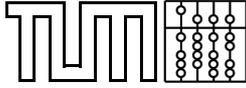


Focus and Context Visualization for Medical Augmented Reality

Felix Wimmer

Diplomarbeit in Informatik



Focus and Context Visualization for Medical Augmented Reality

Diplomarbeit in Informatik

Diplomand: Felix Joseph Wimmer

Ort: Technische Universität München, Deutschland
Fakultät für Informatik
Lehrstuhl für Informatikanwendungen in der Medizin & Augmented Reality

Aufgabensteller: Univ.-Prof. Dr. Nassir Navab

Betreuer: Christoph Bichlmeier

Abgabedatum: 10. August 2007

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Datum, Unterschrift

Abstract

This diploma thesis concerns focus and context visualization of anatomical data (obtained from different imaging modalities) for medical augmented reality (AR) and thereby the correct fusion of real and virtual anatomy. Medical AR is intended to be used for pre-operative diagnosis, intra-operative guidance and post-operative control, but is still in the stadium of research and was not practically applied yet. It is a technique which augments the surgeon's real view on the patient with virtually visualized anatomy of the patient's body. A medical AR system used for the purpose of focus and context visualization includes a tracking system and a video see-through, head-mounted-display (HMD) enabling a stereoscopic, augmented view on the AR scene.

Focus refers to the part of the virtual anatomy the observer (surgeon) is interested in, e.g. the operation site, which has to be perceived at the correct location respective the context of the real skin. In a possible future application of medical AR for surgical interventions, correct perception of position and depth of the focus has to be guaranteed. If the perception of the focus is disturbed or misleading, the danger exists that the surgeon operates at the wrong location and thus vitally important organs of the patient are hurt. Many visualization approaches for medical AR suffer from a misleading depth perception, since the normally hidden interior anatomy is just superimposed on the patient's body. In these approaches the virtual anatomy seems to be located in front of the human body. Partial occlusion of the virtual anatomy by the real skin can solve this problem. Within this diploma thesis further methods for improving the perception of layout (arrangement) and distances of objects in the AR scene are discussed. Visual cues for the perception of layout and distances of focussed virtual anatomy can be enabled by the exploit of context information. Context information can be provided as well by a correct integration of the camera image, recorded by the color cameras mounted on the HMD, as by non-focus parts of the virtual anatomy.

Within the scope of the practical work of this thesis a focus and context visualization framework for medical AR was implemented, which considers and exploits depth cues enabling a correct perception of the focussed virtual anatomy. Therefore general principles and methods for creating and designing focus and context visualizations are taken into account, which are mainly adapted from hand-made illustration techniques. The frame-

work provides a correct fusion of real and virtual anatomy and realizes an intuitive view on the focussed anatomy of the patient. It includes a new technique for modifying the transparency of the camera image of the real body. The transparency is thereby adjusted by means of properties (e.g. curvature) of a virtual skin model. Additionally, a method for clipping parts of the anatomy, hindering the view onto the focus, is introduced. The framework also contains methods for integrating surgical or endoscopic instruments into the medical AR scenario. Instruments are virtually extended as soon as they penetrate into the patient's body. Moreover, the penetration port is highlighted and virtual shadows are used to provide visual feedback from instrument interaction.

The effectiveness of the developed techniques is demonstrated with a cadaver study and a thorax phantom, both visualizing the anatomical region of the upper part of the body, and an in-vivo study visualizing the head.

Contents

1	Introduction	1
2	Augmented Reality	5
2.1	Defintion and General Overview	5
2.2	Medical Augmented Reality	7
3	The Components of a Medical Augmented Reality System	9
3.1	Stereoscopic Head-Mounted Display	11
3.2	Tracking System	12
3.2.1	Optical Outside-In Tracking	12
3.2.2	Single Camera Inside-Out Tracking	13
3.2.3	Merging The Tracking Data	14
3.3	Software Components	15
4	Focus and Context Visualization: Introduction and History	17
4.1	Focus and Context	18
4.2	Focus and Context in Scientific Illustrations	19
4.2.1	History of Hand-Made Scientific Illustrations	20
4.2.2	Modern Illustration with Computer Graphics	21
5	Perception of Distances, Layout and Shape of Objects	23
5.1	Introduction to Visual Perception	24
5.2	Visual Cues for Perceiving Layout and Distances	26
5.2.1	Oculomotor Depth Cues	27
5.2.2	Monocular Depth Cues	28
5.2.3	Motion Induced Depth Cues	34
5.2.4	Stereoscopic Depth Cues	37
5.2.5	Summary of Depth Cues	38
5.3	The Usage of Transparency for the Perception of Layout and Distances	38
5.4	Perception of Shape	40
6	Principles of Designing Focus and Context Visualizations of Volumetric Data	43
6.1	Visualization of Volumetric Data	44
6.1.1	Indirect Volume Rendering	44
6.1.2	Direct Volume Rendering	45

6.2	Extraction of Focus and Context Layers / Objects	47
6.2.1	Segmentation of the Data Set prior to Rendering	47
6.2.2	Extraction of Layers during Rendering	48
6.3	Shading	48
6.3.1	Blinn-Phong Shading	49
6.3.2	Tone-Based Shading	50
6.3.3	Contour Rendering	52
6.3.4	Curvature-Based Shading	53
6.3.5	Solid Texturing	56
6.4	Modulation of Context Layers	56
6.4.1	Cut-away techniques	56
6.4.2	Ghosting: Creating a Semi-Transparent Context	59
6.4.3	Exploded Views and Context Deformations	66
6.5	Designing Focus and Context Visualization for medical AR	67
7	Results: A Focus and Context Visualization Framework for Medical Augmented Reality	69
7.1	Rendering and Shading of the Virtual Anatomy	70
7.2	Anatomy and Studies	70
7.2.1	Phantom	70
7.2.2	Cadaver Study	71
7.2.3	In-Vivo Study	71
7.3	Ghosting of the Camera Image: Fusion of Real and Virtual Anatomy	72
7.3.1	Curvature	74
7.3.2	Angle of Incidence Factor	75
7.3.3	Distance Falloff	76
7.3.4	Final Transparency/Opacity Determination	77
7.3.5	Highlighting the Border of the Semi-Transparent Domain	77
7.3.6	Virtual Window to the Inside of the Patient	78
7.4	Cut-away by Volume Clipping	79
7.4.1	Setting up Form, Size and Position of the Clipping Volume	80
7.4.2	Generation of the Clipping Depth Map	81
7.4.3	Clipping of Virtual Objects	81
7.4.4	Clipping of the Camera Image	81
7.4.5	Displaying and Texturing of the Clipping Volume	81
7.4.6	Clipping by a Clipping Depth Map vs. Standard Clipping Planes	82
7.5	Ghosting of Virtual Anatomy	83
7.6	Shadows	83
7.7	Integration of Surgical Instruments	85
7.7.1	Virtual Extension of Instruments	86
7.7.2	Visual Feedback due to Shadow Effects	86
7.7.3	Highlighting the Penetration Port of the Instrument	88

7.7.4	Light surrounding the Instrument Axis	90
7.8	Visualization Pipeline	91
8	Conclusion and Further Work	93
9	Acknowledgements	95
	Bibliography	97

1 Introduction

Augmented reality (AR) is a field of research that concerns the fusion of virtual and real worlds. Virtually generated objects are thereby embedded into an image of the real scene. These objects are intended to provide additional information to the viewer and are integrated in a preferably smooth and intuitive way. Besides the use of AR for industrial applications (e.g. in the automobile industry), this technology can also be utilized for medical purposes. Medical augmented reality is still in the stadium of research and was not practically applied yet. The main research objectives of medical AR are the use of AR technology for pre-operative diagnoses (surgical planning), intra-operative guidance and navigation and for post-operative control.

Medical AR is a technique that is intended to be applied for augmenting the surgeon's real view on the patient with the patient's interior anatomy. The anatomical data can be obtained from multimodal scans of the body, e.g. scans taken from computer tomography (CT) or magnetic resonance imaging (MRI). The usage of a tracking system makes it possible to visualize the virtual anatomy on its precise position and in its proper size, respective the real body. Exact localization of the body and a high degree of accuracy when registering the medical imaging data with the tracked patient, is an important task for superimposing the virtual anatomy and the real body. The real body is recorded with two cameras, which are mounted above each eye on a head-mounted-display (HMD), worn by the observer (surgeon), and thus simulate the eyes' view. A computer augments the camera images with the virtual anatomy and sends it back to the HMD, where the final images are presented on displays in front of each eye providing a stereoscopic view on the scene. Instead of opening the body completely, the application of AR technology in surgeries improves minimal invasive interventions, resulting in less violation of the patient's body. Moreover, a clear view onto parts of the anatomy, that are hardly visible in conventional surgery, can be provided.

The thesis discusses focus and context visualization of the patient's anatomical data in conjunction with the camera images of the patient's body. Focus refers to the part of the virtual anatomy the observer (surgeon) is interested in, e.g. the operation site. In addition it is important to display context information to tender a correct perception of layout and distances of objects within the focus region. Context information can be provided by virtual anatomy positioned beside or in front of the focus region. In addition to the virtual context objects, the camera image of the real body represents a special context layer, which does not exist in purely virtual visualization of medical data.

Displaying virtual anatomy in the context of the "real" camera image and the correct perception of layout and distances of the virtual anatomy respective the real body, is a special

challenge in medical AR. Although tracking enables the determination of the correct position and size of the virtual models in respect with the real body, simple superimposition of the virtual anatomy onto the camera image still causes a misleading depth perception of the virtual anatomy. Due to occlusion (a powerful depth cue) of the real body by the virtual anatomy, the virtual anatomy is perceived in front of the body. Correct perception of layout and distances is extremely important in medical AR, for instance for accurate navigation of surgical instruments. Incorrect estimation of distances and depth of instruments and anatomy can cause perilous damages and serious consequences for the patient. In this thesis methods are presented that overcome the misleading depth perception and furthermore enable an improved perception of layout and distances.

Focus and context visualization of anatomic data for medical AR has to fulfil the following conditions:

- The visualization has to provide a non-restricted view on the anatomical region of interest (the focus), which includes concealing obstructing anatomy and emphasizing the operation site.
- The visualized imaging data has to be integrated in such a way that positions and relative and absolute distances of objects within the AR scene can be perceived intuitively.
- For using the AR technology intraoperatively, augmented surgical instruments have to be integrated to enable the observer to perceive relative distances between anatomy and instrument and to get visual feedback due to this interaction.

Chapter 2 gives an introduction into AR and discusses its possible medical applications. Chapter 3 describes the components of a medical AR system, exemplary the system used at the "Chair for Computer Aided Medical Procedures and Augmented Reality" (CAMPAR). An introduction into the concepts of focus and context visualization is presented in Chapter 4. Perception of layout, distances and shape of objects is discussed in Chapter 5. Chapter 6 provides general principles of designing focus and context visualizations, which are partially adapted from hand-made illustration techniques. In Chapter 7 a focus and context visualization framework for medical AR is presented, which was developed within the scope of this diploma thesis as part of the NARVIS (navigation- and AR-visualization) project at CAMPAR. For the design of this framework cognitions obtained from the Chapters 5 and 6 were exploited in order to fulfil the three conditions mentioned before.

The framework includes a method for smooth integration of the camera image by modifying the transparency of its pixels according to the properties of the skin surface. A method for clipping anatomy, which hinders the view onto the focus, is also provided. Moreover, the framework contains techniques to integrate surgical or endoscopic instruments into the medical AR scenario. These enable the augmentation of the instruments inside the body and provide visual feedback about relative distances between entities, e.g. by the use of artificially created soft shadows. The framework was evaluated with a ca-

daver and an in-vivo study, which show the effectiveness of the implemented visualization methods.

