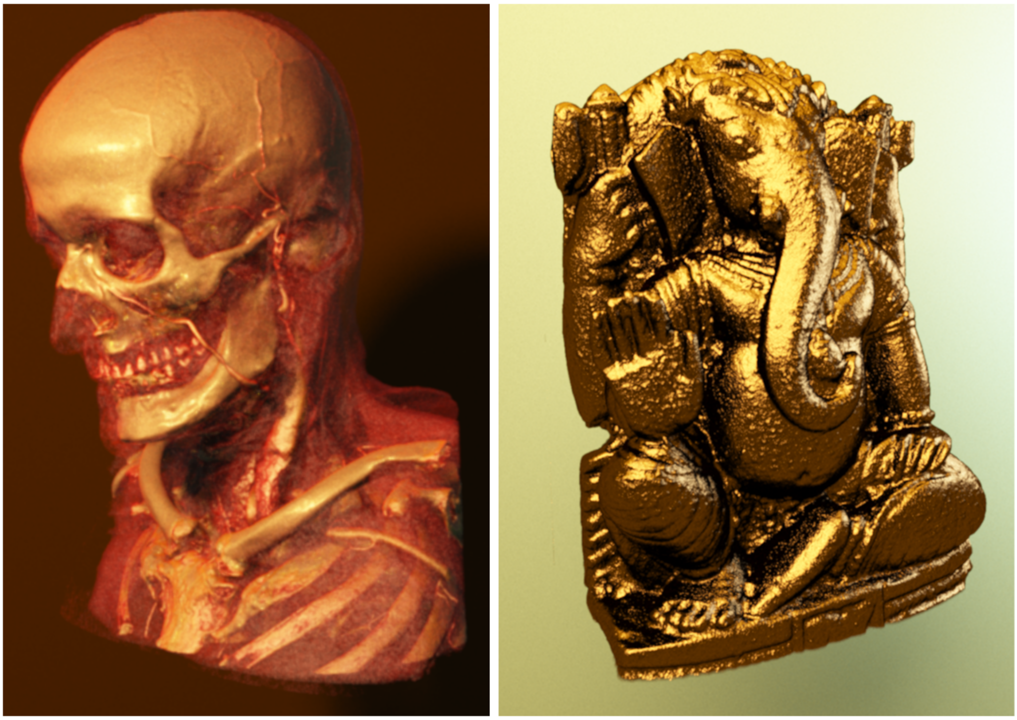


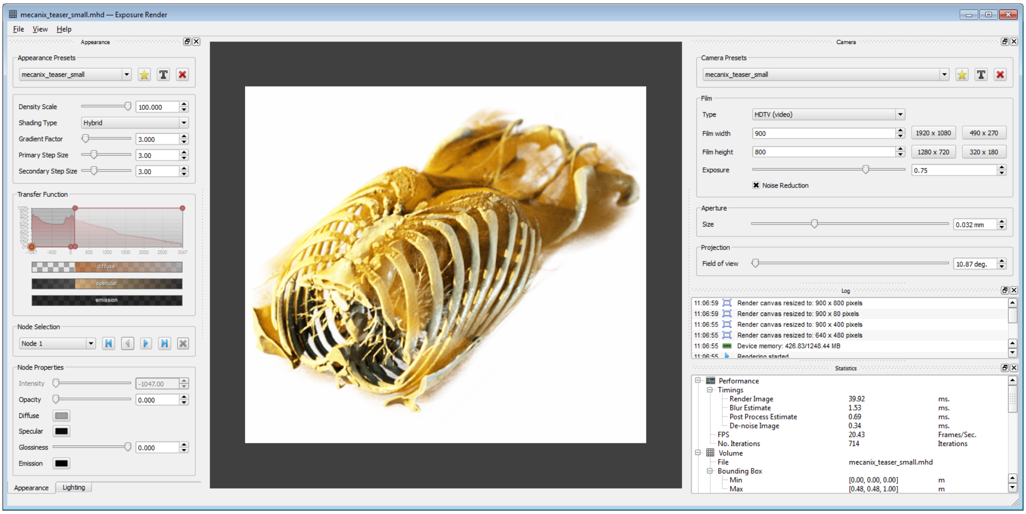
**Figure 5.11:** *Left:* Rendering of the publicly available Manix Data set (Osirix) with a photographic lighting setup. *Right:* By integrating physically based shaders into DVR, a wide spectrum of materials, for instance gold, can be simulated.

## 5.6 DISCUSSION

In this chapter we introduce a new DVR framework that integrates physically-based lighting in order to achieve photo realistic volume rendering images. We show that a number of effects, which are otherwise hard to obtain, such as realistic shadows, depth-of-field and realistic scattering, can be modeled in a unified way using our framework. Furthermore, we demonstrate that with the necessary optimizations, the brute-force ray-tracing of volumetric data using single scattering and efficient shadow sampling can be done interactively on commodity graphics hardware.

This combination enables physically-based volumetric shadows, any number of arbitrarily positioned, shaped and textured area lights and finally the simulation of a real-world camera, including its lens and aperture. With this setup, we are able to reproduce complex lighting setups that are currently used in photography studios, which adds an extra dimension of expressiveness to our volume visualizations.

In addition to the fact that photo-realistic volume renderings tend to be aesthetically more pleasing, it has been shown that realistic lighting contributes to 3D under-



Appearance Presets: mecanix_teaser_small

Density Scale: 100.000

Shading Type: Hybrid

Gradient Factor: 3.000

Primary Step Size: 3.00

Secondary Step Size: 3.00

Transfer Function

Node Selection: Node 1

Node Properties

Intensity: -1047.00

Opacity: 0.000

Diffuse

Specular

Glossiness: 0.000

Emission

Camera Presets: mecanix_teaser_small

Film

Type: HDTV (video)

Film width: 900

Film height: 800

Exposure: 0.75

☒ Noise Reduction

Aperture

Size: 0.032 mm

Projection

Field of view: 10.87 deg.

```
11:06:59  Render canvas resized to: 900 x 800 pixels
11:06:59  Render canvas resized to: 900 x 80 pixels
11:06:55  Render canvas resized to: 900 x 400 pixels
11:06:55  Render canvas resized to: 640 x 480 pixels
11:06:55  Device memory: 426.83/1248.44 MB
11:06:55  Rendering started
```

```
Performance
  Timings
    Render Image             39.92                     ms.
    Blur Estimate            1.53                      ms.
    Post Process Estimate    0.69                      ms.
    De-noise Image           0.34                      ms.
  FPS                        20.43                     Frames/Sec.
  No. Iterations             714                       Iterations
Volume
  File                       mecanix_teaser_small.mhd
  Bounding Box
    Min                      [0.00, 0.00, 0.00]        m
    Max                      [0.48, 0.48, 1.00]        m
```

Figure 5.12: Screenshot of the Exposure Render graphical user interface

standing and can improve depth-related task performance (Lindemann and Ropinski, 2011). With this work and the implementation that we have made available, we hope to contribute to the uptake of realistic illumination in interactive direct volume rendering applications. Although the rendered images are progressively refined, the refinement is already quite rapid on mid-level consumer graphics hardware. We expect that either by combining our technique with traditional GPU raycasting techniques in a level-of-detail approach, or through the expected advances in GPU hardware the coming months, the increased realism our work enables will find its use wherever high-quality interactive DVR is currently in place. Examples include radiological diagnosis of complicated 3D pathology, surgical planning, anatomy education and doctor-patient communication.

With this work, one of our goals was to investigate whether graphics hardware has become fast enough to enable the interactive simulation of physically-based light transport. This is part of a broader question on whether direct volume ray-tracing might soon replace direct volume rendering as the interactive volume visualization method of choice. The photo-realistic renderings our framework is able to generate, the added expressiveness and the measured and perceived interactivity of the visualizations, combined with the current trends in graphics hardware development, lead us to answer both questions positively.

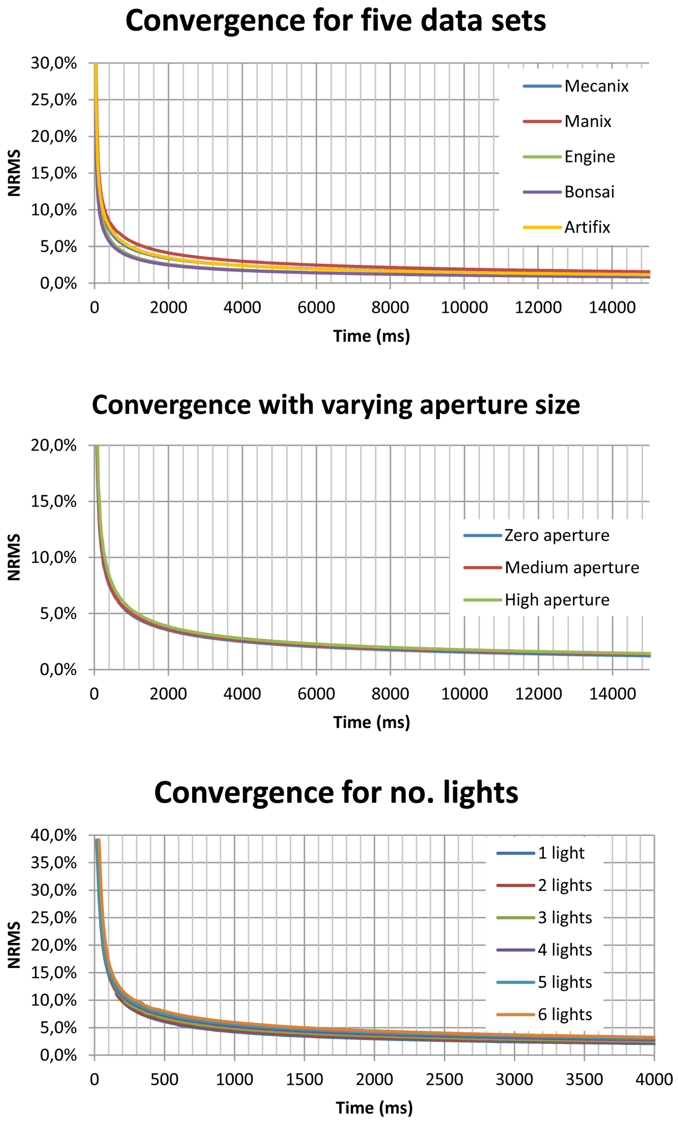


**Figure 5.13:** *Top:* Convergence characteristics for five typical data sets. *Middle:* Manix data set, with constant lighting and shading, subject to infinitely small aperture, medium aperture and large aperture respectively. *Bottom:* Backpack data set, with constant shading, camera parameters and increasing number of lights.

# GPU Accelerated Visibility Sampling for Medical Volume Data[1]

In the previous chapter, we described an interactive, photorealistic, volume rendering framework. This framework is based on progressive Monte Carlo raytracing, as a result, images are built up over time. The rate at which the image converges can be improved by incorporating additional knowledge about the scene in the ray sampling process (importance sampling). In this chapter, we investigate how we can efficiently compute and store visibility information, derived from volume data, using *visibility sweeps*. Further, we investigate how we can incorporate visibility information in the ray sampling process, in an attempt to improve the convergence rate of our volume rendering framework.

Even though we developed visibility sweeps in the context of stochastic volume rendering, our method can be considered general purpose, and as such is useful in other domains as well. In the context of computer-assisted surgery, visibility information is important in various stages of the pipeline. For instance, in order to expedite the exploration and interpretation of medical volume data, visibility can be used to perform automatic camera alignment. Further, visibility information can be exploited to enhance procedures that involve access (§ 2.2.3), resection (§ 2.2.4) and implant planning (§ 2.2.6), for instance to avoid high-risk structures such as blood vessels.

In the context of CSG-based knee replacement surgery, visibility information can be used to factor in the limited space in which surgeons can maneuver surgical instruments. During knee replacement surgery, the maneuverable space is limited, for

[1] T. Kroes, M. Eisemann and E. Eisemann. Presented at Graphics Interface 2015. Best student paper award.

instance due to the size of the incision, or the way that the patella is exposed. The optimization method described in Chapter 3 can be extended to take this into account by configuring the main docking direction such, that all docking directions inside the truncated cone[2] are unobstructed. In order to achieve this, the optimization needs a model of the surgical exposure, and the ability to efficiently query the visibility at an arbitrary position and direction.

[2] The truncated cone is used to model the uncertainty, which is associated with manual docking of the CSG

## 6.1 ABSTRACT

Physically-based light transport in heterogeneous volumetric data is computationally expensive because the rendering integral (particularly visibility) has to be stochastically solved. We present a visibility estimation method in concert with an importance-sampling technique for efficient and unbiased stochastic volume rendering. Our solution relies on a joint strategy, which involves the environmental illumination and visibility inside of the volume. A major contribution of our method is a fast sweeping-plane algorithm to progressively estimate partial occlusions at discrete locations, where we store the result using an octahedral representation. We then rely on a quadtree-based hierarchy to perform a joint importance sampling. Our technique is unbiased, requires little precomputation, is highly parallelizable, and is applicable to a variety of different volume data sets, dynamic transfer functions, and changing environmental lighting.

## 6.2 INTRODUCTION

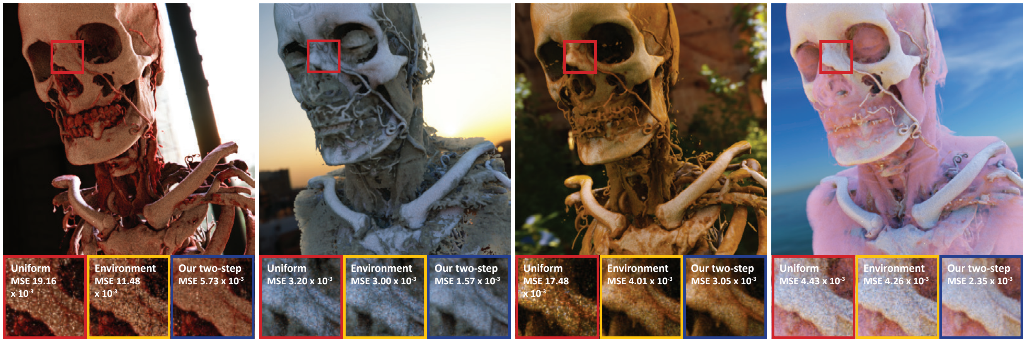


**Figure 6.1:** We compute the product of approximated visibility and environment map lighting in a stochastic Monte Carlo volume renderer to steer a joint importance sampling of the direct lighting. Our proposed two-step approach is well suited for dynamic changes in visibility and lighting functions due to a fast sweeping-plane algorithm to estimate visibility. The insets show how our technique (blue) achieves faster convergence with less samples compared to a uniform sampling (red) and importance sampling of the environment map (yellow). Here, 64 samples per pixel have been used. The Manix data set consists of $512 \times 512 \times 460$ voxels.

Stochastic volume rendering is computationally intensive. To evaluate the rendering equation, many samples (rays) are required in order to compute the light distribution within a volume. In practice, rays are sent from the camera through the volume and a scattering event occurs at random positions along the ray based on the current transfer function, which maps the volume's density values to material properties. Each scattering event requires generating one or more sample rays to evaluate the rendering equation via Monte Carlo (MC) integration. These rays are ultimately absorbed or potentially hit a light source, e.g., the environmental light. Using standard sampling techniques at the scattering events can be inefficient, as no knowledge about the volume absorption or light characteristics is used. As a result, many rays might contribute little or nothing to the final image.

Importance-sampling techniques Debevec (1998); Clarberg et al. (2005); Pharr and Humphreys (2010) incorporate knowledge about the scene to place more effort on potentially light-carrying paths to accelerate the convergence of the result. Some approaches combine information about the material and light positions. However, one important factor, the (volumetric) scene, and, hence, visibility is not taken into account. Previously, visibility approximations were only used directly in the shading evaluation, resulting in biased images Ritschel (2007). Furthermore, a brute-force visibility precomputation is costly and transfer-function changes require a complete reevaluation.

In this work, we focus specifically on evaluating direct lighting for a volume data set with arbitrary and interactively changing transfer functions defining varying diffuse materials in the context of an unbiased MC-based stochastic volume renderer. The volume is lit by a natural illumination in the form of environmental lighting.

The key idea of our approach is to use environmental light and visibility represented as a joint probability density function (pdf), as shown in Figure 6.2. As a result, the sampling process, steered by this pdf, becomes more efficient than in previous work, while keeping the result unbiased. The sampling technique allows us to evaluate the direct light at any scattering event within the volume. While our results could be generalized, we illustrate the application to single scattering.

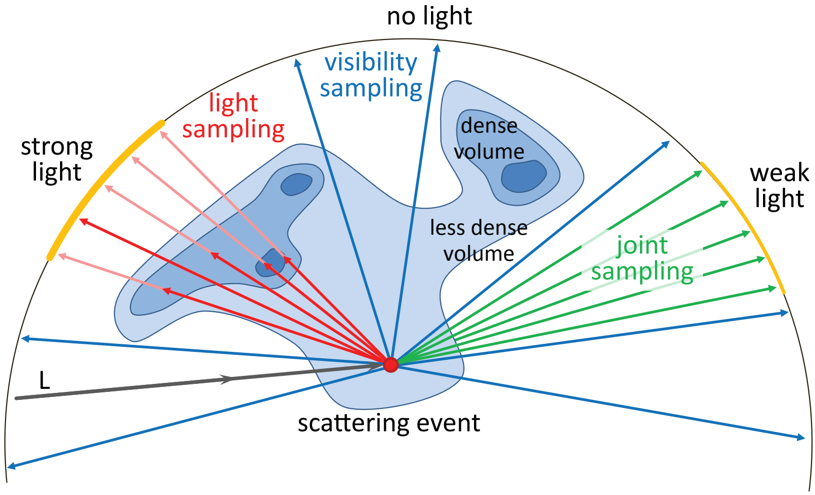


**Figure 6.2:** *Problem statement:* For efficient sampling, samples with both strong light and strong visibility need to be found. Sampling according to the lighting only (red) may give bad results as the samples get absorbed, sampling only according to the visibility (blue) might miss important lights. Product sampling (green) solves the problem. Unfortunately, the visibility is usually unknown beforehand.

Our contributions are as follows:

- An efficient sweeping-plane algorithm to compute approximate visibility within a 3D volume;
- A product importance sampling solution based on joint environmental light and visibility information;
- A GPU-adapted and highly-parallel implementation.

Our technique is useful for any volumetric renderer with dynamically changing content, such as environmental light, transfer functions, etc., making it an interesting addition to visualization and rendering systems aiming for unbiased results.

## 6.3 RELATED WORK

The literature on volumetric-illumination techniques is vast, which is why we will focus only on certain aspects to put our approach in perspective. A recent survey on this topic can be found in (Jnsson et al., 2014).

**Visibility Approximation for Semi-Transparent Structures** are most common in physically-based volume rendering. Opacity shadow maps (Kim and Neumann, 2001) are an extension of shadow maps (Williams, 1978) using a stack that stores alpha values instead of depth values to support shadow computation for complex, potentially semi-transparent structures. Deep shadow maps (Lokovic and Veach, 2000) are a more compact representation, which store a shadow-function approximation per pixel. They have quickly been adopted for volume rendering (Hadwiger et al., 2006; Ropinski et al., 2008b).

All such techniques are fast but inapplicable in our scenario of stochastic MC volume rendering. First, using approximate visibility directly for shading introduces a bias, which is unacceptable for certain applications. Second, these techniques support only point and directional light sources, whereas we aim for environmental lighting. Third, visibility is costly to compute and even approximating it can usually involve many rays, although not all locations might ultimately contribute to the image. Our approach computes visibility in a coarse 3D grid and uses it only to carefully steer the sample generation. In this way, our result remains unbiased, exact, and supports arbitrary environmental lighting.

**Basis-Function Techniques** decouple light-source radiance and visibility, which allows for dynamically changing the illumination. Spherical harmonics (SH) are prominent basis functions, used for example for pre-computed radiance transfer (Sloan et al., 2002), and were first used in the context of volume rendering to pre-compute and store isosurface illumination (Behrens and Ratering, 1998). They have also been used to store visibility for volume rendering under natural illumination (Ritschel, 2007). Other research in this area mostly aimed at generalizations to support advanced material properties (Lindemann and Ropinski, 2010) or reduce memory costs (Kronander et al., 2012).

While SH are well suited to represent low-frequency functions, their direct use for visibility is a strong approximation and introduces bias. Further, only low-frequency illumination is supported, in contrast to our solution.

**Image Plane-Sweep Volume Illumination Approaches** move a virtual plane through a scene to invoke the shading computations for all positions within the plane in parallel. The parallelism makes these approaches highly applicable to modern architectures, such as the GPU. Using carefully-chosen approximations (e.g., a forward peaked phase function, single point or directional light source), single and forward multiple scattering effects can be simulated at interactive frame rates (Sundén et al., 2011). We decouple the plane sweep from a particular light source to enable general illumination and efficient sampling in stochastic MC volume rendering.

Recently, iterative convolutions on volume slices have been used to approximate direct lighting (Patel et al., 2013). The results are approximate, some parameter settings have to be carefully chosen, and only particular light-source configurations are efficiently supported (e.g., usually Gaussian and behind the observer).

**MC Ray Tracing** for volume rendering gained attention with the advances of modern GPUs, which made interactive progressive rendering possible. First attempts sacrificed generality for performance (Rezk-Salama, 2007) and did not support translucent materials. New approaches, such as Exposure Render (Kroes et al., 2012) achieve images of very high realism. They employ all the benefits of physically-based MC techniques: arbitrary natural illumination, real-world cameras with lens and aperture (e.g., for depth-of-field effects). We implemented our approach building upon this open source solution. Only recently, specialized algorithms have been developed to efficiently handle participating media by splitting the evaluation into an analytical and a numerically evaluated part (Novák et al., 2014).

**Importance Sampling** is a powerful sampling technique to render objects illuminated by natural or complex lighting (Debevec, 1998) under an environmental illumination. An efficient method for non-specular materials is to place pre-integrated directional lights at the brightest locations (Agarwal et al., 2003; Kollig and Keller, 2003; Ostromoukhov et al., 2004). These methods work extremely well in the absence of occlusion, but shadowed regions may appear noisy. When materials are increasingly specular, a large number of lights is needed to adequately represent the environment map. Consequently, many physically-based MC techniques sample the environment map directly to avoid any artifacts and its intensity can even be used as a pdf to steer the sampling (Pharr and Humphreys, 2010).

If also visibility or material properties are to be included, the pdfs can be combined in a single MC estimator via multiple importance sampling (MIS) (Veach and Guibas, 1995). MIS is most efficient if only one of the sampled functions is complex and will pick the best one. If both are complex, MIS provides little advantage and is likely to waste samples in regions with little influence. Visibility and lighting can both be complex and only a joint sampling of both functions can be efficient, as shown in Figure 6.2. A first step toward this direction was taken in (Burke, 2004). Their technique importance samples the environment map to produce a candidate sample. Its probability is then evaluated again using a special pdf involving the BRDF to determine if an evaluation is triggered. Such a sampling can quickly become costly, due to potential high rejections rates (in the order of 90%) (Burke, 2004).

More related to our sampling approach are techniques for joint importance sampling that compute the BRDF/environment-map product (Clarberg et al., 2005; Cline et al., 2006; Clarberg and Akenine-Möller, 2008) and BRDF/visibility/environment-map product (Rousselle et al., 2008) to steer sample placement. In the context of participating media, joint importance sampling can also be employed to optimize volumetric paths (Georgiev et al., 2013). In this chapter, we focus on efficient visibility/environment-map sampling. Nonetheless, we also rely on a quadtree-based product to hierarchically warp samples (Clarberg and Akenine-Möller, 2008).

## 6.4 OVERVIEW

In the following, we will describe our algorithm in detail. First, we provide the necessary background knowledge (Sec. 6.4.1). Then, we describe our actual solution, starting with our data structures and data representations (Sec. 6.4.2), which are designed with GPU-efficiency in mind. Our visibility-sweep algorithm (Sec. 6.4.3) is used to compute an approximate visibility within the volume. It is then used in conjunction with the scene illumination to yield a joint sampling technique to steer the MC evaluation (Sec. 6.4.4). Finally, we describe the necessary implementation details (Sec. 6.4.5). The benefits for convergence behavior and the support of dynamic lighting and transfer-function changes will be demonstrated in Sec. 6.5.

### 6.4.1 BACKGROUND AND GOAL

We adopt the notation from (Ritschel, 2007) for the emission-absorption volume rendering equation (Max, 1995) in an isotropic medium:

$$L = \int_{0}^{\infty} A(t)E(\mathbf{x}(t))\mathrm{d}t. \tag{6.1}$$

It describes the recorded radiance $L$ along a camera-ray position $\mathbf{x}(t)$ parameterized by $t$, where

$$\begin{aligned} A(t) &= exp(-\int_{0}^{t} \tau_\alpha(D(\mathbf{x}(s)))\mathrm{d}s) & (6.2) \\ E(\mathbf{x}(t)) &= \int_{\Omega} \tau_\rho(D(\mathbf{x}(t)))V(\mathbf{x}(t),\omega)L_i(\omega)\mathrm{d}\omega. & (6.3) \end{aligned}$$

$E$ is the emission and $A(t)$ is the absorption up to position $\mathbf{x}(t)$. The volume density at location $\mathbf{x}(t)$ is denoted as $D(\mathbf{x}(t))$. The visibility at a position $\mathbf{x}(t)$ in direction $\omega$ is denoted as $V(\mathbf{x}(t),\omega)$. The incoming light $L_i(\omega)$ from direction $\omega$, integrated over all possible directions $\Omega$, is assumed to be independent of $\mathbf{x}(t)$, e.g., we assume an environmental light. A transfer function $\tau$ maps a density value $y$ to an extinction coefficient $\tau_\alpha(y)$ and scattering albedo $\tau_\rho(y)$. For brevity, we will omit the ray parameter $t$ and write only $\mathbf{x}$ to denote a certain location.

We use stochastic ray marching to solve the integral in Equation 6.1 and Equation 6.2 based on (Kroes et al., 2012). To solve Equation 6.3 stochastically, MC integration is applied:

$$E(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N} \frac{\tau_\rho(D(\mathbf{x}))V(\mathbf{x},\omega_j)L_i(\omega_j)}{p(\mathbf{x},\omega_j)}.$$

Here, $p$ is a pdf that is used to weigh and generate the random sample vectors $\omega_j$. The MC integration can become highly inefficient with a bad choice of the pdf $p$ as it may create many samples $\omega_j$ which contribute little to the final result, as shown in Figure 6.2.

The focus of this chapter is on choosing an effective pdf $p$ and its efficient computation. In order to achieve this, we split $p$ into two components

$$p(\mathbf{x},\omega) = \frac{1}{W(\mathbf{x})} p_V(\mathbf{x},\omega) p_{L_i}(\omega).$$

$p_V$ is a pdf based on the visibility, which changes locally throughout the volume based on the location $\mathbf{x}$, $p_{L_i}$ is a pdf based on the position-independent environmental lighting and $W(\mathbf{x}) = \int p_V(\mathbf{x},\omega)p_{L_i}(\omega)\mathrm{d}\omega$ is a normalization factor to produce a valid pdf. $p_{L_i}$ is known and based on the intensity of the environmental lighting, normalized by its overall intensity. The representation of these functions, the computation of $p_V(\mathbf{x},\omega)$, $p(\mathbf{x},\omega)$, and how to draw samples from $p(\mathbf{x},\omega)$ are the core of our method and explained in the following sections. We explain the data structures, then the visibility approximation, which will be used to derive $p_V$, before combining all the elements.

### 6.4.2 OCTAHEDRAL REPRESENTATION

Before explaining the algorithmic part of our approach, we will focus on the chosen data structures. They were developed to ensure an efficient evaluation on modern hardware and to simplify generation, sampling, and product computation. These elements will be necessary to drive the MC sampling process.

As we are dealing with potentially semi-transparent media in volume rendering, we assume $V$ to be locally smooth with respect to $\mathbf{x}$ and $\omega$. This allows us to estimate $V$ at discrete positions $\mathbf{x}_d$ and a few discrete directions $\omega_d$. We arrange the locations within a 3D voxel grid of user-defined size encompassing the original volume. These local estimates are then interpolated during rendering to obtain an approximation of the actual visibility in each location.

For a fixed location $\mathbf{x}$ our functions $V$ and $L_i$ and their respective pdfs $p_V$ and $p_{L_i}$ are spherical functions, e.g., they solely depend on a direction vector $\omega$. Given that we only consider piecewise-constant pdfs, we will represent these functions as octahedral map's, which is a discrete image-based area-preserving representation (Praun and Hoppe, 2003) and can be saved/accessed as a 2D texture, as shown in Figure 6.3. Each texel $p$ in these maps is associated with one direction $\omega_d(p)$ and indicates the accumulated volumetric visibility in direction $\omega_d(p)$ from the maps location. We will refer to these visibility maps for each discrete location $\mathbf{x}_d$ as *visibility voxels* $\mathbf{V}_d$.

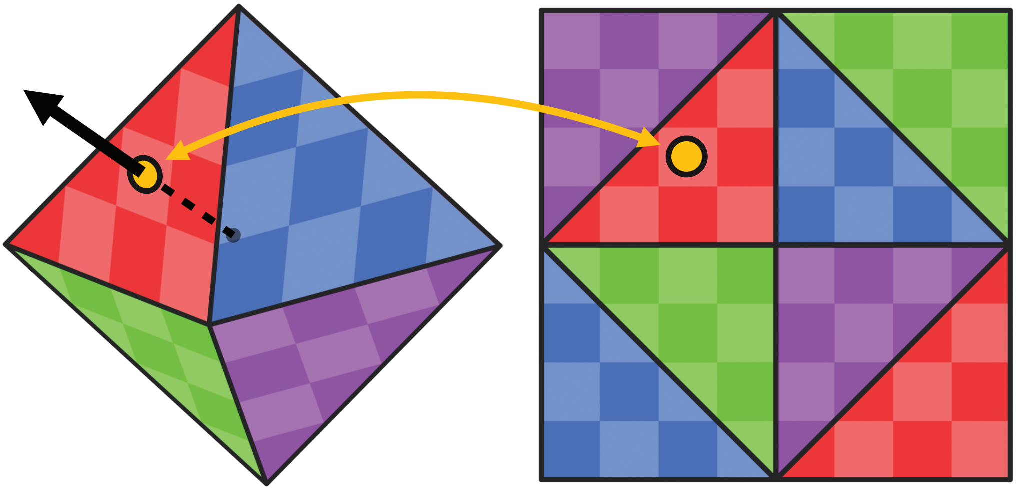


Figure 6.3: *Octahedral representation:* We present spherical functions using an octahedral representation. *Left:* 3D representation. *Right:* unfolded 2D representation.

### 6.4.3 VISIBILITY APPROXIMATION

In this section, we describe how to compute the entries of the visibility voxels via our sweeping plane algorithm. The visibility is computed for one direction $\omega_d(p)$ at the time. In each step, one slice of $\mathbf{V}_d$ is evaluated in parallel. Previous results are reused, making only a few value lookups per step necessary. Therefore, the amortized cost over all $\mathbf{x}_d$ is very low. After all directions were treated, the resulting $\mathbf{V}_d$ is used to derive the pdf $p_V$, which will guide the sampling process. An illustration of a single sweep step is given in Figure 6.4 and described in the following.

Here, we describe the process for one given direction $\omega_d$, for brevity we omit the direction parameter in brackets. First, a plane $\mathcal{P}_d$ with normal $\omega_d$ is defined, the orthogonal projection of the data volume's bounding box defines its size. We then create a set of $r$ rays at uniformly distributed positions within this projection having the same direction as $\omega_d$. To coordinate the ray traversal, we introduce a sweeping plane $\mathcal{S}_d$ which is orthogonal to one of the main axes of the original volume data. This axis is chosen based on the main direction of $\omega_d$ (defined as the maximum of the absolute values of its three components). $\mathcal{S}_d$ is initialized to intersect the first 2D slice of $\mathbf{V}_d$, so that it coincides with the position of the visibility voxels within this slice. We traverse the volume along the rays starting at $\mathcal{P}$ and accumulate visibility changes until they

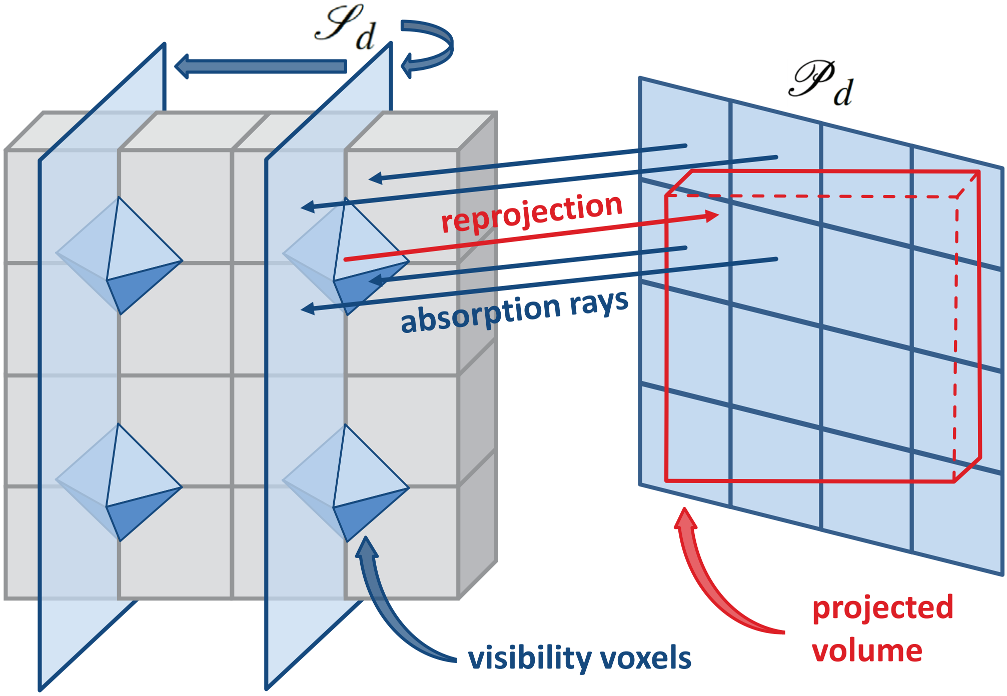


**Figure 6.4:** *Visibility sweeps:* We compute the absorption of sample rays starting at plane $n$ along direction $\omega_d$ up to the positions coinciding with a sweeping plane, which is orthogonal to the main component of $\omega_d$. To compute the absorption at a visibility voxel in direction $-\omega_d$ we reproject its position onto $\mathcal{P}_d$ and query the interpolated absorption value. All components can be efficiently computed on the GPU.

hit $\mathcal{S}$. This accumulation effectively keeps track of all relevant information that lies behind the ray when reaching a new visibility voxel.