2 Augmented Reality

In general, AR describes an extension of the view of the real environment of humans with computer-generated information, and can be considered as a form of human-computer interaction. The following section explains the term AR and the functional principles of AR systems in detail. Furthermore it gives an overview of different fields of application and discusses primarily its usage for medical and surgical purposes.

2.1 Definition and General Overview

In the scientific literature varying definitions for the term AR exist. Generally, three different classes of definitions can be distinguished [63]:

The first class of definition constricts the application of the term AR to the extension of the human visual perception by the usage of HMD's, which enable a "see-through" view of the real world upon which computer generated graphics are superimposed [63].

The second class "relaxes the constraint of needing the equivalent of a HMD and covers any case in which an otherwise real environment is 'augmented' by means of virtual (computer graphic) objects, thereby encompassing both large screen and monitor based displays as well" [63].

A third class completely detaches the term from the usage of specific display devices. It embraces any mixture of real and virtual environment and defines AR as a system having the following three characteristics [3]:

- **Combination of real and virtual world:** The user is located in a real environment and gets needed virtual information for every specific situation.
- **Interactivity and real time capability:** The user can influence the real environment as well as the virtual objects and the system has to react on user actions in a defined and preferably short time.
- **Registration in three dimensions:** Registration means the synchronisation of virtual objects and reality. It requires an exact and perspective correct positioning of the virtual 3D-objects.

Unlike *virtual reality* (VR), the technology of AR has not found widespread use in industry, yet. The term VR describes a three dimensional, computer generated environment, which can be used interactively by the user. The fundamental difference between AR and VR lies in the inclusion of the reality. Goal of VR is to create a preferably detailed and

complete replication of the reality by a computer generated world, whereas AR aims for integration of virtual information into reality. Milgram et al. [64] describe AR in the context of a smooth transition from a real environment to an absolute virtual environment (see Figure 2.1), denoted as reality-virtuality continuum. This continuum is split into four sections:

- **Real Environment:** completely unmodelled
- **Augmented Reality:** the real environment is augmented with virtual objects (partially modelled)
- **Augmented Virtuality:** the virtual environment is augmented with real objects (partially modelled)
- **Virtual Environment:** must be completely modelled in order to be rendered

The real and the virtual environment represent the poles of the continuum. Everything between these poles is called *mixed reality* (MR), which can be subdivided into AR and *augmented virtuality* (AV). Whereas AR focusses on perception of the reality, the main focus of AV lies on the perception of virtuality (compare Figures 2.2). The technology of AR

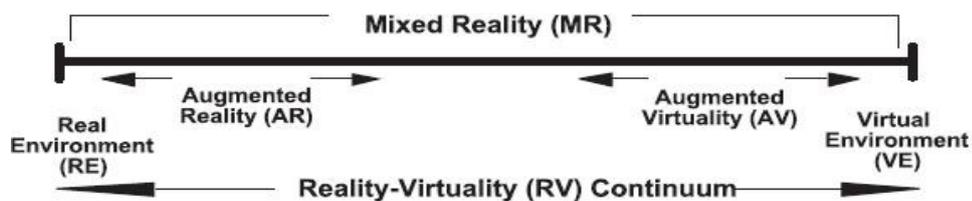


Figure 2.1: The differentiation of Augmented Reality and Augmented Virtuality in the context of the Reality-Virtuality Continuum [64]



Figure 2.2: Illustrations of (a) Augmented Reality (AR) and (b) Augmented Virtuality (AV) [63]

is based on the provision of additional information within the user's field of view. This

computer generated virtual information is superimposed on visual sensation of the real environment and perceived simultaneously with real information.

MR and AR are highly interdisciplinary fields involving signal processing, computer vision, computer graphics, user interfaces, human factors, wearable computing, mobile computing, computer networks, distributed computing, information access, information visualization and hardware design for displays and sensors. MR and AR concepts can be used in a wide range of applications, e.g.:

- Medicine (see Section 2.2)
- Manufacturing and product development [2]
- Industrial training [94]
- Architecture / construction industry [50]
- Computer games [79]
- Hand-held computers with AR functionality using an attached camera [24]

The majority of these applications were not productively applied yet and are still in the stadium of prototypes. The reason for that are fundamental problems including tracking precision, misleading depth perception (see Chapter 5), ergonomics of HMDs, poorly conceived user interfaces, or the lack of robust and mobile computer systems.

The IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR) forms the most important annual international conference for MR and AR.

2.2 Medical Augmented Reality

For quite some time computer technology is used for patient examination, preparation and performance of surgeries. Computer tomography (CT), magnetic resonance imaging (MRI) and other imaging data allow the generation of three dimensional images of the anatomy of the patient. The application of AR technology enables the utilization of such imaging data for the virtual augmentation of a patient's body with its interior anatomy. One field of application is the augmentation of endoscopic camera images. For the performance of minimal invasive (key hole) surgeries, endoscopes are used, which record video images of the inside of human bodies by utilizing miniaturized cameras. AR allows to superimpose these images with rendered three dimensional imaging data in order to visualize hidden structures like tissue, bones and organs. Despite of augmenting the real image, the surgeon here faces the problem to simultaneously keep an eye on the display device (e.g. a video monitor) and the patient. The usage of a HMD for displaying and recording of images however enables a direct augmentation of the surgeon's view with virtual anatomy. This in-situ visualization makes it possible to display viscera and bones

at their exact position and in their correct size even without surgical intervention. Medical AR using a HMD can thus be applied as well for pre operative diagnosis, intra operative guidance as post operative control. The direct augmentation of the view of the surgeon is also denoted as "X-Ray-Vision". It is able to supply minimal invasive surgery, implying a reduction of operating time, less pain for the patient, a diminished duration of hospitalization and reduction of costs.

For a medical and surgical application, a medical AR system has to fulfill the following conditions:

- **High accuracy:** The tracking system has to determine the exact location of the patient's body in order to display the virtual anatomy at its correct position in the view of the surgeon. This condition also concerns computational accuracy and rapidness of synchronization of the virtual and real imagery.
- **Real time capability:** The AR system always has to work in real time so that the surgeon's work is not disturbed by a low framerate. This condition has to be basically fulfilled by fast computer hardware and efficiently programmed software.
- **Reliability and robustness:** The AR system must not fail during an operation and has to work reliably under all conditions which could possibly occur in the operating room.
- **Correct perception of layout and distances of virtual objects in the AR scene:** The avoidance of a misleading depth perception of virtual objects is one of the most important issues in medical AR (see Chapter 5 for further information). Correct perception of layout and distances has to be assured by the rendering system.

Current medical AR systems often do not fulfill these criteria entirely and are thus still in stadium of prototypes. These conditions require good algorithms for tracking, segmentation and registration, high speed cameras with high resolution, broadband networking and a fast computer hardware including fast graphic processors. Applying a system which does not satisfy any of these criteria could be life-threatening for the patient.

Moreover, virtual representation of viscera is subdued to some limitations due to their physical properties. Parts of the viscera change their position, shape and size (e.g. the heart). In order to visualize such anatomy properly, the body has to be scanned all the time, which is not possible with state-of-the-art technology. Rendering of those parts of viscera thus can not fulfill the condition of high accuracy. In contrast to these parts of viscera, bones for instance do not change much their position or size. Thus dorsal surgery is a reasonable field for medical AR applications.

The following chapter discusses the necessary hardware and software components of a medical AR system.

3 The Components of a Medical Augmented Reality System

This chapter explains the hardware and software components, which are necessary for realising a medical AR system. The components of the stereoscopic AR system used at CAMPAR, denoted as navigation- and AR-visualization system (NARVIS), are thereby discussed in detail. Generally an AR system consists of

- a **tracking system**,
- a **computer with a software system** for rendering the virtual anatomy graphics with computer graphics, for computation and inclusion of tracking data and for synchronizing and combining imagery of virtual and real entities,
- **tracking targets**, e.g. the patient and surgical instruments,
- and **display devices**.

The tracking system enables to visualize the virtual anatomy on its precise position and in its proper size, respective the real body of the patient. Therefore retro reflective fiducial markers have to be attached to the body during scanning and tracking. The markers have to be extracted from the scanned anatomical data set and registered to the respective tracking positions of the markers. Exact tracking and thus localisation of the markers and a high degree of accuracy when registering the medical imaging data with the patient, is a very important task for superimposing the virtual anatomy and the real body. In general, two classes of display devices can be applied for AR: see-through HMDs [78] and monitor-based devices. See-through HMDs can be split into two further classes: video-see-through and optical-see-through HMDs. Section 3.1 discusses the different types in detail. Monitor-based devices can be subdivided into fixed and mobile display screens.

In the following the medical AR system of CAMPAR is presented. It includes the following hardware components:

- Optical outside-in tracking system developed by the company A.R.T. GmbH (Weilheim, Germany), which consists of four cameras mounted on a framework under the ceiling of the CAMPAR laboratory to track targets within the workspace of the surgeon. A target has to be visible by at least two cameras in order to make it trackable.
- Stereoscopic video see-through HMD with a display resolution of 1024 * 768

3 The Components of a Medical Augmented Reality System

- Inside-out tracking system which includes a single infrared camera mounted on the HMD
- Personal computer with the following specification: CPU with 3.20 GHz, 1.8 GB RAM, NVIDIA Quadro FX 3400/4400

The following software components are used by the medical AR system of CAMPAR:

- CAMPAR software framework: It provides functionality to develop software for medical image processing, AR, volume rendering and geometric calculations
- RAMP: Software for inside-out tracking performed by the HMD and the tracking target
- A.R.T. tracking software for optical outside-in tracking
- Microsoft Visual Studio .NET 2005 as integrated development environment (IDE)
- Amira: Software used for segmentation and conversion of medical data types

An schematic overview of the medical AR of CAMPAR is illustrated in Figure 3.1. This

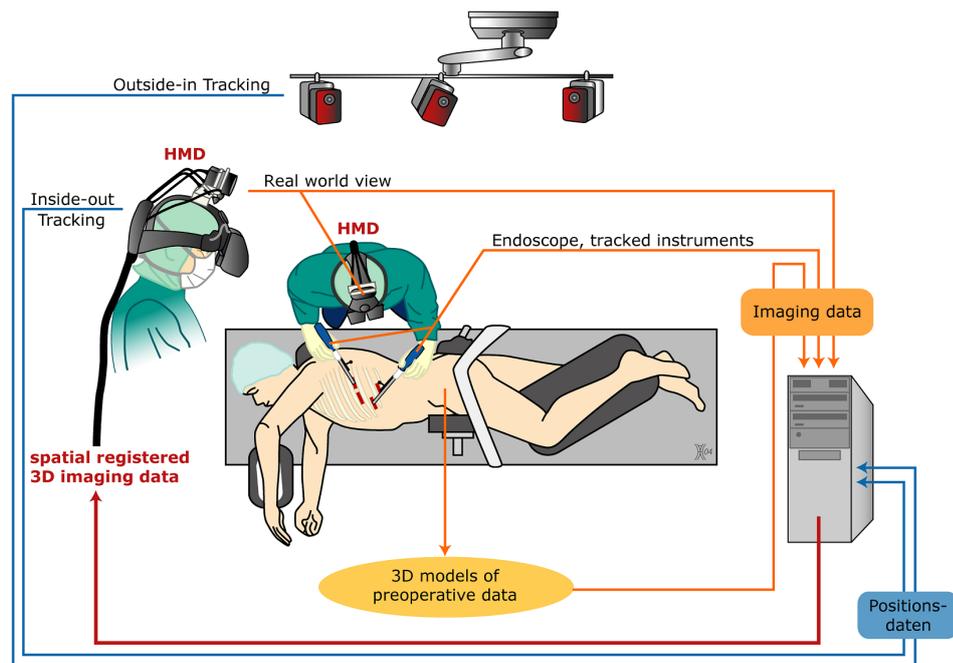


Figure 3.1: Schematic overview of the AR system of CAMPAR, including an optical tracking system and a stereoscopic, video see-through head mounted display.

system was used to generate the focus and context visualization system for medical AR,

presented in Chapter 7. Its components are discussed in the following sections in more detail.