The main loop of the algorithm moves $\mathcal{S}_d$ forward along the main direction by one visibility-voxel slice at a time until all slices are processed. After each step of $\mathcal{S}_d$, the rays advance via ray marching from their previous position until they reach $\mathcal{S}_d$ again. On their way, the absorption values along the ray are accumulated and, when reaching $\mathcal{S}_d$, stored in a 2D texture mapped onto the initial positions on $\mathcal{P}_d$. Please note that the resolution of the visibility voxel grid and the original volume can be different. Next, the visibility voxels coinciding with $\mathcal{S}_d$ (now reached by the rays) are updated by querying the interpolated absorption values produced by the rays. This gathering operation is highly parallelizable and more efficient than a scattering strategy.

After the algorithm finishes and all directions have been processed, we have a discrete approximation $\mathbf{V}_d$ of the visibility within the volume, which, if normalized, results in the pdf $p_V$. We add a small $\epsilon$-value beforehand to prevent zero probabilities, which would introduce a bias. The main observation is that this iterative update is more efficient than individual visibility computations per visibility voxel.

### 6.4.4 JOINT IMPORTANCE SAMPLING

At a scattering event during rendering, we want to make use of a joint importance sampling combining visibility and environmental lighting. We have explained how to produce the pdfs for $p_V$ and $p_{L_i}$. Here, we explain how to combine both. The computation is divided into a preprocess, taking place whenever the environment map or the transfer function changes, and an online process, taking place whenever a scattering event occurs during rendering.

**Preprocess** For the preprocess, we assume that the environment map is also given as an octahedral map, otherwise we convert it first. As a reminder, $p_{L_i}$ is defined as the normalized intensity value of the environmental lighting, giving higher importance to the brighter parts. In general, the resolution of the octahedral maps of $p_{L_i}$ will be higher than for $p_V$. To combine both, we first adapt the resolution of $p_{L_i}$. To simplify explanations, we assume that the resolution in width and height is chosen to be a power of two.

Similar to (Clarberg and Akenine-Möller, 2008), we create a multiresolution pdf in the form of a quadtree, e.g., each node saves the average of its four child nodes, with the leaves being the individual pixels. To match the resolution between lighting and visibility, we choose a level $l$ in $p_{L_i}$ whose resolution is equal to the directional resolution of a single visibility voxel. We then multiply all entries in $\mathbf{V}_d$ with the respective information in $p_{L_i}$ at level $l$. The result is an unnormalized joint pdf of the combined product.

**Rendering** In the rendering phase, we create a final combined pdf $p$ for each scattering event at location $\mathbf{x}$. This pdf is used to draw a single sample, as this strategy is often more efficient in a stochastic volume renderer with semi-transparent media (Kroes et al., 2012). Nonetheless, the sampling algorithm naturally extends to any number and distribution of initial samples, including quasi-MC methods (Niederreiter, 1992).

To derive the pdf $p$, we first linearly interpolate the neighboring visibility voxels which now carry the information of both visibility and lighting as described in the preprocess. Initially, this interpolated result is not a pdf. Nevertheless, we do not normalize it right away, but compute a multiresolution representation in the form of a quadtree where each node is the average of its child nodes. Following the hierarchical warping technique (Clarberg et al., 2005), we can then transform a uniformly distributed $[0, 1)^2$-variable into one that is distributed according to $p$ by passing the sample down in the quadtree according to the local probabilities. In contrast to (Clarberg et al., 2005), we need to normalize each $2 \times 2$ tile that we encounter during the quadtree sampling but as we only draw a single sample per scattering event the effort is only $O(\log n)$ compared to $O(n)$ if we would create a complete pdf for the interpolated visibility voxel. Here, $n$ is the number of texels in the lowest level of the quadtree.

In case the environment map has a high resolution, we propose to use a *two-step approach* that continues the descent on the remaining quadtree of the higher resolved environment map (Cline et al., 2006; Clarberg and Akenine-Möller, 2008). This step is especially beneficial for complex high-frequency illumination, which is otherwise not well taken into account during the sampling.

### 6.4.5 IMPLEMENTATION DETAILS

We found that using $8 \times 8$ maps to represent the visibility at each $\mathbf{x}_d$ is generally a good memory/performance trade-off. The amount of visibility voxels should be based on the scene/feature size and $\mathbf{V}_d$ should be chosen slightly larger to encompass the original volume and ensure a correct boundary sampling. The number of ray-marching steps along the ray should be based on the resolution of the data set and the number of visibility voxels. The maximal step size should be equal to the voxel size in the original data set in order to not miss any details.

For improved cache usage, we actually do not compute all visibility-voxel octahedral maps in advance. Instead, we avoid that rays write to different textures during the ray traversal in the visibility precomputation and store the occlusion values for a direction $\omega_d$ in a separate 3D texture. Once the pass for $\omega_d$ is completed, we perform the multiplication with the environment map as explained above and keep this texture in memory. During the rendering process, when a scattering event occurs, it might not lie directly on a visibility voxel. We thus retrieve the interpolated values using hardware filtering from these 3D textures and construct the visibility-voxel octahedral map on the fly. Although this might sound costly, it turned out that in practice, this cost is outweighed by the cache advantages due to a better data locality and we avoid constructing visibility voxels in areas where no scattering occurs. Nonetheless, on future hardware a first reconstruction pass might become preferable.

## 6.5 RESULTS

We integrated our algorithms into the stochastic CUDA-based volume renderer from (Kroes et al., 2012). All images have been generated on a 64bit Intel© Core$^{\mathrm{TM}}$ i7 920 with $2.67$GHz, 12GB of RAM, and an NVIDIA GeForce GTX 760.

We compare the performance and do a qualitative comparison between the existing and our approach. We compute the mean-squared error (MSE) and compared to reference solutions using $8196$ samples per pixel with uniform sampling. Our environment maps all had a resolution of $2048 \times 2048$ pixels (in the octahedral representation). For all tests, the joint importance pdf $p$ was constructed on the fly for each scattering event via interpolation of eight visibility voxels.

| Vis. Voxels | 256x256x230 | 128x128x115 | 64x64x57 | 32x32x28 |
|---|---|---|---|---|
| Memory | 964.7 | 120.6 | 15.0 | 1.8 |
| Sweep ($16^2$) | 3.76 | 1.93 | 0.76 | 0.53 |
| Sweep ($32^2$) | 3.89 | 1.97 | 0.79 | 0.54 |
| Sweep ($64^2$) | 4.05 | 2.03 | 0.79 | 0.55 |
| Sweep ($128^2$) | 4.31 | 2.40 | 1.04 | 0.59 |
| Sweep ($256^2$) | 6.36 | 3.23 | 1.63 | 1.41 |
| Brute-Force | 97.35 | 15.17 | 2.79 | 0.57 |

Table 6.1: Memory requirements (MB) and timings (seconds) for the visibility sweep algorithm and varying input parameters in comparison to a brute-force visibility computation. We shoot $16^2$, $32^2$, $64^2$, $128^2$ and $256^2$ absorption rays per sweeping direction. All experiments are performed on the Manix data set ($512 \times 512 \times 460$ voxels).

**Timings and Parameters** The overhead during rendering using our visibility sweeps is low compared to the gain in quality, especially as the sweeping-plane algorithm to update the visibilities in $\mathbf{V}_d$ is evoked only if the transfer function or illumination changes. As standard parameters, we use a $8^2$ directional map for each visibility voxel and set one visibility voxel for each $4^3$ voxel subset of the original data volume. The overhead during rendering is roughly only 10%, compared to rendering the same number of samples per pixel using plain uniform sampling. This includes interpolating the visibilities, creating the multiresolution 2D pdf representation on the fly and the joint importance sampling itself.

We compared our visibility sweeps approach to a brute-force computation of the visibility where each entry for the visibility voxels is computed exactly using ray marching. Table 6.1 shows a comparison of the timings for different parameters. For our proposed standard parameters and a reasonable number of absorption rays our approach is approximately $6\times$ faster than the brute-force computation. It is important to note that this factor becomes larger with an increasing number of visibility voxels (up to a factor of 15 in our tests in Table 6.1). Further, the test scene (Manix) resembles and isosurface due to its very steep transfer function, hence, the brute-force ray marching stops if the ray hits the isosurface. In our sweeping-plane algorithm the rays need to traverse the whole volume. So, we deliberately chose a difficult scenario - the benefit will be even bigger for a higher number of visibility voxels and more transparent volumes.

Additionally, we checked our assumption that we can interpolate the queried visibility during the reprojection step in the sweeping-plane algorithm. It should be pointed out that the results will always remain unbiased, independent of the resolution because the values are only used to guide the sampling process. To this extent, we reduced the numbers of absorption rays that are traversed through the volume shown in Figure 6.5, which is of size $512 \times 512 \times 373$ voxels. Consequently, the visibility voxels will have to rely on an interpolated result. The number of rays influences the result only slightly and no visible errors are introduced. This result indicates that we can rely on a relatively cheap preprocess to approximate the visibility, which reduces the additional overhead in our approach.

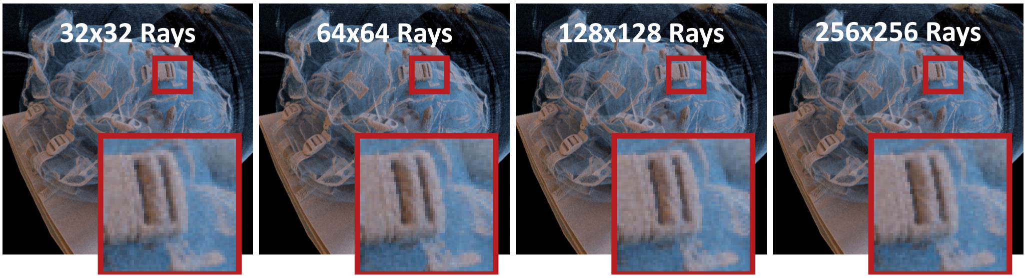


Figure 6.5: Influence of the visibility sampling precision (number of absorption rays) on the result.

**Qualitative Evaluation** We compare our approach to uniform sampling, importance sampling of the environment map only, importance sampling of the visibility only, a combined approach, where the visibility pdf is multiplied with the downsampled pdf from the environment map of the same resolution, as well as our combined two-step approach, which makes use of the combined sampling but switches to the full environment-map resolution as soon as a leaf in the combined pdf representation is reached to further support high-frequency lighting.

Figure 6.6 shows an equal-time comparison of all the techniques after 10 s render time, excluding the visibility precomputation. For comparison, we show the computed number of samples per pixel (SPP) and the Mean-Squared Error (MSE) for each approach. Though the number of samples is lower, due to the computational overhead induced by the joint sampling, the noise is significantly reduced with our approach. Due to a lower ray coherency the uniform sampling creates less samples per pixel in the same time than most of the other approaches.

Figure 6.7 and Figure 6.1 show an equal sample comparison. Figure 6.8 shows an equal quality comparison, where we precomputed the result for various power-of-two number of samples and illustrate the ones closest to the indicated error.

Additionally, we provide error plots for the Statue and Engine Block scene with respect to the number of samples in Figure 6.9 and Figure 6.10. As expected, uniform sampling performs worst. Interestingly, the visibility sampling alone performs better in the beginning but has a worse convergence. We found that the images also contain a lot more firefly artifacts. We, therefore, believe the reason for the convergence behavior lies in the approximate visibility function at occlusion boundaries. If a direction is supposed to be occluded it is sampled with a low probability and therefore a high

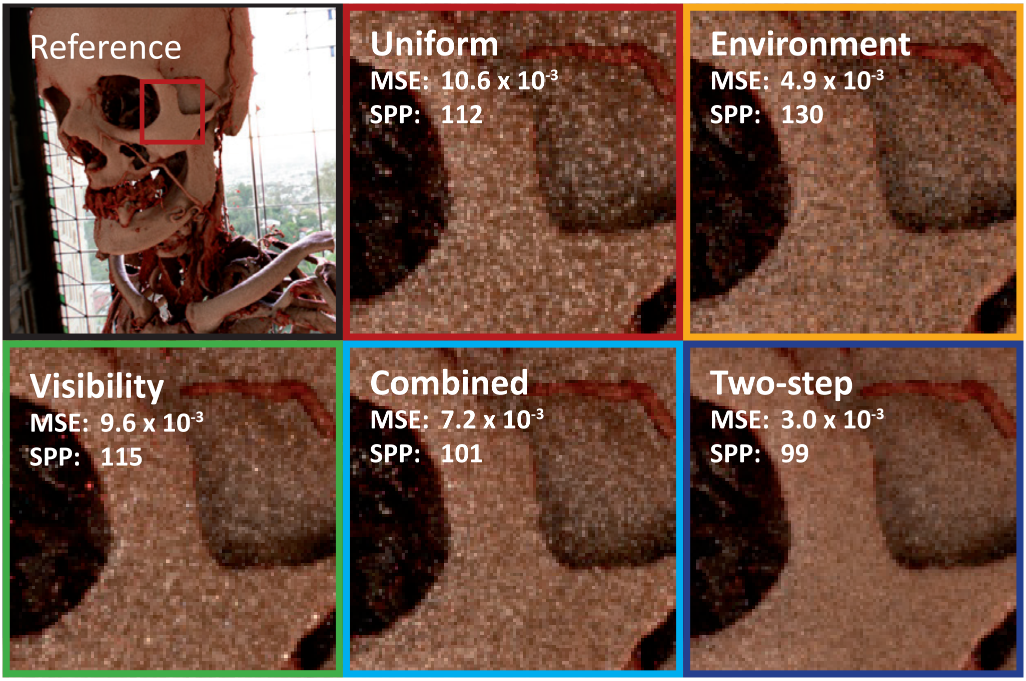


**Figure 6.6:** *Equal time comparison:* All images, except the reference image, have been created using 10 seconds of rendering time.

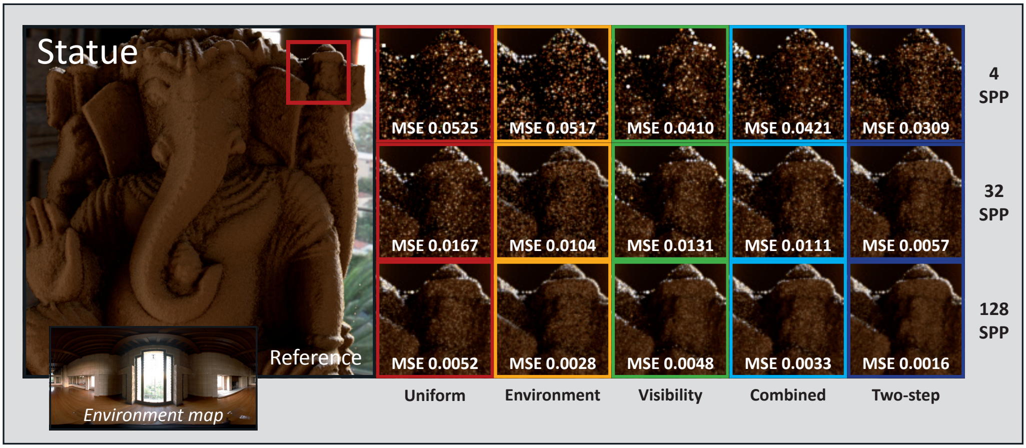


**Figure 6.7:** *Equal sample comparison:* We compare our proposed two step importance sampling technique (dark blue) using 4, 32 and 128 samples to uniform sampling (red) and importance sampling of the environment map only (yellow), the visibility only (green), and the combined low-resolution product (light blue). All images are unbiased and a reference, as well as the environment map are shown on the left.