3.1 Stereoscopic Head-Mounted Display

The medical AR system of CAMPAR uses a stereoscopic video see-through HMD (see Figure 3.2) presented by Sauer et al. [70]. Two displays are attached in front of each eye of the carrier of the HMD. Additionally two color cameras and an infrared camera (for single camera inside-out tracking, see Section 3.2.2) are mounted. The color cameras simulate



Figure 3.2: HMD with attached color cameras and an infrared camera for inside-out tracking.

the eyes' view and the displacement of the two eyes. Their directions of view cross at a predefined and fixed focus, approximately one meter in front of the user. The images recorded by the color cameras are augmented with virtual objects by a computer and sent back afterwards to the attached displays. Thus, the carrier of the HMD has an stereoscopic, augmented view on the scene lying in front of him.

Due to using two cameras and two displays to simulate the eyes' view, the virtual augmentations have to be rendered two times. However, compared to rendering only a single camera image on a monitor, it has the advantage of a intuitive stereoscopic in-situ visualization of medical imagery on the patient. This stereoscopic view enables a better estimation of shape, layout and relative and absolute distances of objects than monocular

displaying can offer. More precisely, stereoscopic perception of a scene provides a depth cue, denoted as *binocular disparity*, which is discussed in section 5.2.4 in detail.

Alternative to the usage of a video see-through head mounted display, an optical see-through HMD can be utilized. Despite of using video cameras and displays, the optical see-through HMD projects the rendered virtual models onto a partially transparent mirror placed in front of the observers eyes. On the one hand, the real environment can be perceived through the mirror without latency, limitations in resolution and colors of the display. On the other hand, the displaying quality of the virtual objects suffers from technical restrictions. The differences between the real scene and the virtual objects are even more artificial, leading to a misleading depth perception which can hardly be resolved. Moreover, imagery of the real world cannot be synchronized with the visualization of virtual entities. There are two major advantages of using a video see-through system for visualization. First, real and virtual imagery can be optimally synchronized to avoid time lags between the images of the camera, which would lead to undesirable - and for the user fatiguing - effects like "perceivable jitter or swimming" [71]. Second, the system allows for more options on how to combine real and virtual imagery, like occluding real objects, since we have full control over the real images. Optical systems offer only a brightening augmentation.

The infrared camera mounted additionally on the HMD is used for inside-out tracking, which is described in section 3.2.2.

3.2 Tracking System

In order to visualize the virtual objects at the correct location respective the recorded camera images of the video cameras mounted on the HMD, the relative positions of the tracking targets (like the patient's body and surgical instruments) and the carrier of the HMD (the surgeon) have to be determined. The AR system used at CAMPAR, realizes this issue by means of two synchronized tracking systems: a single camera inside-out and an optical outside-in tracking system. The outside-in tracking system calculates the absolute positions of the tracking targets, whereas the inside-out tracking system enables the determination of the relative position of the carrier of the HMD respective the tracking targets.

3.2.1 Optical Outside-In Tracking

The optical outside-in tracking system, developed by the company A.R.T. GmbH (Weilheim, Germany), is used to track targets like surgical instruments or the patient. To this end, four infrared cameras are fixed on a framework under the ceiling, which cover a large working area ($3 \times 3 \times 2$ m). As mentioned at the beginning of this chapter, retro reflective fiducial markers are attached to all targets, which can be tracked by this system. To recover the six degrees of freedom of motion of a rigid body, the external optical tracking system requires at least four rigidly attached markers. The markers of a target have to be

perceived at least of two cameras in order to reconstruct 3D information from 2D information via epipolar geometry [93]. A reference frame in the form of an arc, which is tracked by this system, has an exceptional function, as it is also tracked by the inside-out tracking system (see section 3.2.2). It enables the synchronization between the inside-out and the outside-in tracking systems, because both tracking systems calculate a coordinate system for it. Figure 3.3 provides a view on the observer wearing the HMD, two tracking targets (a thorax phantom and a surgical instrument), the reference frame and one of four infrared cameras of the outside-in tracking system. Once this system is calibrated, it provides sta-

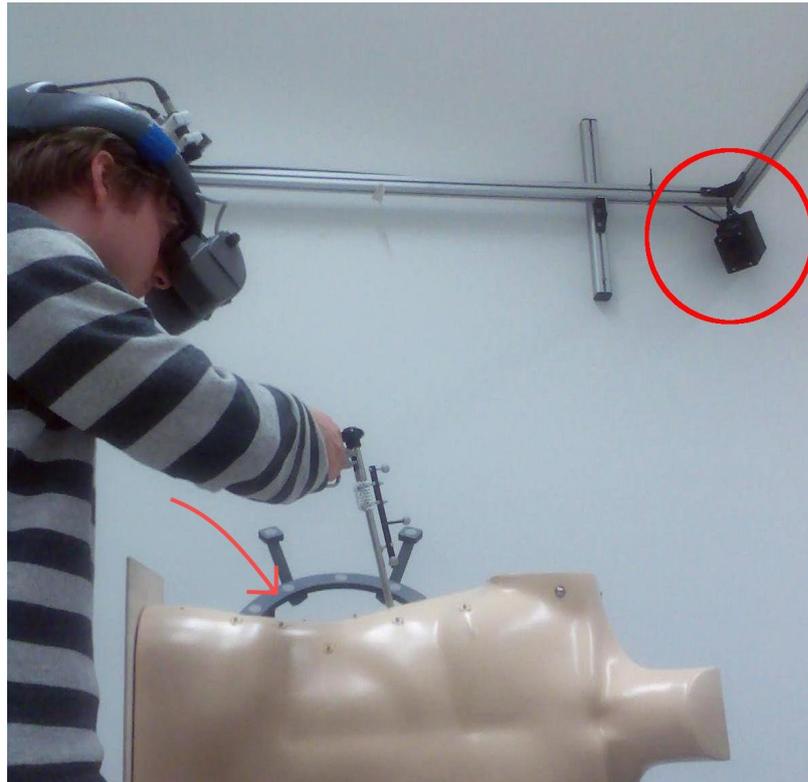


Figure 3.3: The reference frame (see inserted red arrow), a thorax phantom and a surgical instrument are tracked by the infrared camera attached to a framework above the scene (see inserted red circle).

ble tracking data for a long term unless the cameras are moved. The system is capable of tracking the targets with an accuracy of $0.35mm$.

3.2.2 Single Camera Inside-Out Tracking

For estimating the exact position of the carrier of the HMD respective the targets tracked by the ART tracking system, a single camera inside-out tracking system is utilized at CAM-

PAR. This tracking system, denoted as RAMP (Real Time Augmentation for Medical Procedures), was developed at Siemens Corporate Research¹ [72]. For tracking it uses an infrared camera mounted on the HMD. It includes a fish-eye lens with an extended angle of vision to stabilize tracking and visualization. It only tracks markers belonging to the reference frame mentioned in Section 3.2.1. The number and design of the markers, which have a flat circular shape, attached to the reference frame contributes to the accuracy and robustness to the system. The reference frame has to be placed within the field of vision of the infrared camera. The infrared camera is mounted on the HMD above the color cameras as shown in Figure 3.2.

When tracking the reference frame, a coordinate system is calculated, which is overlaid with the coordinate system the A.R.T. tracking system computed for the frame (see Section 3.2.3 for further explanation). After measuring both coordinate systems, the exact position of the carrier of the HMD respective the tracked targets, and thus the coordinates of the virtual objects respective the recorded video images, can be determined.

By using RAMP, the AR system is able to exactly sense movements of the wearer of the HMD and thus to precisely adjust the coordinates of the virtual objects. RAMP allows for a high rotational precision [38], which is necessary for tracking the stereoscopic video see-through HMD.

3.2.3 Merging The Tracking Data

This section explains the merging of the tracking data computed by the single camera inside-out and the optical outside-in tracking system, which enables to determine the position of a tracking target respective the HMD and thus the visualization of virtual objects at their proper. Figure 3.4 illustrates this process. The inside-out tracking system calculates the transformation matrix ${}^{arc}H_{HMD}$ from the reference frame to the infrared camera mounted on the HMD, whereas the outside-in tracking system calculates a transformation matrix ${}^{arc}H_{IR}$ from the reference frame to the origin of the coordinate system of the outside-in-tracking system. All other tracking targets, respectively the attached fiducial markers, are only tracked by the outside-in tracking system, which additionally calculates the transformation matrix ${}^{ft}H_{IR}$ from the fiducial markers of a tracking target to the origin of the outside-in tracking coordinate system. The fiducial markers are registered to a (volumetric) data set containing the virtual information of the target (e.g. data from a CT-scan), whereas the transformation matrix ${}^{CT}H_{ft}$ from the data set to the fiducial markers is calculated. The following equation calculates the transformation ${}^{ft}H_{arc}$ from a (volumetric) data set to the HMD and thus to the cameras:

$${}^{CT}H_{arc} = {}^{CT}H_{ft} \cdot {}^{ft}H_{IR} \cdot ({}^{arc}H_{IR})^{-1} \cdot {}^{arc}H_{HMD}$$

During rendering this transformation has to be applied to the virtual anatomy (contained in a volumetric data set), in order to display it at the correct position in view of the observer.

¹Research and development department of Siemens AG, scr.siemens.com

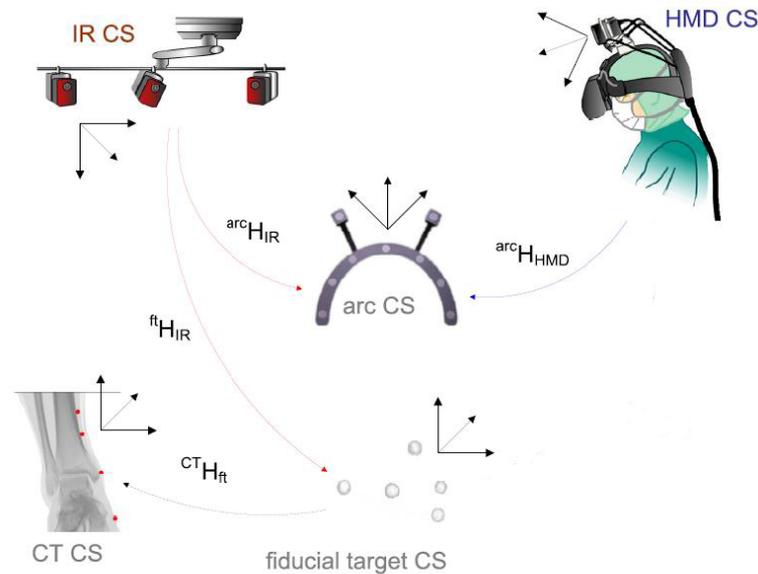


Figure 3.4: Transformations between the A.R.T. and the RAMP coordinate system.

3.3 Software Components

This section discusses the *CAMPAR software framework*. It was created by students and members of CAMPAR and is extended continuously. It is adapted to the demands of medical AR, and medical visualization in general, and contains methods for the processing of medical images, segmentation, registration, visualization and computer vision, whereas some of the methods are provided by a library denoted as *CAMPlib*. The main design requirements the CAMPAR software framework and *CAMPlib* are efficiency, reliability, reusability and flexibility. The software framework provides capability for different visualization techniques such as GPU²-based volume rendering or rendering of polygonal surface models.

The CAMPAR software framework and *CAMPlib* are programmed in C++. For visualization of virtual anatomy and augmentation of instruments, the open graphics library (OpenGL) is used.