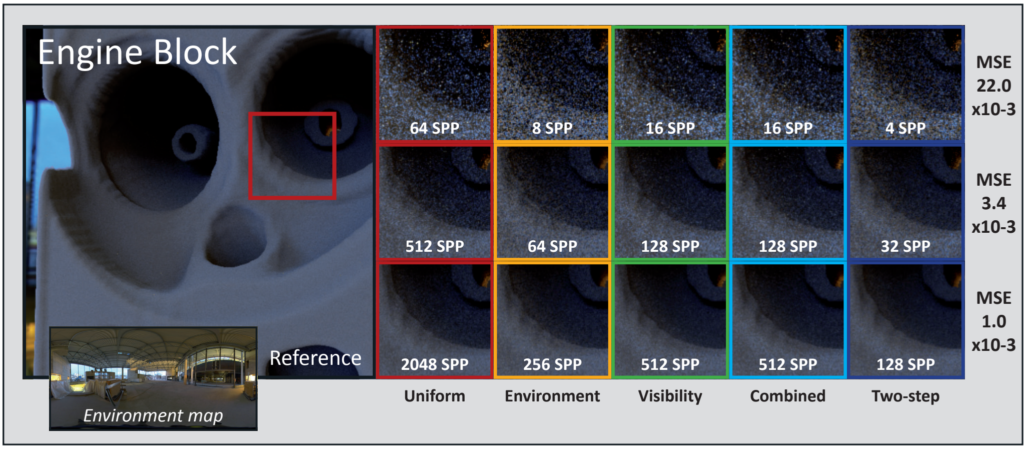


**Figure 6.8:** *Equal quality comparison:* We compare our proposed two step importance sampling technique (dark blue) to uniform sampling (red) and importance sampling of the environment map only (yellow), the visibility only (green), and the combined low-resolution product (light blue). All images are unbiased and a reference, as well as the environment map are shown on the left. For approximately the same quality, our two-step approach requires significantly less samples.

weighting. If it accidentally hits a bright light source behind it, high energy samples are added to the result which result in the fireflies. As the direction around this occlusion boundary is rarely sampled it takes a lot of samples to correct for these errors. If the lighting is incorporated in the pdf, these cases are taken care of sufficiently. Environment map and the low-resolution combined sampling perform almost equally well on the Statue scene. Presumably, importance sampling the light wastes a lot of samples that are absorbed within the volume. The combined sampling approach suffers to some extent from the low resolution of the visibility function and, therefore, the combined pdf is not able to capture the high frequency details of the environment map. This disadvantage is compensated by the two-step approach, which can make use of both the visibility and the high-frequency illumination information and shows better convergence rates even at high sampling rates. The results suggest that it is highly beneficial to incorporate the proposed visibility sweeps and joint sampling in the two-step approach for stochastic MC volume rendering.

## 6.6 CONCLUSIONS

We presented a joint sampling approach relying on visibility and lighting information within an interactive unbiased stochastic volume renderer. The core of our solution is an efficiently-computed visibility approximation based on a sweep-plane algorithm. Its performance allows us to change environmental lighting and transfer functions dy-

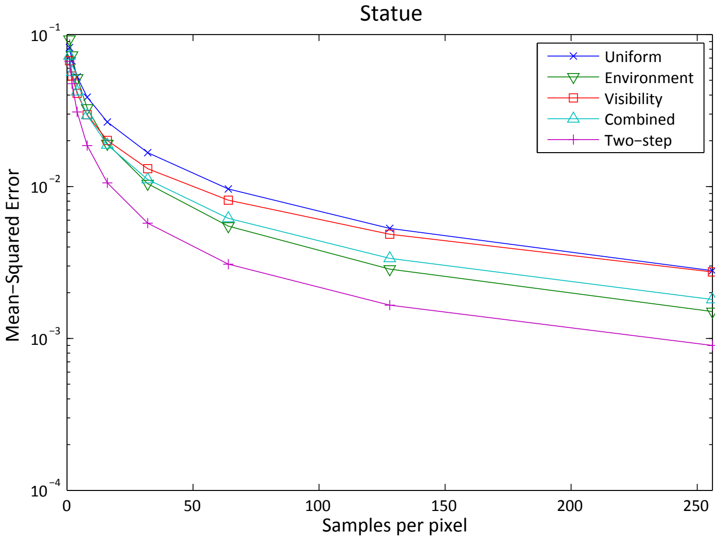


**Figure 6.9:** Convergence graphs for the Statue scene (Figure 6.7)

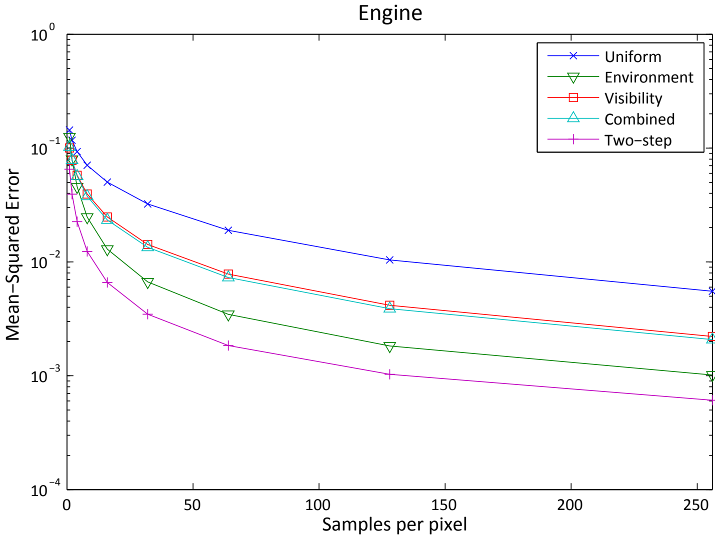


**Figure 6.10:** Convergence graphs for the Engine Block scene (Figure 6.8).

namically. We carefully designed our algorithm for GPU execution and have demonstrated its applicability to different volume data sets.

Visibility sweeps could prove beneficial for traditional boundary-representation rendering as well. To some extent this is illustrated by using transfer functions, which lead to very sharp features. Our approach usually lowers the amount of needed samples significantly compared to previous solutions at equal quality, which is an important result as the evaluation of samples is a very costly element in most production and rendering contexts.

There are still some options for minor optimizations for the traversal algorithm. First, a pruning of the absorption rays that do not intersect with the volume at all, and second, an early exit strategy for rays that are already fully absorbed, could potentially result in a traversal speed-up at the cost of a more complex algorithm. Though not yet implemented, interactive clipping (slicing) of the volume is naturally supported in our approach, as it simply requires disregarding the intensity values in front of the slicing plane during the visibility computation. A remaining challenge is the incorporation of non-diffuse media. For isotropic media one could precompute several pdfs based on the angle of the incoming and outgoing ray and interpolate these during rendering, but general reflection models remain future work.

# General Discussion

The goal of this thesis is to develop novel planning tools for CSG-based surgery. The previous chapters document our efforts toward achieving this goal. In the following sections, we look back at the work, discuss our contributions, put our results into context, and identify exciting avenues for new research.

## 7.1 VISUALIZATION IN COMPUTER ASSISTED SURGERY

Even though the infiltration of medical visualization in the field of CAS is wide-spread, and its importance undisputed, there is no review to date that studies the CAS landscape in its entirety, from different surgical, visualization, and application specific perspectives. There are however more general CAS reviews (Joskowicz and Taylor, 2001; Taylor and Joskowicz, 2003; Cinquin et al., 1995; Taylor et al., 1996; Satava, 1999), and application specific reviews, for instance concerning the use of visualization in image guided surgery (IGS) (Holmes III et al., 2008; Stoyanov et al., 2003; Kersten-Oertel et al., 2010). Our observation that a multi-faceted and visualization-oriented review of CAS was lacking ignited an exhaustive literature search, which resulted in a comprehensive database of over 500 visualization-oriented CAS publications. From these publications, we selected the publications with the biggest impact and most relevance for a deeper discussion in Chapter 2. With this chapter, our contribution to the field of medical visualization is twofold. Our first contribution is that we gathered relevant CAS visualization literature, irrespective of the CAS application area. Nearly all surveys that we came across were application-specific. Our second contribution is that we provided four intuitive and meaningful classifications of our literature database e.g. visualization techniques, surgical tasks, transfer methods and application area.

We gave an overview of the techniques that are used in CAS, and discussed the role of visualization in surgical tasks. We also investigated how visualization is used in four prominent application areas, and how surgical plans are transferred to the operating theater. To date, this type of analysis has not been seen in CAS review literature.

## 7.2 PREOPERATIVE PLANNING OF CSG-BASED SURGERY

Optimizing CSGs, such that their fit is optimized, is a challenging task. This was already recognized in (Haselbacher et al., 2012), in which a pin-based CSG for TKA is described. In their approach, the distribution of pins is determined manually in the planning stage. They performed experiments with a prototype of a pin-based CSG on anatomical specimens of the distal femur. Even though the measured angles are similar to other computer-assisted approaches, there is a considerable proximal-distal shift. Most likely, the manual configuration step is responsible for this significant translation error, which is further supported by the results of similar experiments that we performed with a wide variety of manually configured pin-based CSG and rapid prototyped bones. It turns out that manual configuration with a limited number of pins, and restricted pin-deployment region, nearly always leads to malaligned CSGs and poorly reproducible results.

These observations compelled us to solve the problem of CSG optimization in silico. Even though the problem of static docking stability can be partially solved analytically (Mattheijer et al., 2013, 2015), we are also interested in the dynamic interaction between the CSG, the bone, and other tissues. In order to achieve this, we combined a physical simulator, that models the physical interaction between CSG and patient, with a genetic optimization. We evaluated our approach in the real world and demonstrated that, given an exemplary CSG design and various bone geometries, we are able to reach high accuracy and reproducible results.

Proper depiction of contact during planning of CSG-based surgery can help the surgeon to make decisions, provide mental image-based guidance during surgery, and improve the interaction with the planning e.g. choosing between candidate configurations or making small modifications to an existing configuration. In order to enhance the visual depiction of CSG-bone contact, we developed an efficient ambient occlusion technique for volume data. Ambient occlusion (Zhukov et al., 1998) is used in many areas of the graphics domain and is an excellent way to improve depth and shape perception (Lindemann and Ropinski, 2011; Langer and Bülthoff, 1999). It also aids in the visual communication of contact. It is for instance widely used to visualize molecular structures (Moll et al., 2005; Marco et al., 2006). This technique has also been applied to direct volume rendering, by taking into account a limited amount of neighboring voxels (Stewart, 2003; Penner and Mitchell, 2008; Hernell et al., 2010), or the entire volume (Kroes et al., 2012). In our approach, we harness the power of 3D image filtering and a probabilistic heuristic to compute ambient occlusion in volume data. Our main contribution of this work is that we make ambient occlusion possible in applications that use real-time direct volume rendering, such as computer assisted surgery or scientific visualization. Not only does our technique allow for interactive changes of the transfer function, it is also easy to implement, and does not require expensive preprocessing (Ropinski et al., 2008b), as the ambient occlusion is computed on the fly. Further, we show that it can be used effectively to enhance the depiction of contact.

## 7.3 REALISTIC RENDERING AND REMOTE VOLUME VISUALIZATION

The visualization literature that we reviewed in Chapter 2 clearly shows that realistic rendering (e.g., shadows, depth of field, and reflections) is quite rare in CAS. Most applications that we reviewed use simple phong-based reflection models and typically omit shadows and complex lighting effects. Partly because the algorithms that are capable of producing these effects are computationally expensive, and because the added value of realistic rendering is sometimes disputable.

However, there are cases in which more realism is required. For instance in laparoscopy simulators, significant effort is put into realistic scattering models (Kerwin et al., 2009) and interaction with tissues (Morris et al., 2008; Halic et al., 2010). The premise here is that if there is a high visual similarity between simulated and real operating scenario, the actual realism of the surgical simulator also increases. Other examples include the photo-realistic rendering of tissues in augmented reality for minimally invasive surgery Stoyanov et al. (2003).

This trend is also noticeable in volume visualization, where increasingly realistic illumination models are also developed. Recent research shows that realistic volume illumination helps to understand shape, depth and size in volume visualization (Ropinski et al., 2010; Lindemann and Ropinski, 2011). The vast majority of realistic volume illumination models that are available at the time of this writing are physically plausible, meaning that they trade physical correctness for interactivity (Rezk-Salama, 2007; Schott et al., 2011, 2009; Ritschel, 2007; Šoltészová et al., 2010; Wyman et al., 2006; Csébfalvi and Szirmay-Kalos, 2003; Lindemann and Ropinski, 2010; Schlegel et al., 2011; Kronander et al., 2011; Díaz et al., 2010; Sundén and Ropinski, 2015). All of these techniques impose constraints on the scene configuration, for instance the number, shape and location of lights, accuracy of shadows, and limited scattering models. Techniques that are physically-correct do exist, but have little added value for CAS because the rendering process is very time consuming and usually an offline process (Wyman et al., 2006; Beason et al., 2006).

In Chapter 5, we developed a novel direct volume rendering (DVR) framework (Exposure Render (Kroes et al., 2012)) in an attempt to mitigate the limitations of previous physically plausible techniques. We demonstrate that progressive stochastic Monte Carlo raytracing is possible using modern GPUs, and that it can produce very realistic images, even with single scattering. Since Exposure Render is based on stochastic sampling, effects that are otherwise hard to obtain, are easily integrated in our framework e.g., soft shadows, depth-of-field, and blurred reflections. This being said, raytracing is still a computationally expensive process; numerous rays need to be traced for a perceptually noise-free image. Fortunately, the convergence rate of raytracing programs can be improved by exploiting knowledge about the scene (Debevec, 1998; Clarberg et al., 2005; Pharr and Humphreys, 2010).

In Chapter 6 we show that the performance of our DVR framework can be improved by incorporating visibility information in the ray sampling process. In our unbiased approach, we combine environment lighting with local volume visibility information to guide rays in the sampling process toward high energy carrying paths. In order to achieve this, we developed visibility sweeps, an efficient way to compute and store visibility information in volume data sets. In addition, we developed a two-step ray sampling process which, in contrast to existing techniques (Behrens and Ratering, 1998; Ritschel, 2007; Lindemann and Ropinski, 2010; Kronander et al., 2012), allows us to take low and high frequency illumination into account. As a result, images converge faster, which is very important in CAS.

Finally, we developed a generic, image based, remote visualization framework that can be used to remotely interact with medical visualizations. A key requirement of this framework was accessibility, as a result, the medical visualizations can be consulted from nearly any electronic device that is able to display web pages with a modern browser. With this technology, the visualizations can even be accessed in the operating theater, were intraoperative changes to for instance a planning can be made. Collaboration is also simplified, because medical professionals can access the planning from different locations, even from different devices.

Further, we combine Exposure Render with our remote visualization framework to build an exemplary doctor-patient communication system that allows patients to look at their own data, and aims to improve doctor-patient communication. This system is designed to provide a digital bridge between doctor and patient by improving the way that tomographic data is presented to the patient. The standard way in which tomographic data is presented to a patient makes it difficult to interpret, because the visual representations are relatively simple. Our doctor-patient communication system harnesses the power of photo-realism (e.g., realistic shadows and reflections) to make the tomographic data more comprehensible.

With this work, we have laid the foundations for planning of CSG-based joint replacement surgery. We solve the problem of CSG configuration by optimizing its parameters in sillico. We enhanced the planning environment with rich visual depictions of contact and hyper realistic renderings of the operating scenario, and built a remote visualization prototype that allows for remote collaboration and improved doctor-patient communication.

## 7.4 FUTURE WORK

The optimization method described in Chapter 3 is generic; it takes any pin-based CSG design as input. Although we demonstrated its effectiveness with experiments that included a distal femur and a pin-based CSG design, the optimization of the CSG design itself is also a research question. If the limited space, in which surgeons can maneuver instruments is taken into account, for instance with our visibility sweeps method described in Chapter 6, the design of a CSG can be tailored not only to the specific docking geometry, but also to the type of surgical exposure.

We also see potential use for our visibility sweeps in other areas of medical visualization, for instance in to automatically align cameras (Monclús et al., 2012; Vázquez et al., 2008; Bordoloi and Shen, 2005).

Even though we prove, with experiments, that our optimization leads to consistent and accurate results, our current approach does not factor in the cumulative errors in the CSG-based surgery pipeline e.g., image acquisition, bone reconstruction, and manual configuration errors. In order to make our optimization method more realistic, future research should model the uncertainty associated with bone reconstruction and CSG configuration, and integrate in the optimization routine.

Further, a clinical pilot study should be conducted in which the effectiveness of our optimization method is evaluated with more realistic instruments on anatomical specimens.

# Bibliography

Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, and Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Trans. Graph.*, 22(3):605–612, 2003.

Marco Agus, Andrea Giachetti, Enrico Gobbetti, Gianluigi Zanetti, and Antonio Zorcolo. A multiprocessor decoupled system for the simulation of temporal bone surgery. *Computing and Visualization in Science*, 5:35 – 43, 2002. ISSN 1432-9360. URL `http://dx.doi.org/10.1007/s00791-002-0085-5`.