²graphics processing unit

4 Focus and Context Visualization: Introduction and History

This chapter provides an introduction to focus and context visualization. Volumetric imagery often includes a huge amount of information and complex structures. A complete three-dimensional visualization of the data would overwhelm the viewer, who is usually only interested in a small fraction of the data, which is in general occluded by exterior structures. Focus and context visualization intends to display this part - usually denoted as *focus* or region of interest - preferably uncovered, by discarding regions that conceal this focussed fraction. In medical AR the region of interest usually is the operation site for a surgery. Assumed that the surgeon wants to operate at a specific vertebra of the spinal column, additional unimportant parts like tissue and skin could hinder his view onto the vertebra. Thus the focussed region, in this case the examined vertebra, has to be extracted from the imaging data.

However, especially in medical AR, the observer is also interested in knowing the correct position of the extracted focus respective the surrounding anatomy. Consequently, parts of the surrounding anatomy have to be left over and displayed additionally in order to provide important *context* information, which serves for a correct perception of layout and distances of the focussed objects in the AR scene. The remaining context can even be useful for identifying observed objects. In medical AR one context layer is displayed anyway: the camera image of the real body. The image has to be integrated in such a way that it can serve as context information for the focussed virtual objects and that it avoids misleading depth perception (see Chapter 5 and 7 for further discussion).

Methods for focus and context visualizations are often based on the model of traditional hand-made technical or scientific illustrations. Figure 4.1 shows a technical illustration which provides a view inside a car. Semantically such illustrations are often composed of a focus region - for instance the motor and the inside of the car in Figure 4.1 (a) - and a context region - the carriage of the car. In the shown figures, the context region is faded out using transparency to reveal the inner focus region. At the same time, characteristic parts of the context are visualized to accommodate better understanding of spatial relationships between structures in the data. Methods for designing and creating focus and context visualizations of volumetric data, which are partially adapted from hand-made illustration techniques, are presented in Chapter 6 in detail. In the following sections the terms of focus and context (see Section 4.1), including criteria which have to be fulfilled when designing focus and context visualizations, and scientific focus and context illustrations and their history (see Section 4.2) are discussed.

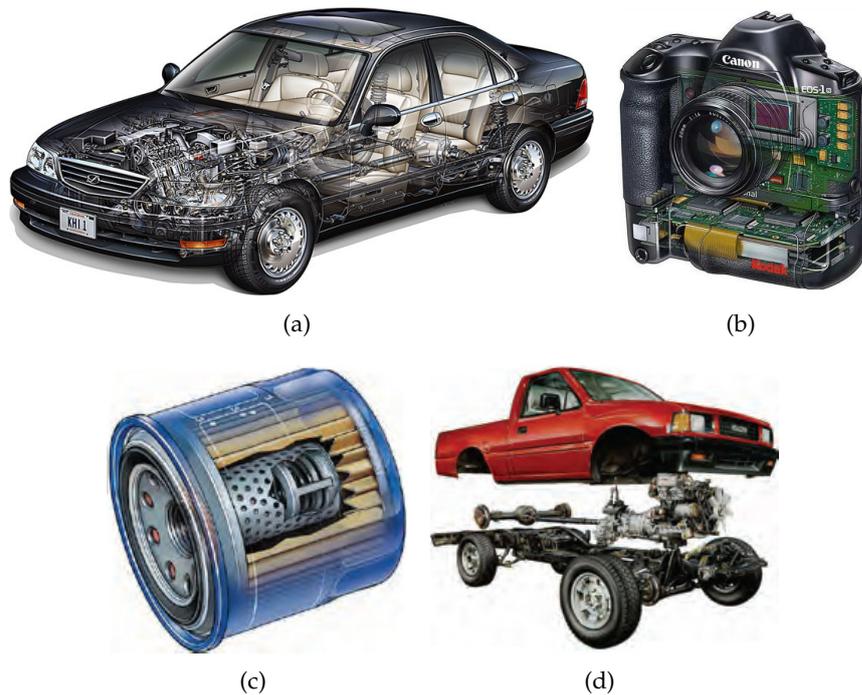


Figure 4.1: Different types of hand crafted expressive illustrations: (a) and (b) ghosted view, (c) cutaway and ghosted view, (d) exploded view. Technical illustrations are courtesy of Kevin Hulsey Illustration, Inc [39].

4.1 Focus and Context

The focus of visualized data marks the region in which the viewer is interested in examining and thus becomes more important than other regions. For visualizing the focussed region the following criteria have to be considered:

- The focus region has to be uncovered and visible to the viewer.
- Objects within the focus region have to be perceived at their right location (see Chapter 5).
- Objects within the focus region have to be correctly identified.
- Objects within the focus region have to be emphasized and displayed in detail.

The first criterion can be achieved by hiding objects that are located in front of the focus. For satisfying the second criterion correct perception of spatial depth and layout has to be assured. Therefore certain depth and positional cues have to be provided, which can be

achieved by remaining certain context information. Particularly in AR, where virtual objects and a camera image have to be displayed simultaneously, misleading depth perception forms a fundamental problem. Chapter 5 discusses this topic in detail. Sometimes an object can be hardly identified if it is displayed without its context, e.g. certain viscera or a vertebra. If parts of the skin, bones or close-by located organs are displayed additionally, it could be much easier for the observer to identify the focussed object. Thus for achieving the third criterion, it might also be necessary to remain additional fragments which provide context information. The fourth criterion can generally be achieved by using highly resolved data sets for focus objects and by applying suitable shading methods for rendering them.

In order to fulfill the second and third criteria it is necessary to display additional context information. In general context is important :

- to provide cues for the perception of layout and distances of entities in the focus region,
- to identify visualized entities by means of the context,
- and to understand complex anatomic structures.

When providing context information it has to be always taken into account that the view onto the focus should not be disturbed by the context fragments.

4.2 Focus and Context in Scientific Illustrations

Computer-generated focus and context visualization techniques are often inspired by traditional hand-made technical or scientific illustrations. An illustration is a picture with a communicative intent. It is an visualization such as drawing, painting, photograph or other work of art that stresses subject more than form. It conveys complex structures or procedures in an easily understandable way. Illustration techniques exploit the perception of the human visual system and provide effective visual abstractions to make the visualization clearly understandable. They provide a selective view on important details while extraneous details are omitted.

Illustrations can communicate the shape and structure of complex objects and provide an improved feeling for depth, occlusion, and spatial relationships. In general, illustration techniques can be summarized in visual emphasis and abstraction. Both have been used for expressive presentation from prehistoric paintings to nowadays scientific and medical illustrations. In general, the abstraction techniques can be subdivided in cut-away, ghosting and explosion techniques (see Figure 4.1), which can be found in most books on technical or medical illustrations [37]. A detailed discussion of these techniques, in conjunction with the generation of focus and context visualizations of volumetric data, follows in Section 6.4.1. Illustrators work with abstraction in order to prevent visual overload and to focus on the essential parts. Abstraction is enabled by the use of distinct stylistic choices

and is fundamental for creating expressive illustrations [85]. Goals of abstraction techniques are [85]:

- Communication of shape and structure
- Emphasizing or de-emphasizing
- Prevention of visual overload
- Suggesting of artificiality
- Ensuring of visibility of important structures
- Providing spatial context

For creating illustrations a basic principle counts: illustrations should be as detailed as necessary, but as simple as possible [85]. The Gestalt Theory says that simplest things will be perceived first ("Rule of Simplicity" [85]). Simplifying and leaving away makes forms clearer, whereas too much details impede the direct perception of the essential form.

This section deals with the history of illustrations (see Section 4.2.1) and discusses the modern need of automated illustration techniques (see Section 4.2.2).

4.2.1 History of Hand-Made Scientific Illustrations

Centuries ago artists and scientists already discovered the focus and context paradigm to create medical illustrations. The first illustrated printed medical book "Fasciculus medicinae" (see Figure 4.2 (a)) was published by Johannes de Ketham in Venice in 1491 and the first printed illustrated anatomy book "De Humani Corporis Fabrica" (see Figure 4.2 (c)) was published by Vesalius in 1543. Both books contain focus and context based anatomical illustrations. About 1510 Leonardo da Vinci created anatomical sketches using the focus and context paradigm to convey relative positions of inner organs and muscles in the human body. In Figure 4.2 (b), which shows two illustrations of Leonardo da Vinci, inner organs and blood vessels are illustrated within the context of parts of the surrounding skin. In 1747, the German physician Bernhard Siegfried Albinus collected focus and context based anatomical sketches for his monumental "Tabulae sceleti et musculorum corporis humani" (see Figure 4.2 (d)), which was first published in Leiden, largely at his own expense. In 1837, the German Carl Haag (1820 - 1915) created the illustration "Anatomical Hands" (see Figure 4.2 (e)).

Also nowadays artists and illustrators make use of the focus and context paradigm. The medical scientist Alexander Tsiaras [81] creates illustrations of the inside of human bodies (see Figure 4.3). Therefore anatomical structures are presented within the context of the patient's skin or nearby other anatomical structures. Like mentioned before, technical illustrations like Figure 4.1 are also often composed of a focus and a context region.

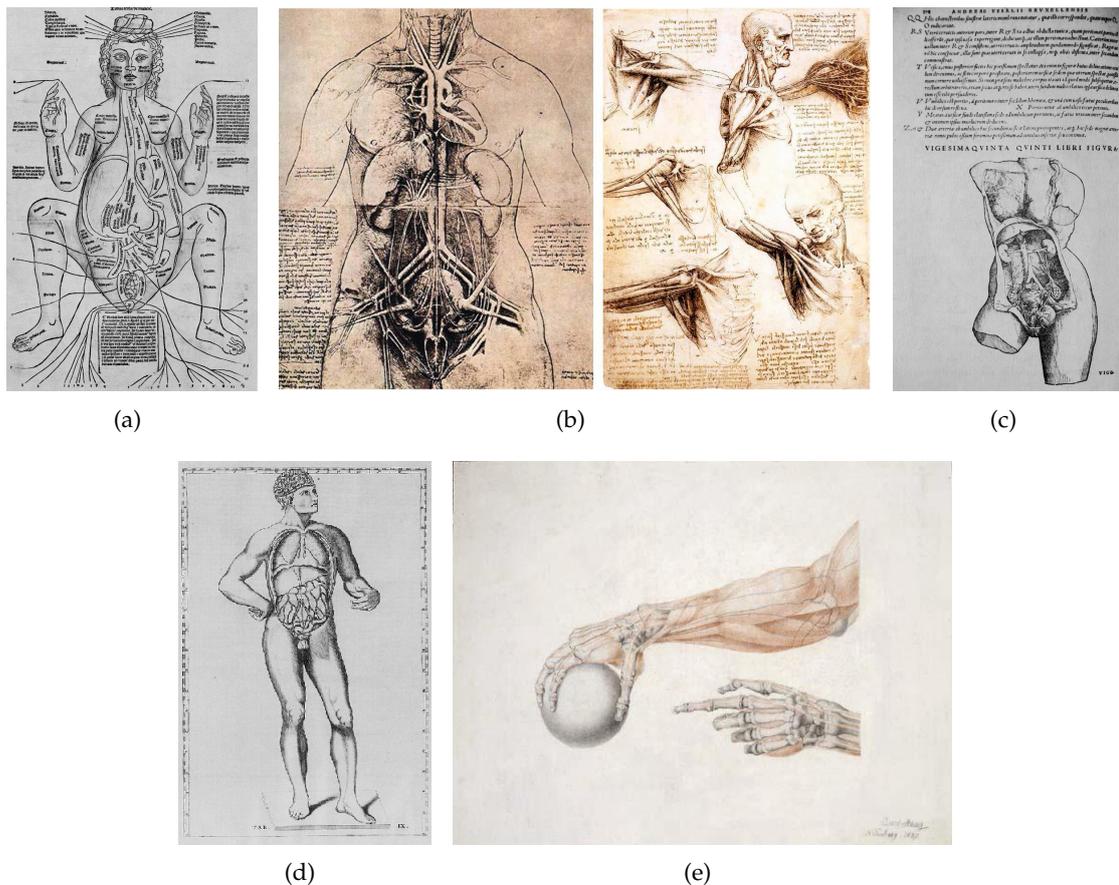


Figure 4.2: Early anatomical illustrations: (a) Johannes de Ketham in 1491, (b) Leonardo da Vinci about 1510, (c) Vesalius in 1543, (d) Bernhard Siegfried Albinus in 1747, (e) Carl Haag (1820 - 1915): Anatomical Hands (Germany, 1837)