Neculai Archip, Olivier Clatz, Stephen Whalen, Dan Kacher, Andriy Fedorov, Andriy Kot, Nikos Chrisochoides, Ferenc Jolesz, Alexandra Golby, Peter M. Black, and Simon K. Warfield. Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery. *NeuroImage*, 35(2):609–624, April 2007. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2006.11.060. URL `http://www.sciencedirect.com/science/article/pii/S1053811906011694`.

Michael Ashikmin, Simon Premože, and Peter Shirley. A microfacet-based brdf generator. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pages 65 – 74, 2000.

H. Atmani, F. Merienne, D. Fofi, and P. Trouilloud. Computer aided surgery system for shoulder prosthesis placement. *Computer Aided Surgery*, 12(1):60 – 70, 2007.

Emmanuel Audenaert, Karla De Smedt, Frederik Gelaude, Tim Clijmans, Christophe Pattyn, and Ben Geebelen. A custom-made guide for femoral component positioning in hip resurfacing arthroplasty: development and validation study. *Computer Aided Surgery*, 16(6):304–309, 2011.

Nora Baka, Bart L Kaptein, J Erik Giphart, Marius Staring, Marleen de Bruijne, Boudewijn PF Lelieveldt, and Edward Valstar. Evaluation of automated statistical shape model based knee kinematics from biplane fluoroscopy. *Journal of biomechanics*, 47(1):122–129, 2014.

D. C Banks and K. Beason. Decoupling illumination from isosurface generation using 4D light transport. *IEEE Transactions on Visualization and Computer Graphics*, 15 (6):1595–1602, December 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.137.

Giuseppe Barbarino, Mahmood Jabareen, Juergen Trzewik, and Edoardo Mazza. Physically based finite element model of the face. In Fernando Bello and P. Edwards, editors, *Biomedical Simulation*, volume 5104 of *Lecture Notes in Computer Science*, pages 1–10. Springer Berlin / Heidelberg, 2008. URL http://dx.doi.org/10.1007/978-3-540-70521-5_1.

B. Barsky, D. Horn, S. Klein, J. Pang, and M. Yu. Camera models and optical systems used in computer graphics: part ii, image-based techniques. *Computational Science and Its Applications ICCSA 2003*, pages 983–983, 2003.

H Bäthis, L Perlick, M Tingart, C Lüring, D Zurakowski, and J Grifka. Alignment in total knee arthroplasty a comparison of computer-assisted surgery with the conventional technique. *Journal of Bone & Joint Surgery, British Volume*, 86(5):682–687, 2004.

K.M. Beason, J. Grant, D.C. Banks, B. Futch, and M.Y. Hussaini. Pre-computed illumination for isosurfaces. In *Proceedings of SPIE*, volume 6060, pages 98 – 108. Citeseer, 2006.

U. Behrens and R. Ratering. Adding shadows to a texture-based volume renderer. In *Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 39 – 46. ACM, 1998. ISBN 1581131054.

J. Beyer, M. Hadwiger, S. Wolfsberger, and K. Buhler. High-Quality multimodal volume rendering for preoperative planning of neurosurgical interventions. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1696 –1703, December 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70560.

Philippe Blasi, Bertrand Le Saec, and Christophe Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12:201–210, 1993. ISSN 1467-8659. doi: 10.1111/1467-8659.1230201. URL http://dx.doi.org/10.1111/1467-8659.1230201.

J.F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198. ACM, 1977.

Udeepta D Bordoloi and Han-Wei Shen. View selection for volume rendering. In *Visualization, 2005. VIS 05. IEEE*, pages 487–494. IEEE, 2005.

Andrea Bottino, Politecnico Torino, Aldo Laurentini, and Luisa Rosano. A new computer-aided technique for planning the aesthetic outcome of plastic surgery. 2008.

H. Bourquain, A. Schenk, F. Link, B. Preim, G. Prause, and HO Peitgen. Hepavision2: A software assistant for preoperative planning in living-related liver transplantation and oncologic liver surgery. *Computer Assisted Radiology and Surgery (CARS 2002)*, pages 341–346, 2002.

M. Bro-Nielsen. *Medical Image Registration and Surgery Simulation.* PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 1996. URL `http://www.imm.dtu.dk/~bro/phd.html`. IMM-PHD-1996-25.

M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490 –503, March 1998. ISSN 0018-9219. doi: 10.1109/5.662874.

A. Buades, B. Coll, and J.M. Morel. The staircasing effect in neighborhood filters and its solution. *Image Processing, IEEE Transactions on*, 15(6):1499–1505, 2006.

David Burke. Bidirectional importance sampling for illumination from environment maps. Master's thesis, UBC, 2004.

David Cash, Michael Miga, Sean Glasgow, Benoit Dawant, Logan Clements, Zhujiang Cao, Robert Galloway, and William Chapman. Concepts and preliminary data toward the realization of image-guided liver surgery. *Journal of Gastrointestinal Surgery*, 11(7):844–859, 2007. ISSN 1091-255X. doi: 10.1007/s11605-007-0090-6. URL `http://www.springerlink.com/content/v87155557t660605/abstract/`.

Juan Cebral, Rainald Löhner, Orlando Soto, Peter Choyke, and Peter Yim. Patient-Specific simulation of carotid artery stenting using computational fluid dynamics. In Wiro Niessen and Max Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001*, volume 2208 of *Lecture Notes in Computer Science*, pages 153–160. Springer Berlin / Heidelberg, 2001. ISBN 978-3-540-42697-4. URL `http://www.springerlink.com/content/xtkb0u10w1m6n1h9/abstract/`.

M. Chabanas, C. Marecaux, Y. Payan, and F. Boutault. Computer aided planning for orthognatic surgery. *ArXiv Physics e-prints*, October 2006.

Matthieu Chabanas, Vincent Luboz, and Yohan Payan. Patient specific finite element model of the face soft tissues for computer-assisted maxillofacial surgery. *Medical Image Analysis*, 7(2):131–151, June 2003. ISSN 13618415.

Jonas Chapuis, Alexander Schramm, Ion Pappas, Wock Hallermann, Katja Schwenzer-Zimmerer, Frank Langlotz, and Marco Caversaccio. A new system for computer-aided preoperative planning and intraoperative navigation during corrective jaw surgery. *Information Technology in Biomedicine, IEEE Transactions on*, 11(3):274–287, 2007.

Gang Chen, Xue-cheng Li, Guo-qing Wu, Yi Wang, Bin Fang, Xiao-feng Xiong, Ri-gao Yang, Li-wen Tan, Shao-xiang Zhang, and Jia-hong Dong. The use of virtual reality for the functional simulation of hepatic tumors (case control study). *International journal of surgery (London, England)*, 8(1):72–8, January 2010. ISSN 1743-9159.

Nuttapong Chentanez, Ron Alterovitz, Daniel Ritchie, Lita Cho, Kris K. Hauser, Ken Goldberg, Jonathan R. Shewchuk, and James F. O'Brien. Interactive simulation of surgical needle insertion and steering. *ACM Trans. Graph.*, 28:88:1–88:10, July 2009. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/1531326.1531394. URL http://doi.acm.org/10.1145/1531326.1531394.

Wooshin Cho, MY Kim, and YS Youm. *Knee joint arthroplasty.* Springer, 2013.

M. Cimerman and A. Kristan. Preoperative planning in pelvic and acetabular surgery: the value of advanced computerised planning modules. *Injury*, 38(4):442–449, 2007.

P Cinquin, E Bainville, C Barbe, E Bittar, V Bouchard, L Bricault, G Champleboux, M Chenin, L Chevalier, Y Delnondedieu, et al. Computer assisted medical interventions. *Engineering in Medicine and Biology Magazine, IEEE*, 14(3):254–263, 1995.

Petrik Clarberg and Tomas Akenine-Möller. Practical product importance sampling for direct illumination. *Computer Graphics Forum (Proceedings of Eurographics)*, 27 (2), 2008.

Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph.*, 24(3):1166–1175, 2005.

David Cline, Parris K. Egbert, Justin F. Talbot, and David L. Cardon. Two Stage Importance Sampling for Direct Lighting. In *Eurographics Symposium on Rendering*, 2006.

Steve Cohan. Robodoc achieves pinless registration. *Industrial Robot: An International Journal*, 28(5):381–386, 2001.

O. Comas, Z.A. Taylor, J. Allard, S. Ourselin, S. Cotin, and J. Passenger. Efficient nonlinear fem for soft tissue modelling and its gpu implementation within the open source framework sofa. *Lecture Notes in Computer Science*, 5104:28, 2008. ISSN 0302-9743.

S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(1):62 – 73, 1999. ISSN 1077-2626.

Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*, 16(8):437–452, 2000. ISSN 0178-2789. doi: 10.1007/PL00007215. URL http://www.springerlink.com/content/cy5lctanvgnmb3x2/abstract/.

Hadrien Courtecuisse, Hoeryong Jung, Jrmie Allard, Christian Duriez, Doo Yong Lee, and Stphane Cotin. Gpu-based real-time soft tissue deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology*, 103(2-3):159 – 168, 2010. ISSN 0079-6107.

B Couteau, P Mansat, E Estivalèzes, R Darmana, M Mansat, and J Egan. Finite element analysis of the mechanical behavior of a scapula implanted with a glenoid prosthesis. *Clinical biomechanics (Bristol, Avon)*, 16(7):566–75, August 2001. ISSN 0268-0033.

Cyril Crassin, Fabrice Neyret, Miguel Sainz, and Elmar Eisemann. *GPU Pro*, chapter X.3 Efficient Rendering of Highly Detailed Volumetric Scenes with GigaVoxels, pages 643–676. AK Peters, 2010.

B. Csébfalvi and S.K. Szirmay-Kalos. Monte carlo volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 59. IEEE Computer Society, 2003.

B. Dagon, C. Baur, and V. Bettschart. Real-time update of 3d deformable models for computer aided liver surgery. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1 – 4. IEEE, 2009.

Brian Davies, Matjaz Jakopec, Simon J Harris, Ferdinando Rodriguez y Baena, Adrian Barrett, Alexander Evangelidis, Paula Gomes, Johann Henckel, and Justin Cobb. Active-constraint robotics for surgery. *Proceedings of the IEEE*, 94(9):1696–1704, 2006.

Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 189–198. ACM, 1998.

Xavier Décoret. N-buffers for efficient depth map query. *Computer Graphics Forum*, 24(3), 2005. URL `http://maverick.inria.fr/Publications/2005/Dec05`.

José Díaz, Pere-Pau Vázquez, Isabel Navazo, and Florent Duguet. Real-time ambient occlusion and halos with summed area tables. *Computers & Graphics*, 34(4):337–350, 2010.

C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Computational steering for patient-specific implant planning in orthopedics. *Proceedings of Visual Computing for Biomedicine*, page 83–92, 2008.

C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, December 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.184.

Anthony M. Digioia, Branislav Jaramaz, Constantinos Nikou, Richard S. Labarca, James E. Moody, and Bruce D. Colgan. Surgical navigation for total hip replacement with the use of hipnav. *Operative Techniques in Orthopaedics*, 10(1):3–8, January 2000. ISSN 1048-6666. doi: 10.1016/S1048-6666(00)80036-1. URL http://www.sciencedirect.com/science/article/pii/S1048666600800361.

a Digioiaiii, B Jaramaz, C Nikou, R Labarca, J Moody, and B Colgan. Surgical navigation for total hip replacement with the use of hipnav. *Operative Techniques in Orthopaedics*, 10(1):3–8, January 2000. ISSN 10486666.

Prashanth Dumpuri, Logan W Clements, Benoit M Dawant, and Michael I Miga. Model-updated image-guided liver surgery: Preliminary results using surface characterization. *Progress in biophysics and molecular biology*, (September):1–11, September 2010. ISSN 1873-1732.

Paul S. D'Urso, Timothy M. Barker, W. John Earwaker, Lain J. Bruce, R. Leigh Atkinson, Michael W. Lanigan, John F. Arvier, and David J. Effeney. Stereolithographic biomodelling in cranio-maxillofacial surgery: a prospective trial. *Journal of Cranio-Maxillofacial Surgery*, 27(1):30–37, February 1999. ISSN 1010-5182. doi: 10.1016/S1010-5182(99)80007-9. URL http://www.sciencedirect.com/science/article/pii/S1010518299800079.

Mohamed ElHelw, Stella Atkins, Marios Nicolaou, Adrian Chung, and Guang-Zhong Yang. Photo-realistic tissue reflectance modelling for minimally invasive surgical simulation. In James Duncan and Guido Gerig, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2005*, volume 3749 of *Lecture Notes in Computer Science*, pages 868–875. Springer Berlin / Heidelberg, 2005.

Philippe Favre, Beat Moor, Jess G Snedeker, and Christian Gerber. Influence of component positioning on impingement in conventional total shoulder arthroplasty. *Clinical biomechanics (Bristol, Avon)*, 23(2):175–83, February 2008. ISSN 0268-0033.

M. Feuerstein, T. Mussack, S.M. Heining, and N. Navab. Intraoperative laparoscope augmentation for port placement and resection planning in minimally invasive liver resection. *Medical Imaging, IEEE Transactions on*, 27(3):355 – 369, March 2008. ISSN 0278-0062.

Aaron G Filler. The history, development and impact of computed imaging in neurological diagnosis and neurosurgery: Ct, mri, and dti. *Nature Precedings*, 7(1):1–69, 2009.

M. Fleute, S. Lavallee, and R. Julliard. Incorporating a statistically based shape model into a system for computer-assisted anterior cruciate ligament surgery. *Medical Image Analysis*, 3(3):209, 1999.

Henry Fuchs, Mark Livingston, Ramesh Raskar, D'nardo Colucci, Kurtis Keller, Andrei State, Jessica Crawford, Paul Rademacher, Samuel Drake, and Anthony Meyer. Augmented reality visualization for laparoscopic surgery. In William Wells, Alan Colchester, and Scott Delp, editors, *Medical Image Computing and Computer-Assisted Interventation — MICCAI'98*, volume 1496 of *Lecture Notes in Computer Science*, pages 934–943. Springer Berlin / Heidelberg, 1998. ISBN 978-3-540-65136-9. URL `http://www.springerlink.com/content/h114gt752mj4u002/abstract/`.

Christos C Galanis, Michael M Sfantsikopoulos, Petros T Koidis, Nikolaos M Kafantaris, and Pavlos G Mpikos. Computer methods for automating preoperative dental implant planning: implant positioning and size assignment. *Computer methods and programs in biomedicine*, 86(1):30–8, April 2007. ISSN 0169-2607.

Iliyan Georgiev, Jaroslav Křivánek, Toshiya Hachisuka, Derek Nowrouzezahrai, and Wojciech Jarosz. Joint importance sampling of low-order volumetric scattering. *ACM Transactions on Graphics (TOG)*, 32(6):164, 2013.

David Gering, Arya Nabavi, Ron Kikinis, W. Grimson, Noby Hata, Peter Everett, Ferenc Jolesz, and William Wells. An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. In Chris Taylor and Alain Colchester, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI'99*, volume 1679 of *Lecture Notes in Computer Science*, pages 809–819. Springer Berlin / Heidelberg, 1999. ISBN 978-3-540-66503-8. URL `http://www.springerlink.com/content/83316311413115v5/abstract/`.

I.M. Germano. The neurostation system for image-guided, frameless stereotaxy. *Neurosurgery*, 37(2):348, 1995. ISSN 0148-396X.

E. Gladilin, S. Zachow, P. Deuflhard, and H.C. Hege. Realistic prediction of individual facial emotion expressions for craniofacial surgery simulations. In *Proceedings of SPIE*, volume 5029, page 520, 2003.

E. Gladilin, S. Zachow, P. Deuflhard, and H. Hege. Anatomy- and physics-based facial animation for craniofacial surgery simulations. *Medical and Biological Engineering and Computing*, 42:167 – 170, 2004. ISSN 0140-0118.

G Glombitza, W Lamadé, a M Demiris, M R Göpfert, a Mayer, M L Bahner, H P Meinzer, G Richter, T Lehnert, and C Herfarth. Virtual planning of liver resections: image processing, visualization and volumetric evaluation. *International journal of medical informatics*, 53(2-3):225–37, 1999. ISSN 1386-5056.

J. Goffin, K. Van Brussel, K. Martens, J. Vander Sloten, R. Van Audekercke, and M.H. Smet. Three-dimensional computed tomography-based, personalized drill guide for posterior cervical stabilization at c1-c2. *Spine*, 26(12):1343, 2001. ISSN 0362-2436.

W L Grimson, G J Ettinger, S J White, T Lozano-Perez, W M Wells, and R Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. *IEEE transactions on medical imaging*, 15(2):129–40, January 1996. ISSN 0278-0062.

P. Grunert, K. Darabi, J. Espinosa, and R. Filippi. Computer-aided navigation in neurosurgery. *Neurosurgical Review*, 26(2):73–99, 2003. ISSN 0344-5607. doi: 10.1007/s10143-003-0262-0. URL http://www.springerlink.com/content/9a7hw41pcbm7k46q/abstract/.

Markus Hadwiger, Andrea Kratz, Christian Sigg, and Katja Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 49 – 52, New York, NY, USA, 2006. ACM. ISBN 3-905673-37-1. doi: 10.1145/1283900.1283908. URL http://portal.acm.org/citation.cfm?id=1283900.1283908.

MA Hafez, KL Chelule, BB Seedhom, and KP Sherman. Computer-assisted total knee arthroplasty using patient-specific templates: the custom-made cutting guides. *Navigation and MIS in Orthopedic Surgery*, pages 182 – 188, 2007.

Tansel Halic, Ganesh Sankaranarayanan, and Suvranu De. Gpu-based efficient realistic techniques for bleeding and smoke generation in surgical simulators. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 6(4):431–443, 2010. ISSN 1478-596X. doi: 10.1002/rcs.353. URL http://dx.doi.org/10.1002/rcs.353.

Walter a Hall and Charles L Truwit. Intraoperative mr-guided neurosurgery. *Journal of magnetic resonance imaging : JMRI*, 27(2):368–75, February 2008. ISSN 1053-1807.

Lianghao Han, John Hipwell, Zeike Taylor, Christine Tanner, Sebastien Ourselin, and David Hawkes. Fast deformation simulation of breasts using gpu-based dynamic explicit finite element method. In Joan Mart, Arnau Oliver, Jordi Freixenet, and Robert Mart, editors, *Digital Mammography*, volume 6136 of *Lecture Notes in Computer Science*, pages 728–735. Springer Berlin / Heidelberg, 2010. URL http://dx.doi.org/10.1007/978-3-642-13666-5_98.

C. Hansen, S. Zidowitz, A. Schenk, K.-J. Oldhafer, H. Lang, and H.-O. Peitgen. Risk maps for navigation in liver surgery. In *Proc. of SPIE Medical Imaging*, pages 762528–762528–8, 2010. doi: 10.1117/12.843493. URL http://link.aip.org/link/PSISDG/v7625/i1/p762528/s1&Agg=doi.

Christian Hansen. *Software Assistance for Preoperative Risk Assessment and Intraoperative Support in Liver Resection Surgery.* Ph.D. thesis, Jacobs University, Bremen, 2012.

Christian Hansen, Alexander Kohn, Stefan Schlichting, Florian Weiler, Stephan Zidowitz, Markus Kleemann, and Heinz-Otto Peitgen. Intraoperative modification of resection plans for liver surgery. *International Journal of Computer Assisted Radiology and Surgery*, 3(3-4):291–297, June 2008a. ISSN 1861-6410.

Christian Hansen, Stefan Schlichting, Stephan Zidowitz, Alexander K, Milo Hindennach, Markus Kleemann, and Heinz-otto Peitgen. Intraoperative adaptation and visualization of preoperative risk analyses for oncologic liver surgery. *Methods*, 2008b.

M Harders, A Barlit, Ch Gerber, J Hodler, and G Sz. An optimized surgical planning environment for complex proximal humerus fractures. 2007.

M Haselbacher, K Sekyra, E Mayr, M Thaler, and M Nogler. A new concept of a multiple-use screw-based shape-fitting plate in total knee arthroplasty. *Journal of Bone & Joint Surgery, British Volume*, 94(SUPP XLIV):65–65, 2012.

D Hashimoto, T Dohi, M Tsuzuki, T Horiuchi, Y Ohta, K Chinzei, M Suzuki, and Y Idezuki. Development of a computer-aided surgery system: three-dimensional graphic reconstruction for treatment of liver cancer. *Surgery*, 109(5):589–596, May 1991. ISSN 0039-6060. URL http://www.ncbi.nlm.nih.gov/pubmed/1850556. PMID: 1850556.

Stefan Hassfeld and Joachim Mühling. Computer assisted oral and maxillofacial surgery – a review and an assessment of technology. *International Journal of Oral and Maxillofacial Surgery*, 30(1):2–13, February 2001. ISSN 0901-5027. doi: 10.1054/ijom.2000.0024. URL http://www.sciencedirect.com/science/article/pii/S0901502700900249.

J. Heisterkamp, R. van Hillegersberg, and J. N. M IJzermans. Interstitial laser coagulation for hepatic tumours. *British Journal of Surgery*, 86(3):293–304, March 1999. ISSN 1365-2168. doi: 10.1046/j.1365-2168.1999.01059.x. URL http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2168.1999.01059.x/abstract.

Bradley Hemminger, Paul Molina, Thomas Egan, Frank Detterbeck, Keith Muller, Christopher Coffey, and Joseph Lee. Assessment of Real-Time 3D visualization for cardiothoracic diagnostic evaluation and surgery planning. *Journal of Digital Imaging*, 18(2):145–153, 2005. ISSN 0897-1889. doi: 10.1007/s10278-004-1909-2. URL http://www.springerlink.com/content/x05302n06n505856/abstract/.

Frida Hernell, Patric Ljung, and Anders Ynnerman. Local ambient occlusion in direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 16 (4):548–559, 2010.

John H Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

David Holmes III, Maryam Rettmann, and Richard Robb. Visualization in image-guided interventions. *Image-Guided Interventions: Technology and Applications*, page 45, 2008.

Hamid Reza Seyyed Hosseinzadeh, Samih Tarabichi, Ali Sina Shahi, Mehrnoush Hassas Yeganeh, Usama Hassan Saleh, Gholam Reza Kazemian, and Aidin Masoudi. Special considerations in asian knee arthroplasty. 2013.

Q. Hu, U. Langlotz, J. Lawrence, F. Langlotz, and L.P. Nolte. A fast impingement detection algorithm for computer-aided orthopedic surgery. *Computer Aided Surgery*, 6 (2):104 – 110, 2001. ISSN 1097-0150.

J.P. iannotti, E.E. Spencer, et al. Prosthetic positioning in total shoulder arthroplasty. *Journal of Shoulder and Elbow Surgery*, 14(1):S111 – S121, 2005.

M. Jakopec, R. y Baena, S.J. Harris, P. Gomes, J. Cobb, and B.L. Davies. The hands-on orthopaedic robot. *Robotics and Automation, IEEE Transactions on*, 19(5):902 – 911, 2003. ISSN 1042-296X.

Branislav Jaramaz, Mahmoud A Hafez, and Anthony M DiGioia. Computer-assisted orthopaedic surgery. *Proceedings of the IEEE*, 94(9):1689–1695, 2006.

Daniel Jnsson, Erik Sundn, Anders Ynnerman, and Timo Ropinski. A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering. *Computer Graphics Forum*, 33(1):27–51, 2014.

Leo Joskowicz and Russell H Taylor. Computers in imaging and guided surgery. *Computing in Science & Engineering*, 3(5):65–72, 2001.

Sebastien Jourdain, Utkarsh Ayachit, and Berk Geveci. Paraviewweb, a web framework for 3d visualization and data processing. In *IADIS international conference on web virtual reality and three-dimensional worlds*, volume 7, page 1, 2010.

Vladimir Kajalin. Screen space ambient occlusion. In Wolfgang Engel, editor, *ShaderX7*. Cengage Learning, Boston, 2009.

Armin Kanitsar, Dominik Fleischmann, Rainer Wegenkittl, Petr Felkel, and Meister Eduard Grller. Cpr - curved planar reformation. In *In Visualization'02. IEEE*, pages 37–44, 2002.

Erwin Keeve, Sabine Girod, Paula Pfeifle, and Bernd Girod. Anatomy-based facial tissue modeling using the finite element method. 1996.

Patrick J. Kelly and George J. Alker Jr. A stereotactic approach to deep-seated central nervous system neoplasms using the carbon dioxide laser. *Surgical Neurology*, 15 (5):331–334, May 1981. ISSN 0090-3019. doi: 10.1016/0090-3019(81)90158-0. URL `http://www.sciencedirect.com/science/article/pii/0090301981901580`.

Marta Kersten-Oertel, Pierre Jannin, and D Collins. Dvv: Towards a taxonomy for mixed reality visualization in image guided surgery. *Medical Imaging and Augmented Reality*, pages 334–343, 2010.

T. Kerwin, Han-Wei Shen, and D. Stredney. Enhancing realism of wet surfaces in temporal bone surgical simulation. *Visualization and Computer Graphics, IEEE Transactions on*, 15(5):747 –758, October 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.31.

Rostislav Khlebnikov, Bernhard Kainz, Judith Muehl, and Dieter Schmalstieg. Crepuscular rays for tumor accessibility planning. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2163–2172, December 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.184.

R. Kikinis, P.L. Gleason, T.M. Moriarty, M.R. Moore, E. Alexander III, P.E. Stieg, M. Matsumae, W.E. Lorensen, H.E. Cline, P.M.L. Black, et al. Computer-assisted interactive three-dimensional planning for neurosurgical procedures. *Neurosurgery*, 38(4):640, 1996. ISSN 0148-396X.

Hyungmin Kim, Philipp Jrgens, Lutz-Peter Nolte, and Mauricio Reyes. Anatomically-driven soft-tissue simulation strategy for cranio-maxillofacial surgery using facial muscle template model. In Tianzi Jiang, Nassir Navab, Josien Pluim, and Max Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2010*, volume 6361 of *Lecture Notes in Computer Science*, pages 61 – 68. Springer Berlin / Heidelberg, 2010a.

Hyungmin Kim, Philipp Jrgens, Stefan Weber, Lutz-Peter Nolte, and Mauricio Reyes. A new soft-tissue simulation strategy for cranio-maxillofacial surgery using facial muscle template model. *Progress in Biophysics and Molecular Biology*, 103(2-3):284 – 291, 2010b. ISSN 0079-6107. Special Issue on Biomechanical Modelling of Soft Tissue Motion.

Tae-Yong Kim and Ulrich Neumann. Opacity shadow maps. In *In Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182. Springer-Verlag, 2001.

Brian A Klatt, Nitin Goyal, Matthew S Austin, and William J Hozack. Custom-fit total knee arthroplasty (otisknee) results in malalignment. *The Journal of arthroplasty*, 23(1):26–29, 2008.

J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, pages 270 – 285, 2002. ISSN 1077-2626.

J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):150 – 162, 2003. ISSN 1077-2626.

D.E. Knuth. *Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley Professional, 1997.

P. Kohlmann, S. Bruckner, M. Eduard Groller, and A. Kanitsar. LiveSync: deformed viewing spheres for Knowledge-Based navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1544 –1551, December 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70576.

Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. In *Rendering Techniques*, volume 44, pages 45–51. Eurographics Association, 2003.

O. Konrad-Verse, B. Preim, and A. Littmann. Virtual resection with a deformable cutting plane. In *Proceedings of simulation und visualisierung*, volume 2004, pages 203 – 214. Citeseer, 2004.

J. Kontkanen, J. Räsänen, and A. Keller. Irradiance filtering for monte carlo ray tracing. *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 259–272, 2006.

J Kowal, F Langlotz, and L-P Nolte. Basics of computer-assisted orthopaedic surgery. In *Navigation and MIS in Orthopedic Surgery*, pages 2–8. Springer, 2007.

P R Krekel, P W de Bruin, E R Valstar, F H Post, P M Rozing, and C P Botha. Evaluation of bone impingement prediction in pre-operative planning for shoulder arthroplasty. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 223(7):813–822, October 2009. ISSN 0954-4119.

Peter R. Krekel, Charl P. Botha, Edward R. Valstar, Paul W. de Bruin, Piet M. Rozing, and Frits H. Post. Interactive simulation and comparative visualisation of the bone-determined range of motion of the human shoulder. In Thomas Schulze, Graham Horton, Bernhard Preim, and Stefan Schlechtweg, editors, *Proceedings of Simulation and Visualization*, page 275–288. SCS Publishing House Erlangen, March 2006. ISBN 3-936150-46.x. Best Paper Award.

Thomas Kroes, Frits H Post, and Charl P Botha. Exposure render: An interactive photo-realistic volume rendering framework. *PloS one*, 7(7):e38586, 2012.

Joel Kronander, Daniel Jonsson, Joakim Low, Patric Ljung, Anders Ynnerman, and Jonas Unger. Efficient visibility encoding for dynamic illumination in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 99 (PrePrints), 2011. ISSN 1077-2626. doi: http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.35.

Joel Kronander, Daniel Jönsson, Joakim Löw, Patric Ljung, Anders Ynnerman, and Jonas Unger. Efficient Visibility Encoding for Dynamic Illumination in Direct Volume Rendering. *IEEE TVCG*, 18(3):447–462, 2012.

P. Lamata, A. Jalote-Parmar, F. Lamata, and J. Declerck. The resection map, a proposal for intraoperative hepatectomy guidance. *International Journal of Computer Assisted Radiology and Surgery*, 3(3):299–306, 2008. ISSN 1861-6410. doi: 10.1007/s11548-008-0226-5. URL `http://www.springerlink.com/content/h2868u3105277160/abstract/`.

Thomas Lange, Sebastian Eulenstein, Michael Hünerbein, Hans Lamecker, and Peter-Michael Schlag. Augmenting intraoperative 3D ultrasound with preoperative models for navigation in liver surgery. In Christian Barillot, David Haynor, and Pierre Hellier, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*, volume 3217 of *Lecture Notes in Computer Science*, pages 534–541. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-22977-3. URL `http://www.springerlink.com/content/49xvpqjba752dk3p/abstract/`.

Michael S Langer and Heinrich H Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 1999.

Frank Langlotz and Lutz-Peter Nolte. Technical approaches to computer-assisted orthopedic surgery. *European Journal of Trauma*, 30(1):1–11, February 2004. ISSN 1439-0590.

Su-Lin Lee, Adrian Chung, Mirna Lerotic, Maria a Hawkins, Diana Tait, and Guang-Zhong Yang. Dynamic shape instantiation for intra-operative guidance. *Medical image computing and computer-assisted intervention : MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention*, 13(Pt 1): 69–76, January 2010.

Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 55 – 62. ACM, 1995. ISBN 0897917014.

Seth S Leopold. Minimally invasive total knee arthroplasty for osteoarthritis. *New England Journal of Medicine*, 360(17):1749–1758, 2009.

Bing Nan Li, Phu Binh Nguyen, S.H. Ong, Jing Qin, Liang Jing Yang, and C.K. Chui. Image processing and modeling for active needle steering in liver surgery. In *Informatics in Control, Automation and Robotics, 2009. CAR '09. International Asia Conference on*, pages 306 –310, February 2009.

Florian Lindemann and Timo Ropinski. Advanced light material interaction for direct volume rendering. *Eurographics/IEEE VGTC on Volume Graphics*, pages 101–108, 2010.

Florian Lindemann and Timo Ropinski. About the influence of illumination models on image comprehension in direct volume rendering. *IEEE TVCG(Vis Proceedings)*, 17(12):1922–1931, 2011.

T. Lokovic and E. Veach. Deep shadow maps. In *Proc. ACM SIGGRAPH*, pages 385–392, 2000.

W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason. Enhancing reality in the operating room. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pages 410 –415, October 1993. doi: 10.1109/VISUAL.1993.398902.

PAUL A Lotke and MALCOLM L Ecker. Influence of positioning of prosthesis in total knee replacement. *J Bone Joint Surg Am*, 59(1):77–79, 1977.

R. J. Maciunas. Computer-assisted neurosurgery. *Clinical neurosurgery*, 53:267, 2006.

T Marco, C Paolo, and M Claudio. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 2–3, 2006.

J Bohannon Mason, Thomas K Fehring, Rhonda Estok, Deirdre Banel, and Kyle Fahrbach. Meta-analysis of alignment outcomes in computer-assisted total knee arthroplasty surgery. *The Journal of arthroplasty*, 22(8):1097–1106, 2007.

Joost Mattheijer, Just L Herder, Gabriëlle JM Tuijthof, Rob GHH Nelissen, Jenny Dankelman, and Edward R Valstar. Shaping patient specific surgical guides for arthroplasty to obtain high docking robustness. *Journal of Mechanical Design*, 135 (7):071001, 2013.

Joost Mattheijer, Just L Herder, Gabriëlle JM Tuijthof, and Edward R Valstar. Docking robustness of patient specific surgical guides for joint replacement surgery. *Journal of Mechanical Design*, 137(6):062301, 2015.

Georg Matziolis, Doerte Krocker, Ulrike Weiss, Stephan Tohtz, and Carsten Perka. A prospective, randomized study of computer-assisted and conventional total knee arthroplastythree-dimensional evaluation of implant alignment and rotation. *The Journal of Bone & Joint Surgery*, 89(2):236–243, 2007.

N Maurel, a Diop, and J Grimberg. A 3d finite element model of an implanted scapula: importance of a multiparametric validation using experimental data. *Journal of biomechanics*, 38(9):1865–72, September 2005. ISSN 0021-9290.