4.2.2 Modern Illustration with Computer Graphics

Nowadays, modern computer graphics take advantage of illustration techniques for generating focus and context visualizations. Modern imaging, data acquisition, and recording techniques like X-ray, CT, MRI, ultrasound, electron microscope or photography and film create a huge amount of scientific data. New imaging modalities provide spatial (and temporal) reconstruction of organic structures and multidimensional information (e.g. soft tissue, metabolism, brain activities). To effectively convey the most important visual information out of large and multi-dimensional volumetric data sets, there is a significant need for visual abstraction. Visualization approaches face the challenge of "making visible the invisible", for instance structures on atomic level or anatomic structures. Therefore traditional illustration techniques can be combined with modern media and imaging tech-

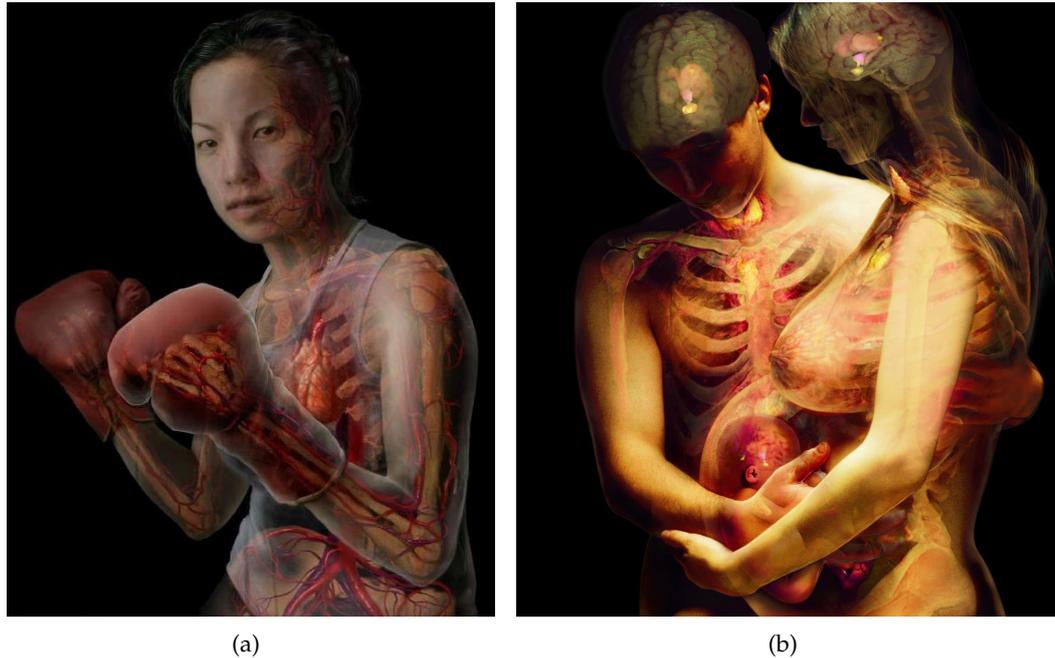


Figure 4.3: Anatomical illustrations by Alexander Tsiaras

niques.

Many of the expressive techniques used in hand-made illustrations and art are adopted in computer graphics. Illustrative volume rendering has become a great topic of research in the past years. It refers to approaches dealing with feature emphasis or feature suppression of focus and context regions in the volumetric data, using different transparency levels or specific shading techniques. These rendering methods are often subsumed under the term non-photorealistic volume rendering. An important aspect as compared to traditional medical or scientific illustrations is the interactivity and real-time manipulation of the acquired data. Chapter 6 describes methods for designing automated focus and context visualizations by exploiting illustrative techniques.

5 Perception of Distances, Layout and Shape of Objects

In medical AR and especially in intra-operative AR a correct perception of the layout (arrangement) and distances and thus positions and depth of focussed objects is extremely important. In intra-operative AR a misleading or inexact perception of positions of objects includes the risk that the surgeon operates at the wrong place and thus hurts regions of the body, which normally would not have been involved in the surgery. In the extrem case essential and vitally important organs could be violated.

Perception of layout and distances of objects in the AR scene primarily means perception of spatial depth. For the perception of spatial depth, specific depth cues are exploited by the visual system, which can be denoted as specific positional cues, which help to perceive the third dimension (the depth) of the environment for the determination of layout and distances. Depth perception is thus the visual ability to perceive the world in three dimensions. Detecting if an object is positioned in front or behind an other object is a much more complicated task for the visual system, than determining whether an object is placed above, below, to the left or to the right of an other object. Depth cues and their utilization for medical AR and focus and context visualization are discussed in Section 5.2.

Like mentioned in Chapter 3.2, tracking systems enable the projection of virtual objects to the correct position and their scaling to the right size respective the real scene. Despite that fact, depth perception can still be perturbed in medical AR visualizations, which constitutes one of the major problems of medical AR. Examples for such misleading depth perception are illustrated in Figures 5.1, where a virtual spinal column (Figure 5.1 (a)) and a skull (Figure 5.1 (a)) are superimposed on the patient's body. Since the virtual objects occlude parts of the body and are nowhere occluded by the skin, they appear as being located in front of the patient. The reason for that lies in a misinterpretation in the perception process of the visual system (see Section 5.1) and more precisely the depth cue occlusion (see Section 5.2.2), which is discussed later in this chapter.

When designing a focus and context visualization framework for medical AR, as the one described in Chapter 7, it has to be assured that no misleading cues are provided, which disturb the perception of the correct layout and distances of objects lying in the focus of the surgeon. All possible depth cues have to be considered and exploited in order to enable a preferable exact perception of the focus objects. Therefore as many depth cues as possible can be made available by an appropriate application of context information (see Chapter 4), which includes for instance a correct transparency modulation of context objects. The use of transparency for an improved perception of layout is discussed in

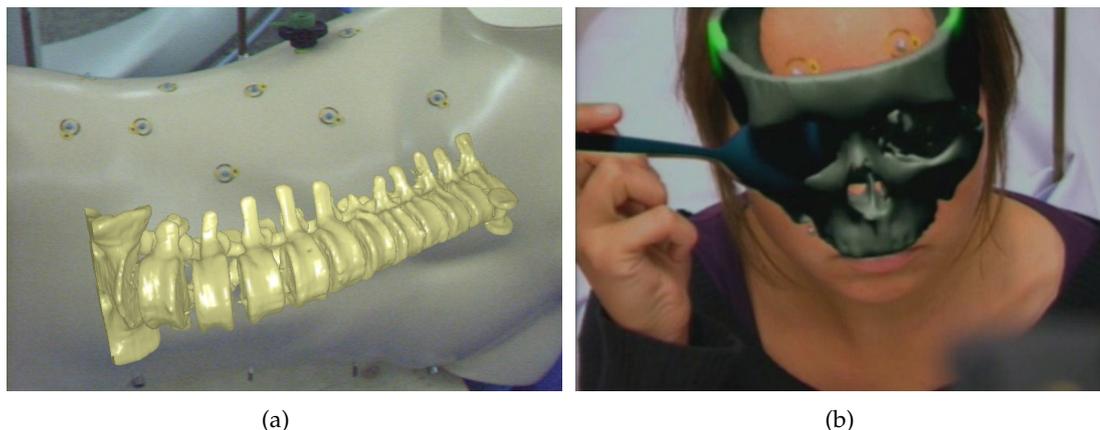


Figure 5.1: Misleading depth perception of virtual anatomy: the spinal column (a) and the skull (b) are superimposed on the patient's body and seem to be located in front of the patient.

Section 5.3 in detail. The main task in medical AR is to utilize the camera image (beside virtual objects) as context object for the virtual focus objects and thus for their correct spatial perception.

For a clear cognition and identification of objects, a good perception of the shape particularly of focus objects, but as well of context objects, is an additional task an framework of focus and context visualization has to fulfill. The perception of shape and orientation is discussed in Section 5.4. The chapter starts with a short introduction to the process of human visual perception.

5.1 Introduction to Visual Perception

Human visual perception is a sophisticated process, which has been examined by psychologists for decades. The perception process starts with a projection of parts of the environment onto the retina, which is located at the back of the eyeball (see Figure 5.2 (a)). Light that is reflected by the objects of the environment thereby meets the eye, passes the iris and forms an image on the retina (see Figure 5.2 (b)). The most important difference between the retinal image and the environment is that the environment is three-dimensional, whereas the retinal image is two-dimensional. Physical objects of the environment differ much from their retinal images. Thus, psychologists denote the physical object of the environment as the *distal stimulus* (distant to the observer) and its optical image on the retina as the *proximal stimulus* (proximal to the observer) [96]. The main task of visual perception is to reconstruct the distal stimulus from the proximal stimulus. Perception encompasses all processes, which structure and interpret the information included in the sensory image, as

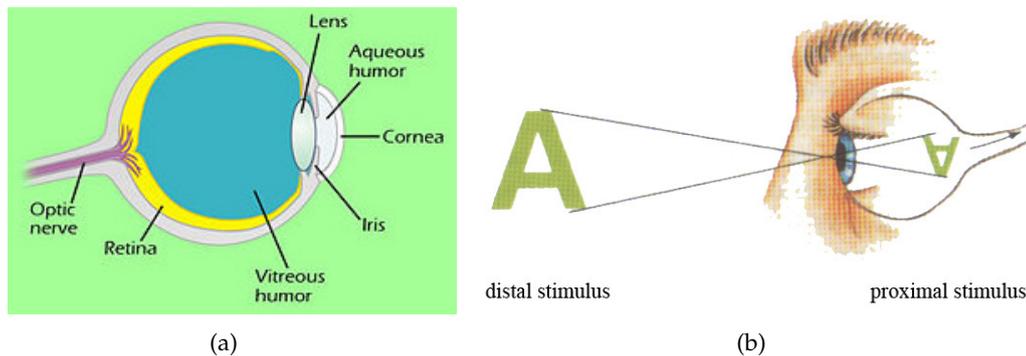


Figure 5.2: Eye [22] (a) and projection of an object (distal stimulus) on the retina (proximal stimulus, b) [96].

if this information was directly produced by properties of objects or events of the external world [96]. Thus, the two-dimensional retinal image has to be processed in such a way that it reproduces the three-dimensional real world. Perceptual processes extract meaning from the continuously changing and chaotic sensory input and organize it to stable, ordered percepts. A *percept* is the result of the perception process. It is neither a physical object, nor a retinal image, but the mental product of the perception activity.

The process of perception can be divided into three stages [96]:

1. Sensation - The projection of parts of the environment on the retinal receptors in the observer's eyes. Physical energy is thereby converted to neural coded information that is identifiable for the brain. The retinal cells yield fundamental facts of the field of vision by highlighting edges and contrasts, whereas they react only little to changeless and constant stimulations.

2. Organization - This stage deals with questions like "*How does the viewed object look like?*" and "*Where is the object positioned?*"

The human visual system generates an internal representation of an object and out of it a percept of the distal stimulus. The percept is a first description of the exterior environment and includes assumptions about the probable size, form, movement, orientation and distance of an object. It is created by a synthesis of information about simple sensory attributes like hues, edges of viewed objects, lines and homogeneous areas. Therefore experience values and actual sensory information are merged.