N. Max. Optical models for direct volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 1(2):99 – 108, 1995. ISSN 1077-2626.

Michael D. McCool. Anisotropic diffusion for monte carlo noise reduction. *ACM Trans. Graph.*, 18:171–194, April 1999. ISSN 0730-0301. doi: http://doi.acm.org/10.1145/318009.318015. URL http://doi.acm.org/10.1145/318009.318015.

À. Méndez-Feliu and M. Sbert. From obscurances to ambient occlusion: A survey. *The Visual Computer*, 25:181–196, 2009. ISSN 0178-2789. URL http://dx.doi.org/10.1007/s00371-008-0213-4. 10.1007/s00371-008-0213-4.

Robert Metzger and Thomas M Vanasse. Patient-specific shoulder guide, November 3 2010. US Patent App. 12/938,913.

Andreas Moll, Andreas Hildebrandt, Hans-Peter Lenhof, and Oliver Kohlbacher. Ballview: An object-oriented molecular visualization and modeling framework. *Journal of computer-aided molecular design*, 19(11):791–800, 2005.

W. Mollemans, F. Schutyser, N. Nadjmi, F. Maes, and P. Suetens. Predicting soft tissue deformations for a maxillofacial surgery planning system: From computational strategies to a complete clinical validation. *Medical Image Analysis*, 11 (3):282–301, June 2007. ISSN 1361-8415. doi: 10.1016/j.media.2007.02.003. URL http://www.sciencedirect.com/science/article/pii/S1361841507000242.

Eva Monclús, Pere-Pau Vázquez, and Isabel Navazo. Efficient selection of representative views and navigation paths for volume data exploration. In *Visualization in Medicine and Life Sciences II*, pages 133–151. Springer, 2012.

D. Morris, C. Sewell, F. Barbagli, N.H. Blevins, S. Girod, and K. Salisbury. Visuohaptic simulation of bone surgery. 2008.

J. Mosegaard, P. Herborg, and T.S. Sorensen. A gpu accelerated spring mass system for surgical simulation. *Studies in health technology and informatics*, 111:342–348, 2005. ISSN 0926-9630.

K. Mühler, M. Neugebauer, C. Tietjen, and B. Preim. Viewpoint selection for intervention planning. In *IEEE/eurographics symposium on visualization (EuroVis)*, page 267–274, 2007. URL http://www.medvis-book.de/MedVisBookMaterial/Chapter20/Muehler2007_EuroVis.pdf.

Laurent Mundeleer, David Wikler, Thierry Leloup, and Nadine Warzee. Development of a computer assisted system aimed at rfa liver surgery. *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society*, 32(7):611–21, October 2008. ISSN 0895-6111.

M Nakao, H Oyama, M Komori, T Matsuda, G Sakaguchi, M Komeda, and T Takahashi. Haptic reproduction and interactive visualization of a beating heart for cardiovascular surgery simulation. *International journal of medical informatics*, 68(1-3):155–63, December 2002. ISSN 1386-5056.

Nikhil Navkar, Nikolaos Tsekos, Jason Stafford, Jeffrey Weinberg, and Zhigang Deng. Visualization and planning of neurosurgical interventions with straight access. In Nassir Navab and Pierre Jannin, editors, *Information Processing in Computer-Assisted Interventions*, volume 6135 of *Lecture Notes in Computer Science*, pages 1–11. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-13710-5. URL `http://www.springerlink.com/content/a090341283111705/abstract/`.

A. Nealen, M. Mller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. In *Computer Graphics Forum*, volume 25, pages 809 – 836. Wiley Online Library, 2006.

Vincent Y Ng, Jeffrey H DeClaire, Keith R Berend, Bethany C Gulick, and Adolph V Lombardi Jr. Improved accuracy of alignment with patient-specific positioning guides compared with manual instrumentation in tka. *Clinical Orthopaedics and Related Research®*, 470(1):99–107, 2012.

SA Nicolau, X. Pennec, L. Soler, and N. Ayache. A complete augmented reality guidance system for liver punctures: First clinical evaluation. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005*, pages 539 – 547, 2005.

S.A. Nicolau, X. Pennec, L. Soler, X. Buy, A. Gangi, N. Ayache, and J. Marescaux. An augmented reality system for liver thermal ablation: Design and evaluation on clinical cases. *Medical Image Analysis*, 13(3):494 – 506, 2009. ISSN 1361-8415.

Harald Niederreiter. *Random Number Generation and quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992. ISBN 0-89871-295-5.

J. Novák, V. Havran, and C. Dachsbacher. Path regeneration for interactive path tracing. *Proc. EUROGRAPHICS Short Papers*, 2010.

Jan Novák, Andrew Selle, and Wojciech Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics (TOG)*, 33(6):179, 2014.

Kirsti Numminen, Outi Sipila, and Heikki Makisalo. Preoperative hepatic 3d models: virtual liver resection using three-dimensional imaging technique. *European journal of radiology*, 56(2):179–84, November 2005. ISSN 0720-048X.

A.M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508, 2004. ISSN 0143-991X. doi: 10.1108/01439910410566362. URL `http://www.emeraldinsight.com/10.1108/01439910410566362`.

Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, 2004.

Yoshito Otake, Naoki Suzuki, Asaki Hattori, Hidenobu Miki, Mitsuyoshi Yamamura, Nobuo Nakamura, Nobuhiko Sugano, Kazuo Yonenobu, and Takahiro Ochi. Estimation of dislocation after total hip arthroplasty by 4-dimensional. *Studies in health technology and informatics*, 111:372–377, 2005.

Brent D Owen, Gary E Christensen, Joseph M Reinhardt, and Timothy C Ryken. Rapid prototype patient-specific drill template for cervical pedicle screw placement. *Computer Aided Surgery*, 12(5):303–308, 2007.

Steven G Parker, James Bigler, Andreas Dietrich, Heiko Friedrich, Jared Hoberock, David Luebke, David McAllister, Morgan McGuire, Keith Morley, Austin Robison, et al. Optix: a general purpose ray tracing engine. In *ACM Transactions on Graphics (TOG)*, volume 29, page 66. ACM, 2010.

Daniel Patel, Veronika Šoltészová, Jan Martin Nordbotten, and Stefan Bruckner. Instant convolution shadows for volumetric detail mapping. *ACM Transactions on Graphics (TOG)*, 32(5):154, 2013.

Eric Penner and Ross Mitchell. Isosurface ambient occlusion and soft shadows with filterable occlusion maps. In *Proceedings of the Fifth Eurographics/IEEE VGTC conference on Point-Based Graphics*, pages 57–64. Eurographics Association, 2008.

Terry M Peters. Image-guidance for surgical procedures. *Physics in Medicine and Biology*, 51(14):R505–R540, July 2006. ISSN 0031-9155, 1361-6560. doi: 10.1088/0031-9155/51/14/R01. URL `http://iopscience.iop.org/0031-9155/51/14/R01`.

R. Petzold, H.-F. Zeilhofer, and W.A. Kalender. Rapid prototyping technology in medicine—basics and applications. *Computerized Medical Imaging and Graphics*, 23(5):277–284, October 1999. ISSN 0895-6111. doi: 10.1016/S0895-6111(99)00025-7. URL `http://www.sciencedirect.com/science/article/pii/S0895611199000257`.

Bernhard Pflesser, Andreas Petersik, Ulf Tiede, Karl Heinz Hhne, and Rudolf Leuwer. Volume cutting for virtual petrous bone surgery. *Computer Aided Surgery*, 7(2):74–83, 2002. ISSN 1097-0150. doi: 10.1002/igs.10036. URL `http://dx.doi.org/10.1002/igs.10036`.

Matt Pharr and Greg Humphreys. *Physically Based Rendering, Second Edition: From Theory To Implementation.* Morgan Kaufmann Publishers Inc., 2nd edition, 2010. ISBN 0123750792, 9780123750792.

Emil Praun and Hugues Hoppe. Spherical parametrization and remeshing. *ACM Trans. Graph.*, 22(3):340–349, 2003.

B. Preim, D. Selle, W. Spindler, K. Oldhafer, and H.O. Peitgen. Interaction techniques and vessel analysis for preoperative planning in liver surgery. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2000*. Springer, 2000.

Bernhard Preim and Charl Botha. *Visual Computing for Medicine*. Morgan Kaufmann, 2 edition, 2013. ISBN 978-0124158733. URL `http://medvisbook.com/`. 00000.

Bernhard Preim, Christian Tietjen, Wolf Spindler, and Heinz-otto Peitgen. Integration of measurement tools in medical 3d visualizations. *Work*, pages 21 – 28, 2002.

Martijn Raaijmaakers, Frederik Gelaude, Karla De Smedt, Tim Clijmans, Jeroen Dille, and Michiel Mulier. A custom-made guide-wire positioning device for hip surface replacement arthroplasty: description and first results. *BMC Musculoskeletal Disorders*, 11(1):161, July 2010. ISSN 1471-2474. doi: 10.1186/1471-2474-11-161. URL `http://www.biomedcentral.com/1471-2474/11/161`.

Klaus Rademacher. *Computerunterstützte Operationsplanung und-ausführung mittels individueller Bearbeitungsschablonen in der Orthopädie*. Shaker, 1999.

K. Radermacher, F. Portheine, M. Anton, A. Zimolong, G. Kaspers, G. Rau, and H.W. Staudte. Computer assisted orthopaedic surgery with image based individual templates. *Clinical orthopaedics and related research*, 354:28, 1998.

K Radermacher, F Portheine, and E Schkommodau. Rechnerbasierte entscheidungsunterstützung zur planung von kontaktflächen zur manuellen referenzierung mit individualschablonen. *Biomedizinische Technik/Biomedical Engineering*, 45(s1): 227–228, 2000.

a Radetzky. Visualization and simulation techniques for surgical simulators using actual patient's data. *Artificial Intelligence in Medicine*, 26(3):255 – 279, November 2002. ISSN 09333657.

S Rajasekaran, S Vidyadhara, Perumal Ramesh, and Ajoy P Shetty. Randomized clinical study to compare the accuracy of navigated and non-navigated thoracic pedicle screws in deformity correction surgeries. *Spine*, 32(2):E56–64, January 2007. ISSN 1528-1159. doi: 10.1097/01.brs.0000252094.64857.ab. URL `http://www.ncbi.nlm.nih.gov/pubmed/17224800`. PMID: 17224800.

Bernhard Reitinger, Alexander Bornik, Reinhard Beichel, and Dieter Schmalstieg. Liver surgery planning using virtual reality. *Liver*, (December):36 – 47, 2006.

Christof Rezk-Salama. GPU-based monte-carlo volume raycasting. In *Proc. of the Pacific Conference on Computer Graphics and Applications (PG07)*, pages 411 – 414. IEEE Computer Society, 2007.

C. Rieder, F. Ritter, M. Raspe, and H.O. Peitgen. Interactive visualization of multimodal volume data for neurosurgical tumor treatment. In *Computer Graphics Forum*, volume 27, page 1055–1062, 2008.

Christian Rieder, Michael Schwier, Andreas Weihusen, Stephan Zidowitz, and Heinz-Otto Peitgen. Visualization of risk structures for interactive planning of image guided radiofrequency ablation of liver tumors. In *SPIE Medical Imaging*, pages 726134–726134. International Society for Optics and Photonics, 2009.

Christian Rieder, Tim Kröger, Christian Schumann, and Horst K. Hahn. GPU-Based Real-Time approximation of the ablation zone for radiofrequency ablation. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1812–1821, December 2011.

Tobias Ritschel. Fast gpu-based visibility computation for natural illumination of volume data sets. In Paolo Cignoni and Jiří Sochor, editors, *Short Paper Proceedings of Eurographics 2007*, pages 17 – 20, September 2007.

D. W. Roberts, A. Hartov, F. E. Kennedy, M. I. Miga, K. D. Paulsen, R. J. Maciunas, P. J. Kelly, R. a. E. Bakay, and G. H. Barnett. Intraoperative brain shift and deformation : A quantitative analysis of cortical displacement in 28 cases. commentaries. *Neurosurgery*, 43(4):749–760, 1998. ISSN 0148-396X. URL `http://cat.inist.fr/?aModele=afficheN&cpsidt=2395719`.

T. Ropinski, C. Doring, and C. Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. In *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, pages 169 – 176. IEEE, 2010.

Timo Ropinski, Jens Kasten, and Klaus H. Hinrichs. Efficient shadows for gpu-based volume raycasting. In *Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2008)*, pages 17 – 24, 2008a. URL `http://viscg.uni-muenster.de/publications/2008/RKH08`.

Timo Ropinski, Jennis Meyer-Spradow, Stefan Diepenbrock, Jörg Mensmann, and Klaus Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. In *Computer Graphics Forum*, volume 27, pages 567–576. Wiley Online Library, 2008b.

Timo Ropinski, Jennis Meyer-Spradow, Stefan Diepenbrock, J??rg Mensmann, and Klaus Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum*, 27(2):567 – 576, 2008c. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2008.01154.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2008.01154.x`.

Fabrice Rousselle, Petrik Clarberg, Luc Leblanc, Victor Ostromoukhov, and Pierre Poulin. Efficient product sampling using hierarchical thresholding. *The Visual Computer*, 24(7-9):465–474, 2008.

J. Ruppin, A. Popovic, M. Strauss, E. Sp
"untrup, A. Steiner, and C. Stoll. Evaluation of the accuracy of three different computer-aided surgery systems in dental implantology: optical tracking vs. stereolithographic splint systems. *Clinical Oral Implants Research*, 19(7):709–716, 2008. ISSN 1600-0501.

Richard M Satava. Emerging technologies for surgery in the 21st century. *Archives of Surgery*, 134(11):1197–1202, 1999.

P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based shading and illumination in gpu volume ray-casting. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1795–1802, 2011.

M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. In *Computer Graphics Forum*, volume 28, pages 855–862. Citeseer, 2009.

Mathias Schott, A.V. Pascal Grosset, Tobias Martin, Vincent Pegoraro, Sean T. Smith, and Charles D. Hansen. Depth of field effects for interactive direct volume rendering. *Computer Graphics Forum*, 30(3):941–950, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2011.01943.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2011.01943.x`.

Reuben Shamir, Idit Tamir, Elad Dabool, Leo Joskowicz, and Yigal Shoshan. A method for planning safe trajectories in Image-Guided keyhole neurosurgery. In Tianzi Jiang, Nassir Navab, Josien Pluim, and Max Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, volume 6363 of *Lecture Notes in Computer Science*, pages 457–464. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-15710-3. URL `http://www.springerlink.com/content/h4615373xq556227/abstract/`.

N. Shevchenko, B. Seidl, J. Schwaiger, M. Markert, and T.C. Lueth. Mimed liver: A planning system for liver surgery. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 1882 – 1885, 2010.

J Sikorski and S Chauhan. Aspects of current management. *J Bone Joint Surg [Br]*, 2003 (85-B):319–23, 2003.

David A Simon and Takeo Kanade. Geometric constraint analysis and synthesis: Methods for improving shape-based registration accuracy. In *CVRMed-MRCAS'97*, pages 181–190. Springer, 1997.

David A Simon and Stéphane Lavallée. Medical imaging and registration in computer assisted surgery. *Clinical Orthopaedics and related research*, 354:17–27, 1998.

David A Simon, Martial Hebert, and Takeo Kanade. Techniques for fast and accurate intrasurgical registration. *Computer Aided Surgery*, 1(1):17–29, 1995.

Sekou Singare, Qin Lian, Wei Ping Wang, Jue Wang, Yaxiong Liu, Dichen Li, and Bingheng Lu. Rapid prototyping assisted surgery planning and custom implant design. *Rapid Prototyping Journal*, 15(1):19 – 23, 2009. ISSN 1355-2546.

Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 527–536. ACM, 2002.

Noeska Natasja Smit, Anne C. Kraima, Daniel Jansma, Marco C. Deruiter, and Charl P. Botha. A unified representation for the model-based visualization of heterogeneous anatomy data. In Miriah Meyer and Tino Weinkauf, editors, *EuroVis - Short Papers*, pages 85–89, Vienna, Austria, 2012. Eurographics Association. URL `http://graphics.tudelft.nl/Publications-new/2012/SKJDB12`.

T Sohmura, H Hojo, M Nakajima, K Wakabayashi, M Nagao, S Iida, T Kitagawa, M Kogo, T Kojima, K Matsumura, et al. Prototype of simulation of orthognathic surgery using a virtual reality haptic device. *International journal of oral and maxillofacial surgery*, 33(8):740–750, 2004.

Veronika Šoltészová, Daniel Patel, Stefan Bruckner, and Ivan Viola. A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum*, 29(3):883 – 891, June 2010. URL `http://www.cg.tuwien.ac.at/research/publications/2010/solteszova-2010-MOS/`.

Erich Sorantin, Georg Werkgartner, Reinhard Beichel, Alexander Bornik, Bernhard Reitinger, Nikolaus Popovic, and Milan Sonka. Virtual liver surgery planning. In Emanuele Neri, Davide Caramella, and Carlo Bartolozzi, editors, *Image Processing in Radiology*, Medical Radiology, pages 411 – 418. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-49830-8.