Since distances are already allocated in this stage between objects of a certain scene, this stage is crucial for depth perception. If an object A is partially occluded by object B, object A is perceived behind object B. In medical AR that means if the virtual anatomy is simply superimposed on the "real" camera image, two homogeneous areas are perceived in this stage. These areas represent the real object, i.e. the skin of the patient and the virtual anatomy, e.g. the spinal column. In this stage these

areas are not identified as body or spinal column, but the homogeneous area, which represents the spinal column, is perceived in front of the homogenous area, which represents the body, since the first area partially occludes the second one (see Figure 5.1). Consequently, simply superimposing the virtual objects onto the camera image causes a failure in this perception stage.

3. Identification and recognition - This stage deals with the questions: *"Which object is it?"* and *"Which function has the object?"*

This stage assigns a certain meaning and signification to the percepts. Round objects "become" footballs, coins, clocks or oranges; humans are identified as male or female. When observing a medical AR scene, virtual objects "become" a spinal column, the heart or the lung and the real body (or patient) is also identified. Therefore higher cognitive processes are necessary, which include theories, memories, value system, dogmas and attitudes with regard to the object.

Regarding perception of a medical AR scene, where the virtual objects are simply superimposed on the camera image, the second stage perceives the virtual objects in front of the real objects. The last stage identifies the virtual objects as objects belonging to the inside of the body. Thus the visual systems notices that the constellation of the objects was assessed wrongly by the second stage. The last two stages of the perception process provide contrary information about depth, which results in an undesired misleading depth perception. A misleading depth cue can be hardly compensated by the last stage. The following section presents depth cues which are utilized in the second stage of perception. Within that context solutions for the described misleading depth perception are also discussed (see Section 5.2.2).

5.2 Visual Cues for Perceiving Layout and Distances

Various visual cues provide information about layout, respectively depth order, and distances of objects. As seen above, information about the spatial depth of objects is already gained in the "organisation" stage of the visual perception process. This section analysis different information for spatial depth and their combination. Single projection points of objects of the environment on the retina do not include information about spatial depth of these objects. As an example, Figure 5.3 shows the sensation of a tree and a house and thereby the projection of a point localized at the tree and a point localized on the house onto the retina. The projection positions on the retina give no information about absolute and relative distances of these two points, they offer no immediate distance information. Instead of observing these single points, image processing in the brain involves the whole image on the retina for this analysis. Moreover it considers that the two eyes record images out of slightly different positions and that during eye fixation (in close-up range) the eyes have to adjust to the right convergency angle. By doing that, a whole amount of

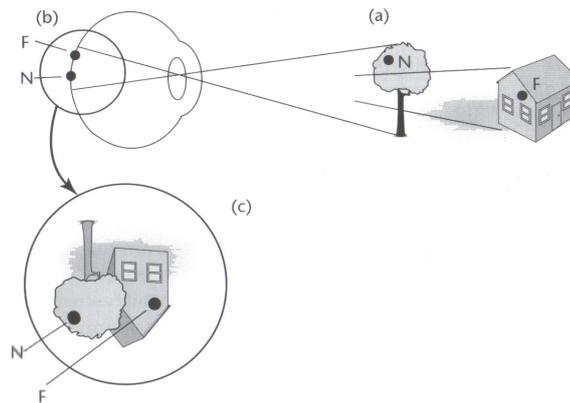


Figure 5.3: The house is further away than the tree (a). The projection of the points F and N - that belong to the house and the tree - onto the retina, do not contain information about the different distance of the two objects (b and c). [28]

information for spatial depth is available. From this information, the visual system can reconstruct the spatial structure of the observed environment. The cues which provide this depth information can be subdivided into the following groups [28]:

- **Oculomotor depth cues:** spatial information gained from feedback of eye muscles and lense thickness
- **Monocular depth cues:** spatial information gained from the structure of an unmoved image (e.g. a photo)
- **Motion induced depth cues:** spatial information gained from motion of the observer or the observed objects
- **Stereoscopic depth cues:** spatial information gained from observing the scene with two eyes

Cutting et al. [17] defined three areas, where depth cues can be effective: personal space (the zone immediately surrounding the observer's head, up to 2m), action space (up to 30m) and vista space (beyond 30m). The workspace of a surgeon is entirely located in the personal space, which is thus important for applications in medical AR. Depth cues which are effective in this area are discussed in more detail.

5.2.1 Oculomotor Depth Cues

Oculomotor depth cues result from the ability of the visual system to evaluate the convergence angle and the accommodation of the eyes. *Convergency*, respectively *divergency*, are signals generated in the eyes, when the eyes move inwards or outwards in order to detect

near or far located objects. *Accommodation* is the change of the shape of the lense when fixating an object. In order to set a near object sharp the lense becomes thicker. To get a sharp picture of a distant object it becomes flat. Out of changes of the convergency and the accommodation in conjunction with the associated control and feedback signals, the visual system is able to gain information about spatial depth. The state of the eyes and the shape of the lense correspond systematically with the distance of the observed object.

These depth cues may be most effective in personal space, but they can not be exploited in AR applications using a video see-through HMD described in Section 3.1. The HMD uses cameras with a fixed convergency angle and a lense which is adjusted to a fixed distance. In those AR applications eye movement and lense thickness of the eye do not influence the observed image and thus are not able to provide important depth cues. This is definitely a lack of such AR systems, because "these sources could be extremely effective in measuring distance, yielding metric information within the near distance" [17]. In medical AR systems depth perception is thus limited to the other depth cues described in the following sections.

5.2.2 Monocular Depth Cues

Monocular depth cues describe depth information that is still effective when using one eye for observing the environment. They result from regularities in projections, which correspond to the distance of objects. These cues are spatial information, which can be gained out of the structure of a motionless picture (e.g. a photo). The following depth cues are monocular.

Occlusion/Interposition

Occlusion occurs when an object A (e.g. the blue disk in Figure 5.4 (a) or the left white foot in Figure 5.4 (b)) partially hides an object B (e.g. the red disk in Figure 5.4 (a) or the right white foot in Figure 5.4 (b)). Thus object A is observed as lying in front of object B. Thereby one has to notice that occlusion does not provide any information about absolute and relative distances of an object. "It offers information about depth order, but not about the amount of depth" [17]. That means that if only the depth cue of occlusion is provided to the observer of object A and B, he or she does not know how far the objects are located in front the observer and moreover has no information about the relative distance from object A to B. Nevertheless occlusion is the most important cue for perceiving a depth order of objects and its "threshold for ordinal depth judgements generally exceeds all other sources of information." [17]. Occlusion is effective at any distance [17] and thus also in the workspace of a surgeon.

This depth cue is responsible for the misleading depth perception mentioned at the beginning of this chapter and in Section 5.1. This misleading depth perception occurs if a virtual object is simply superimposed on the camera image of the patient's body. It can only be solved if the "real" body occludes parts of the virtual object and furthermore the

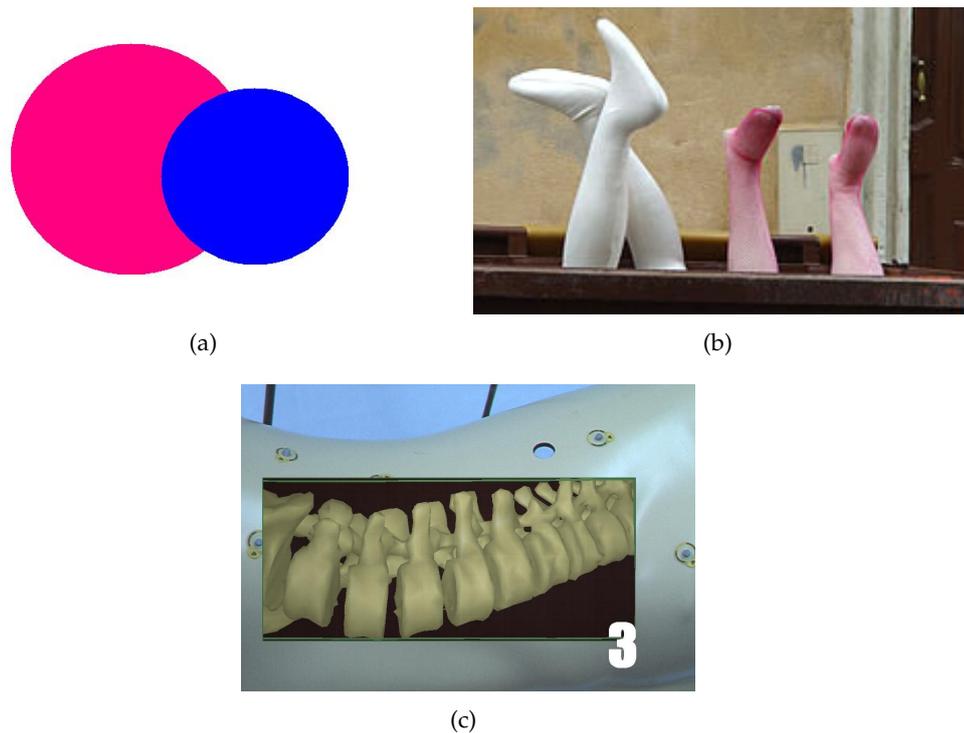


Figure 5.4: Depth information by occlusion

virtual object does not occlude parts of the body. This can be realised by cutting a window into the image of the body [5]. The virtual object then is only displayed at screenspace belonging to the window. This effect is shown in Figure 5.4 (c). In contrast to Figure 5.1 the depth order is perceived correctly.

Relative Height in Visual Field

Objects, which are positioned at higher position in the visual field as others are usually perceived as further away. If the horizon line can be seen by the observer, this holds only to objects which are located below the horizon line. Otherwise objects (e.g. clouds) which are positioned above the horizon line, seem to be further away, if they are located on low position. Thus, the nearer an object is positioned to the horizon line, the further its distance is perceived. This depth cue can be able to give information about absolute distances, but it is not very effective in the personal space and thus less important for realising medical AR visualizations.

Relative Size in Visual Field

A further depth cue is the relative size of similar objects. "Relative size is the measure of the projected retinal size of objects or textures which are physically similar in size but at different distances" [17]. If two objects have nearly the same size, the nearer one covers a bigger part of the visual field.

Relative size can yield scaled information, instead of only offering a depth order (e.g. as occlusion). If the observer additionally knows the exact size of the objects, this depth cue can furthermore offer information about relative distances between them. In this case the depth cue of familiar size is also taken into account, which is described in the following section. "Like occlusion, relative size can generally be trusted throughout the visible range of distances from say 0.5 m to 5000 m". Thus, it is effective in the personal space and has to be considered for creating medical AR visualizations, even though there are only few objects in the human body that look similar and have the same size. Figure 5.5 shows four blue circles in different sizes, whereas small circles are perceived as more distant than big circles. The fact that the big circle in Figure 5.4 (a) is perceived as further away than the small circle demonstrates that the depth information gained from occlusion is stronger than the depth cue of relative size.

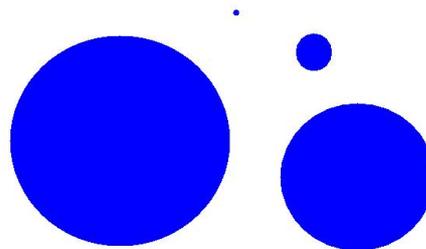


Figure 5.5: Relative size of objects

Familiar Size

An observer of an object is able to estimate the distance of an object, if he or she knows the size of the object in advance. The depth cue provided by this knowledge is called familiar size. An experiment by William Epstein [21] showed that knowledge about the size of objects can influence the perception of depth of these objects. For his experiment, Epstein used equally sized photos of a 10-cent coin, a quarter and a 50-cent coin, which were placed in equal distance to the observers. The observers judged the distance to the 10-cent coin as least and the distance to the 50-cent coin as largest. Thus, the judgements of the observers were influenced by their knowledge about the real sizes of the coins. This depth cue is most effective if other depth cues (apart from relative size) are widely absent

[28].

Linear Perspective

When looking along parallel lines (e.g. train rails), these lines converge in distance until they unify in a vanishing point (see for instance Figure 5.6). The perceived distance of train rails, roadsides or the rows of a cornfield becomes so small at great distances that they finally merge to a point. This legality of depth perception is called linear perspective. For hundreds of years it was used in painting for generating the effect of depth. In the fifteen's century a technique called Alberti's window, developed by Leon Battista Alberti, became a famous technique for perspective painting. This depth cue can be exploited in focus and context visualization for medical AR (see Section 7.4.5).



Figure 5.6: Linear Perspective: parallel lines converge with increasing distance

Texture Gradient

When observing a regularly structured surface, which has a spatial extension into the view direction, linear perspective and relative size act together [45]: parallel structures converge, whereas simultaneously the horizontal equally spaced elements of the surface (the texture) appear more densely compressed with increasing distance. An example would be a corn field that gets a more and more densely structure with increasing distance. Gibson [27] described this source for spatial depth as texture gradient. The texture gradient allows to estimate the expansion of surfaces and to distinguish horizontal and vertical surfaces. Unlike textures of horizontal surfaces, the texture of a vertical surface does not show these

perceptual effects. Texture gradients normally occur on the plane the observer is standing on. Studies showed that depth perception degrades when "taking away" the floor, and the observer has a worse estimation of distance [76]. Texture gradients can also be useful for perceiving the shape of an object, which is discussed in Section 5.4.

Aerial Perspective

The depth cue of aerial perspective assumes that the observer is looking through air which holds a certain amount of moisture or pollutants. The further away an object is located, the more air and particles lie in the line of view. Consequently, distant objects become more blue and appear less sharp than near objects (like the mountains in Figure 5.7). In computer graphics these effects are called participating media. The effectiveness of aerial perspective "increases with the logarithm of distance" [17] and provides only depth information for objects which are located several meters away from the observer. In personal space (the area of interest in medical AR) this depth cue is negligible.

Nevertheless, the concept of aerial perspective can also be applied to the perception of transparency [26, 62, 17]. Instead of looking through air, the observer hereby looks through two or more transparent objects which become the medium. Unlike aerial perspective, where a graded continuous function is assumed, transparency generates a discrete-valued step function. In this connection aerial perspective can be helpful for the purpose of generating a focus and context based visualization system for medical AR. In such a system, objects which are located in front of the focus can be rendered transparently, such that the observer gets certain information about spatial depth of objects located in the focus. The use of transparency for providing positional cues is discussed in Section 5.3.



Figure 5.7: Aerial Perspective: mountains in the distance appear more blue.

Chromostereopsis

Chromostereopsis is another phenomenon of visual perception: different wavelengths of light (and thus different colors) are projected onto slightly different positions on the retina [87]. The stimulation of the different areas of the retinal receptors is aligned with depth

information by the visual system. Thus objects with cool colors (with shorter wavelength, like blue) appear further away than objects with warm colors (longer wavelength, like red or yellow). Figure 5.8 illustrates this effect.

For generating a medical AR visualization system, the coloring of objects is thus very important. By coloring near objects (e.g. the skin) with a warm color and distant objects (e.g. bones) with a cooler color, the distance of the objects can be additionally emphasized.

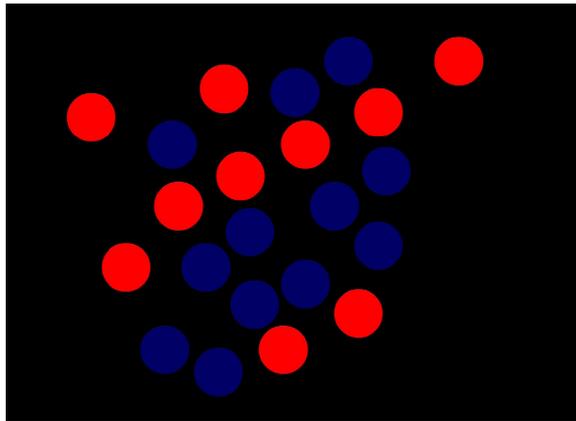


Figure 5.8: Chromostereopsis: warm tones (red) appear nearer to the observer than cool tones (blue).

Light and Shadows

Cues for layout and distances can also be obtained from light distributions and shadow casts. Illumination of an object usually causes the cast of a shadow from this object onto other objects. If an object is shadowed by another object, the observer obtains information about the constellation of these two objects. He or she can estimate a spatial relation between the objects and the light source. Figure 5.9 (a) shows an example for the obtainment of positional cues from shadow cast. Due to the shadow which is casted by the boy, the observer can estimate the distance between the boy and the wall in the background. Furthermore the observer can figure out the position of the light source. If the observer has the additional ability to move the light source, the observer gets further information about constellation of the illuminated objects.

Kersten et al. [47] arranged a psychological experiment presenting the so-called ball-in-box scene. The authors claim, that "the results support the hypothesis that the human visual system incorporates a stationary light source constraint in the perceptual processing of spatial layout of scenes" and "the information provided by the motion of an object's shadow overrides other strong sources of information and perceptual biases, such as the assumption of constant object size and a general viewpoint". Figure 5.10 illustrates the

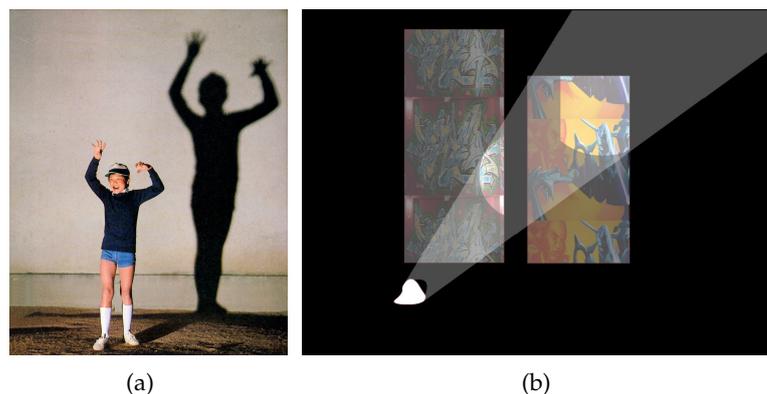


Figure 5.9: Light and Shadows: (a) Shadow cast of a boy onto a wall and (b) the use of spotlights for obtaining positional cues.

described importance of shadows for the perception of depth. Figure 5.10 (a) shows the ball-in-box scene, where a ball is positioned on a surface top right in the image, whose cast shadow is also positioned top right. Figure 5.10 (b) shows the same ball positioned at the same position in the image, whereas here the cast shadow is positioned bottom right. As a result the ball in the first Figure 5.10 (a) is perceived more distant than the ball in Figure 5.10 (b). In addition the relative distance between the ball and the surface is perceived as a lot smaller in the first figure. Figure 5.10 (b) shows three squares with different positions of the cast shadow. Here the position of the shadow is also crucial for the perception of the distance between the squares and the ground, and between the squares and the observer.

When using a spotlight instead of a light shining equally in every direction, the effect of the shadow cast can even be amplified. A spotlight is a very strongly bundled or bounded light and is usually used to direct the observer's attention to a certain location. But it can additionally provide information about the constellation of objects. By means of the size of the projected cones of light on certain objects, the observer can estimate the relative distance of the objects and the distance of the objects to the light source. Figure 5.9 (b) shows two walls, which are illuminated by a spotlight left to the observer. Due to the illumination of the two walls the observer is able to figure out that the left wall is positioned closer to him or her. In order to use a spotlight, the ambient light of the scene should not be too strong.

5.2.3 Motion Induced Depth Cues

All former described depth cues also work if the observer is not moving. Motion induced depth cues emerge combined with motion of the observer or the motion of objects in the environment. Motion "is even claimed to be the foundation of human vision" [17]. Never-

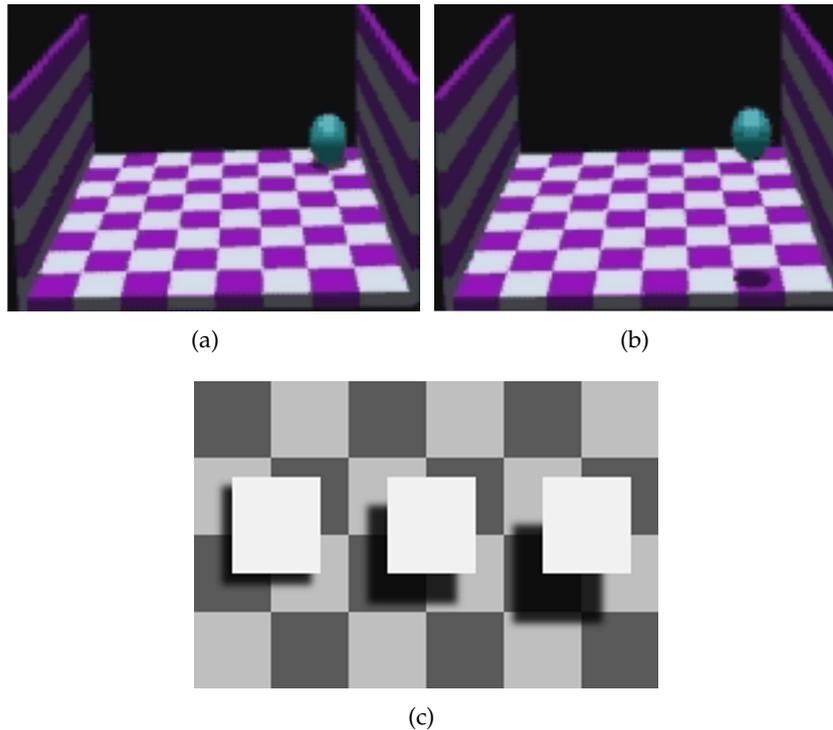


Figure 5.10: Perception of depth through different cast shadows [48]: distant perception of a ball (a), near perception of a ball (b) and perception of three squares with different cast shadows

theless, Cutting and Vishton think that its role in the perception of layout “may have been overplayed” [17]. They expect that it is relatively important, but not all encompassing.

Motion Parallax/Motion Perspective

In this section motion parallax and motion perspective are discussed. Motion parallax means the “relative movement of the projections several objects stationary caused by observer movement” [17]. It thus always appears when the observer is moving. If he or she for example travels by train or car and looks out of the window, near objects pass very fast whereas far objects at the horizon move only little. Thus motion parallax means the difference in velocity of those objects. It serves as criterion for perception of spatial depth based on relative velocities between the observer and viewed objects. Figure 5.11 shows the reason for the appearance of motion parallax [28]. The eye moves from position 1 to position 2. Thereby the projections of a near object A and a distant object B change their positions on the retina. Notice that during position change of the eye the projection of the near object A moves much more on the retina than the projection of the far object B.

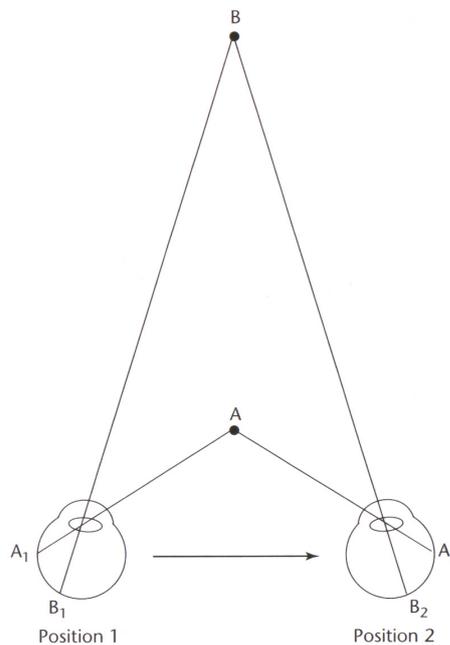


Figure 5.11: Motion parallax: changing of the retinal projection of objects A and B by moving of the eye from position 1 to position 2 [28].