M Sparmann, B Wolke, H Czupalla, D Banzer, and A Zink. Positioning of total knee arthroplasty with and without navigation support a prospective, randomised study. *Journal of Bone & Joint Surgery, British Volume*, 85(6):830–835, 2003.

Axel Thomas Stadie, Ralf Alfons Kockro, Robert Reisch, Andrei Tropine, Stephan Boor, Peter Stoeter, and Axel Perneczky. Virtual reality system for planning minimally invasive neurosurgery. technical note. *Journal of neurosurgery*, 108(2):382 – 94, February 2008. ISSN 0022-3085.

Simon Steppacher, Jens Kowal, and Stephen Murphy. Improving cup positioning using a mechanical navigation instrument. *Clinical Orthopaedics and Related Research®*, 469(2):423–428, 2011. ISSN 0009-921X. doi: 10.1007/s11999-010-1553-8. URL http://www.springerlink.com/content/2356780723711174/abstract/.

A James Stewart. Vicinity shading for enhanced perception of volumetric data. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 47. IEEE Computer Society, 2003.

E. Stindel, JL Briard, P. Merloz, S. Plaweski, F. Dubrana, C. Lefevre, and J. Troccaz. Bone morphing: 3d morphological data for total knee arthroplasty. *Computer Aided Surgery*, 7(3):156 – 168, 2002. ISSN 1097-0150.

Bernd Stöckl, Michael Nogler, Rafal Rosiek, Martin Fischer, Martin Krismer, and Oliver Kessler. Navigation improves accuracy of rotational alignment in total knee arthroplasty. *Clinical orthopaedics and related research*, 426:180–186, 2004.

Danail Stoyanov, Mohamed ElHelw, Benny P Lo, Adrian Chung, Fernando Bello, and Guang-Zhong Yang. Current issues of photorealistic rendering for virtual and augmented reality in minimally invasive surgery. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 350–358. IEEE, 2003.

K. Subburaj, B. Ravi, and M. G. Agarwal. Automated 3d geometric reasoning in computer assisted joint reconstructive surgery. *2009 IEEE International Conference on Automation Science and Engineering*, pages 367 – 372, August 2009.

Erik Sundén and Timo Ropinski. Efficient Volume Illumination with Multiple Light Sources through Selective Light Updates. In *IEEE Pacific Visualization*, 2015. to appear.

Erik Sundén, Anders Ynnerman, and Timo Ropinski. Image Plane Sweep Volume Illumination. *IEEE TVCG(Vis Proceedings)*, 17(12):2125–2134, 2011.

F. Suykens and Y.D. Willems. Adaptive filtering for progressive monte carlo image rendering. In *WSCG 2000 Conference Proceedings*, 2000.

Gwen R J Swennen, Wouter Mollemans, and Filip Schutyser. Three-dimensional treatment planning of orthognathic surgery in the era of virtual imaging. *Journal of oral and maxillofacial surgery : official journal of the American Association of Oral and Maxillofacial Surgeons*, 67(10):2080 – 92, October 2009. ISSN 1531-5053.

L?szl? Szirmay-Kalos, Bal?zs T?th, and Mil?n Magdics. Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum*, 30(1): 85–97, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2010.01831.x. URL http://dx.doi.org/10.1111/j.1467-8659.2010.01831.x.

Russell Taylor and Leo Joskowicz. Computer-integrated surgery and medical robotics. *Standard Handbook of Biomedical Engineering and Design, 2nd edn, New York, NY: McGraw Hill*, pages 29–3, 2003.

Russell H Taylor, Janez Funda, Leo Joskowicz, Alan D Kalvin, Stephen H Gomory, AP Gueziec, and Lisa MG Brown. An overview of computer-integrated surgery at the ibm thomas j. watson research center. *IBM journal of research and development*, 40(2):163–183, 1996.

Z.A. Taylor, M. Cheng, and S. Ourselin. High-speed nonlinear finite element analysis for surgical simulation using graphics processing units. *Medical Imaging, IEEE Transactions on*, 27(5):650 – 663, 2008. ISSN 0278-0062.

D. van Antwerpen. Improving simd efficiency for parallel monte carlo light transport on the gpu. *Proceedings of High Performance Graphics 2011*, 2011.

Johan Van Cleynenbreugel, Filip Schutyser, Jan Goffin, Karel Van Brussel, and Paul Suetens. Image-based planning and validation of c1–c2 transarticular screw fixation using personalized drill guides. *Computer Aided Surgery*, 7(1):41–48, 2002.

Thisbe van Strien, Bart Kaptein, Arjan van Erkel, Edward Valstar, Rob Nelissen, et al. Computer assisted versus conventional cemented total knee prostheses alignment accuracy and micromotion of the tibial component. *International orthopaedics*, 33 (5):1255–1261, 2009.

Michael W. Vannier, Jeffrey L. Marsh, and James O. Warren. Three dimensional computer graphics for craniofacial surgical planning and evaluation. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '83, page 263–273, New York, NY, USA, 1983. ACM. ISBN 0-89791-109-1. doi: 10.1145/800059.801157. URL `http://doi.acm.org/10.1145/800059.801157`.

PP Vázquez, Eva Monclús, Isabel Navazo, et al. Representative views and paths for volume models. 2008.

E. Veach. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Citeseer, 1997.

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 419–428. ACM, 1995.

L.R. Wanger, J.A. Ferwerda, and D.P. Greenberg. Perceiving spatial relationships in computer-generated images. *Computer Graphics and Applications, IEEE*, 12(3):44–58, 1992.

Uriel Weinstein. Hard tissue surface geometry determination, March 21 2006. US Patent 7,014,461.

A. Westermark, S. Zachow, and B.L. Eppley. Three-dimensional osteotomy planning in maxillofacial surgery including soft tissue prediction. *Journal of Craniofacial Surgery*, 16(1):100, 2005. ISSN 1049-2275.

D White, KL Chelule, and BB Seedhom. Accuracy of mri vs ct imaging with particular reference to patient specific templates for total knee replacement surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 4(3):224–231, 2008.

Lance Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274, 1978.

E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth. Techniques used in the gem code for monte carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Conf. Applications of Computing Methods to Reactor Problems*, page 557, 1965.

C. Wyman, S. Parker, P. Shirley, and C. Hansen. Interactive display of isosurfaces with global illumination. *IEEE Transactions on Visualization and Computer Graphics*, pages 186 – 196, 2006. ISSN 1077-2626.

Chunming Xie, Kai Liu, Luxin Xiao, and Rong Tang. Clinical outcomes after computer-assisted versus conventional total knee arthroplasty. *Orthopedics*, 35(5):368, 2012.

Kai Xie and Y. Zhu. Interactive surgery simulation for the nose augmentation using ct data. *Neural Computing &amp; Applications*, 19:61–65, 2010. ISSN 0941-0643. URL `http://dx.doi.org/10.1007/s00521-009-0245-3`.

Guanglei Xiong and Charles Taylor. Virtual stent grafting in personalized surgical planning for treatment of aortic aneurysms using Image-Based computational fluid dynamics. In Tianzi Jiang, Nassir Navab, Josien Pluim, and Max Viergever, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, volume 6363 of *Lecture Notes in Computer Science*, pages 375–382. Springer Berlin / Heidelberg, 2010. ISBN 978-3-642-15710-3. URL `http://www.springerlink.com/content/lw4np565ph469485/abstract/`.

R. Xu and S.N. Pattanaik. Non-iterative, robust monte carlo noise reduction. *IEEE Computer Graphics and Applications*, 25(4):31–35, 2005.

J. Yamanaka, S. Saito, and J. Fujimoto. Impact of preoperative planning using virtual segmental volumetry on liver resection for hepatocellular carcinoma. *World journal of surgery*, 31(6):1251 – 1257, 2007. ISSN 0364-2313.

Ziv Yaniv and Leo Joskowicz. Precise robot-assisted guide positioning for distal locking of intramedullary nails. *Medical Imaging, IEEE Transactions on*, 24(5):624–635, 2005.

S. Zachow, H.C. Hege, and P. Deuflhard. Computer assisted planning in cranio-maxillofacial surgery. *Journal of Computing and Information Technology*, 14(1):53, 2004. ISSN 1846-3908.

Stefan Zachow, Hans-Christian Hege, and Peter Deuflhard. Computer assisted planning in cranio-maxillofacial surgery. *CIT. Journal of computing and information technology*, 14(1):53–64, 2006.

Sergey Zhukov, Andrei Iones, and Grigorij Kronin. An ambient light illumination model. In *Rendering Techniques 98*, pages 45–55. Springer, 1998.

# List of Figures

# List of Tables

# Summary

This thesis revolves around the development of medical visualization tools for the planning of CSG-based surgery. To this end, we performed an extensive computer-assisted surgery (CAS) literature study, developed a novel optimization technique for customizable surgical guides (CSG), and introduce three visualization techniques to make the planning more realistic and allow for remote visualization.

In Chapter 2 we document the results of an extensive overview study, in which the use of visualization in CAS is analysed. We collected a comprehensive database of visualization relevant CAS publications, and analyse the visualization techniques that are used. We also classify important CAS-related surgical tasks and explain how and why visualization is used. Further, we analysed how surgical plans are transferred to the operating theater. Finally, we discuss how visualization is used in the four most prominent application areas of CAS. Based on this review, we were able to pinpoint interesting new research directions. One of these is the apparent lack of proper tools for CSG-based surgery, a challenge that we addressed in Chapter 3.

The optimization of CSG parameters such that the CSG can be docked on bone in an accurate and stable way, is important in the planning of CSG-based surgery. The adjustable nature of the CSG, which allows it to become patient-specific, unfortunately also makes it inherently unstable. Optimizing the configuration by hand leads to poor results as we demonstrated with experiments. In Chapter 3, we therefore solve the problem in sillico. We described a novel planning tool that is able to automatically optimize a CSG for an arbitrary patient. We established this by combining a physical simulator, which models the physical interaction between the CSG and the bone, with a genetic optimization process. With experiments, we were able to prove that our optimization tool produces CSG configurations that lead to accurate and stable docking.

In Chapter 4, we address the challenge of enhancing the planning environment with appropriate visualization techniques that help to understand how a CSG is connected to the bone. The state-of-the-art rendering tools in CAS applications are not able to accurately and effectively communicate how the CSG attaches to the bone. However, *ambient occlusion* (AO) is an illumination technique that is particularly effective at depicting contact between objects, but is generally computationally expensive. Therefore, we developed an efficient version of this algorithm such, that it can be used in the planning pipeline to effectively depict CSG-bone contact.

We took the visualization one step further by introducing photo-realistic and physically based volume rendering. Chapter 5 describes *Exposure Render*, a complete volume rendering framework based on stochastic raytracing, and is able to incorporate a host of otherwise difficult to obtain photorealistic camera, light, and material effects. It is a well known fact that these help to understand shape, depth and size. Therefore,

we employed *Exposure Render* to build a prototype doctor-patient communication system. With this remote visualization system, a doctor can counsel a patient from a distance, or a patient can perform self health management by uploading their tomographic data.

In Chapter 6 we optimize the performance of *Exposure Render*. We introduce *visibility sweeps*, an efficient method to compute and store visibility information in volume data sets. With this method, it becomes possible to efficiently query approximate global visibility information in a volume data set. We demonstrate that this visibility information can be harnessed to improve the efficiency of the ray sampling processes in *Exposure Render*, which results in faster convergence. Though we demonstrate the effectiveness of *visibility sweeps* in the context of stochastic volume rendering, its use stretches beyond this application. Many areas of medical visualization and CAS rely on visibility information, such as automatic view finding in volume data and in various areas of CAS e.g., access, resection and implant planning. In our project it is also relevant because the visibility information can be used to make the physical simulator more realistic, for instance by avoiding docking trajectories that are associated with high risk of tissue damage.

The research described in this thesis was part of the project *Novel pre-operative planning and intraoperative guidance system for shoulder replacement surgery (10812)*, funded by the Dutch Technology Foundation.

# Samenvatting

Dit proefschrift beschrijft de ontwikkeling van nieuwe medische visualisatie tools voor het plannen van gewrichtsvervangende operaties, met behulp van customizable surgical guides (CSG), oftewel personaliseerbare richt instrumenten. Om dit te bewerkstelligen hebben we eerst uitvoerig bekeken hoe computers tegenwoordig worden gebruikt om operaties te plannen en uit te voeren. Verder hebben we een nieuwe optimalisatie techniek ontwikkeld voor het automatisch instellen van CSGs. Tevens introduceren we nieuwe visualisatie technieken die de planning realistischer maken.

In Hoofdstuk 2 laten we de resultaten zien van een uitgebreide studie naar het gebruik van computer visualisatie in computer assisted surgery (CAS). Dit heeft geresulteerd in een grote collectie van publicaties. We hebben deze publicaties op meerdere manieren geanalyseerd. We beginnen met een overzicht van de meest gebruikte visualizatie technieken. Daarnaast analyseren we de rol van visualizatie voor het uitvoeren van de belangrijkste chirurgische taken. Tevens bestuderen we hoe met behulp van CAS een chirurgisch plan wordt ingezet in de operatie kamer. Tot slot nemen we vier prominente toepassings gebieden onder de loep, en kijken hoe visualisatie daar wordt gebruikt. Op basis van dit werk konden we vaststellen dat er weinig oplossingen zijn voor het plannen van gewrichtsvervangende operaties met behulp van personaliseerbare richt instrumenten. Hoofdstuk 3 gaat hier dieper op in.

Het instellen van personaliseerbare richt instrumenten, zodanig dat deze accuraat en stabiel kunnen worden vastgemaakt, is belangrijk tijdens het preopreratief plannen van dit soort gewrichtsvervangende operaties. De instelbaarheid van het personaliseerbare richt instrument, die een patient specifieke plaatsing mogelijk maakt, heeft helaas ook tot gevolg dat het lastiger wordt om het richtinstrument stabiel en accuraat te plaatsen op bot. Het handmatig optimaliseren van de instellingen bleek een onmogelijke opgave, daarom maken wij hiervoor gebruik van een computer algoritme. In Hoofdstuk 3 beschrijven wij planning software die in staat is om personaliseerbare richt instrumenten automatisch in te stellen. Om dit te bewerkstelligen, combineren wij een simulator, die de fysische interactie tussen instrument en bot modelleert, met een genetische optimalisatie. We tonen met experimenten aan dat onze software in staat is om de richt instrumenten accuraat in te stellen.

In Hoofdstuk 4 verrijken we onze planning software met nieuwe visualisatie technieken die een gebruiker van ons systeem helpen bij het inspecteren van de optimalisatie resultaten en de planning. De belichtingstechnieken die op dit moment gangbaar zijn, zijn onvoldoende om op een effectieve en intuitieve manier het contact van het richtinstrument met het bot weer te geven. Ambient occlusion daarentegen is zeer toepasselijk, maar vereist veel rekenkracht. Omdat deze techniek erg effectief is hebben wij daarom een efficiente variant ontwikkeld waarmee het contact tussen richt instrument en bot goed kan worden gevisualiseerd.

We hebben ook aandacht besteed aan het fotorealistisch weergeven van volumetrische data sets. Hoofdstuk 5 beschrijft *Exposure Render*, een complete systeem voor het realistisch weergeven van volumetrische data sets, op basis van stochastische raytracing. Dit systeem stelt ons in staat om zeer complexe, en anderzijds bijna onmogelijke visuele effecten te genereren. Deze effecten, zoals schaduw en reflectie, zijn tevens onmisbaar voor het waarnemen van vorm en diepte. We gebruiken *Exposure Render* daarom ook om de communicatie tussen doktor en patient te verbeteren en om patient aan te moedigen zelf online zijn data te bekijken.

In Hoofdstuk 6 maken we *Exposure Render* efficienter, op deze manier worden afbeeldingen van de volumetrische data sets sneller op het scherm weergegeven. Om het raytracen van volumetrische data sets efficienter te maken, gebruiken we zichtbaarheids informatie, afgeleid van de volumetrische data set, om het schieten van rays te optimaliseren. We introduceren *visibility sweeps*, een method om efficient zichtbaarheids informatie te kunnen berekenen en opslaan. Met behulp van deze methode kan de zichtbaarheid op een willekeurige positie in het volume worden opgevraagd. Ondanks dat we de effectiviteit aantonen in de context van rendering, zijn visibility sweeps ook bruikbaar in andere toepassingsgebieden. Bijvoorbeeld voor het automatisch uitlijnen van camera's op basis van zichtbaarheid van voxels en het plannen van diverse chirurgische taken.

Het onderzoek dat gepresenteerd is in deze thesis was onderdeel van een groter project *Novel pre-operative planning and intraoperative guidance system for shoulder replacement surgery (10812)*, en is gesubsidieerd door the Stichting voor de Technische Wetenschappen.