Motion perspective means the motion of a whole field of objects whereas the observer stays at his or her position. It provides the same information for depth perception as motion parallax.

Motion provides a depth cue, which is relevant for the personal space and beyond. In a medical AR system as described in Chapter 3, the observer (the surgeon) is able to move around while carrying the HMD. The positions of virtual object are adjusted to the point of view of the observer by means of the tracking system. Thus, the depth cues presented by motion can always be utilized by a medical AR application.

Cover and Uncover Planes

If two planes are in different distances, motion of the observer which is non-perpendicular to these planes, has the effect that the planes seem to be moved towards each other. The far plane becomes covered by the near plane if the observer moves in one direction, and uncovered if the observer moves to the opposite direction. This depth cue is especially effective for the perception of spatial depth at edges [44]. Covering and uncovering of planes is a combination of the depth cues of motion parallax and occlusion.

5.2.4 Stereoscopic Depth Cues

All previously discussed depth cues can also be gained from observing a scene with only one eye. In this Section **binocular disparity** is explained. It originates in the participation of both eyes. The two eyes are positioned about six centimeter away from each other and thus have a slightly different view of the world (see Figure 5.12 (a)). When observing the environment two different images are projected to their retinas. These differences in the relative position of the retinal projections of the same objects are called binocular disparities. They are decoded by the visual system and transferred to stereo vision and **stereopsis** (see Figure 5.12 (b)). Stereopsis (respectively stereoscopic viewing) means the impression of spatial depth by using both eyes [28]. For the decoding process of binocular dispari-

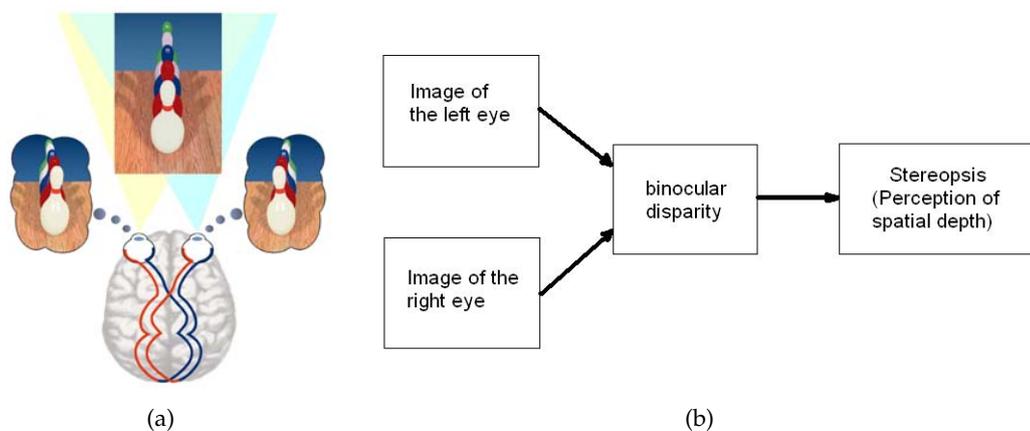


Figure 5.12: Binocular disparity is generated by slightly different views of the two eyes (a). Binocular disparity leads to stereopsis [28] (b)

ties the visual system has to find corresponding points on the two retinal images and a complicated fusion process has to be performed [28].

Disparities and thus stereopsis are the key source for three-dimensional perception of the environment. Only through stereopsis a three dimensional image of the perceived world can be reconstructed in the human brain. It "has often been taken to the strongest among all sources of information about layout" [17]. Binocular disparities offer an absolute metric for distance and are most important in personal space [17] and thus also very important for medical AR systems.

In medical AR systems stereopsis can be exploited using two cameras, which simulate the views of the eyes, and two displays attached to the HMD. Thus three-dimensional perception of layout and the important stereoscopic depth cues do not get lost in a medical AR system.

5.2.5 Summary of Depth Cues

Personal space is the space, which has to be considered for designing and generating medical AR systems and visualizations. Within this space, Cutting et al. identified occlusion, binocular disparity, motion parallax and relative size as the most effective depth cues [17], but also other depth cues, including the ones composed of other depth cues, can be very important for perception of the layout within the personal space. These cues are familiar size, linear perspective, texture gradients, aerial perspective (respectively transparency, see Section 5.2.2), chromostereopsis and light effects and shadows. All these depth cues were considered for the design of the focus and context visualization framework presented in Chapter 7. The correct perception of distances and layout is of capital importance for medical AR visualizations.

The most important cue for perceiving a depth order of objects is occlusion. It is responsible for the misleading depth perception, when superimposing virtual objects on the camera image. This misleading depth perception can be solved by cutting a window into the camera image, which separates the real and the virtual part (see Section 5.2.2). Further methods are presented in Chapter 7.

For visualization purposes only monocular depth cues have to be considered, or can be generated artificially. Motion induced and stereoscopic depth cues are provided implicitly by a medical AR system, which includes a tracking system and a HMD. Unfortunately oculomotor depth cues can not be provided by current AR systems yet.

5.3 The Usage of Transparency for the Perception of Layout and Distances

In focus and context visualization the usage of transparent context objects is able to improve the perception of layout and distances of focus objects. Objects hindering the view onto the focus have either to be displayed completely transparent or slightly semi-transparent in order to guarantee the visibility of the focus objects. By displaying those objects in a completely transparent manner, a clear view onto the focus would be enabled, but in this case those objects would not provide any positional cues for the focus objects and thus no context information. Displaying those objects in a semi-transparent way can solve this problem. Thus the relative position of focus objects respective surrounding objects can be estimated a lot easier. By the use of semi-transparent context objects the depth cue of aerial perspective can be enabled to a certain degree (see also Section 5.2.2). Thereby, instead of air, the transparent objects become the medium. Since the correct determination and perception of the focus objects is the most important task in medical AR, the use of transparency can be very helpful for creating a focus and context visualization for medical AR.

However, using semi-transparency for displaying context objects is a very complicated issue. It has the disadvantage that there would be always a certain disturbance of the

view onto the focus. A false application of transparency could rather lead to confusion of the perception of underlying information than to provision of positional cues for it. Furthermore, the focus object could even be perceived in front of the context objects and thus a false perception of depth would be created. Figure 5.13 (a) shows two transparent

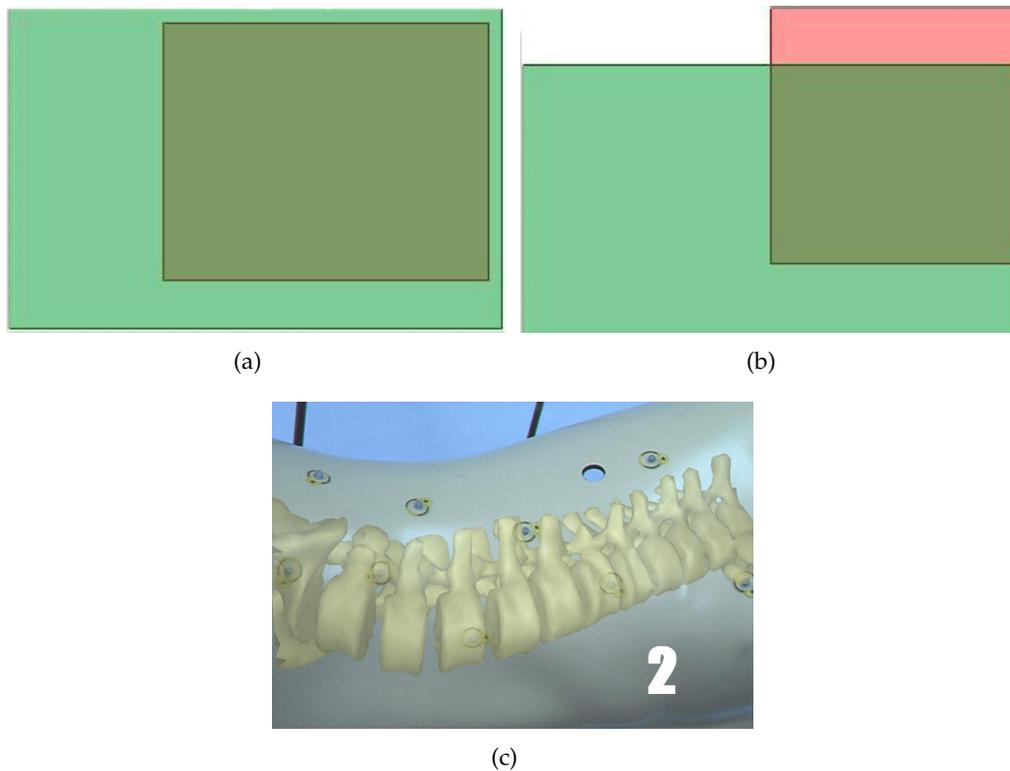


Figure 5.13: Perception of transparent objects

objects, where one object encloses the other one completely. Without knowing the color of the inner object, an observer cannot be assured which object is nearer to him or her. If there is only a partial occlusion of the two objects, the visual system would be able to organize the objects respective their depth, because at the intersection area the color of the near object overbalances the color of the distant object (see Figure 5.13 (b)). An evaluation study, performed by Sielhorst et al. [75], compared depth perception of different methods of rendering a virtual spinal column, which was superimposed on the camera image of a body phantom. The rendering of a transparent spinal column achieved better results for depth perception than rendering of an opaque spinal column - despite the fact that the focus (the spinal column) was rendered transparently instead of the context (the skin) and the fact that the focus was completely surrounded by the context. Figure 5.13 (c) shows the transparently rendered spinal column. In Section 7.3 a method is presented, which

adjusts the transparency of the context object - the camera image of the body - instead of the virtual objects, which leads to a more intuitive and thus better result.

In the example above and in the evaluation study the transparency of the objects is uniform at every position of the objects. However, the transparency of the object's surface can also be adjusted non-uniformly, for instance by means of the curvature of surface positions or by distance to contour lines. This can solve problems mentioned above. Different methods for adjusting the transparency of context objects are discussed in Section 6.4.2 in detail.

5.4 Perception of Shape

Besides the perception of layout and distances, correct perception of shape particularly of focus objects, but as well of context objects, is also important when designing a focus and context visualization framework. Thereby the shape of objects have to be perceived as clear as possible.

Extraction of contours, which are comprehended by the visual system as the borders between surfaces of different brightness, is the base process of visual perception [45]. During the "sensation" stage of visual perception (see Section 5.1) contours and contrasts are detected and emphasized by the retinal cells. In the organisation stage, closed contours, which completely surround a surface in the field of view, are perceived as forms. Perception of shape thus first means perception of contours [45]. Figure 5.14 (a) illustrates the contours of a David bust and demonstrates that contours include major information for the perception of shape.

Thus surfaces of virtual objects have to be shaded in such a way that contours can be extracted efficiently by the visual system. Therefore the brightness or contrast of a certain point on an object surface can be adjusted by the orientation of that point towards the observer or a specific light source. By doing that, different orientations of surface positions cause different coloring. This can be achieved by applying lighting calculations to virtual objects. Highlights and shadows can provide information about an object's dimension and shape (see Figure 5.14 (b)). Thereby it has to be considered, that the visual system assumes that the light comes from above. The shading indicates that the left image in Figure 5.14 (b) is convex, whereas the right image appears concave. A totally different perception is obtained if the image is viewed upside down. In addition non-realistic shading methods can be used for shading an object. They often provide a better result for obtaining the shape and orientation of objects or object surfaces and are therefore often used for illustrative rendering. Furthermore shape information are communicated especially by regions with high curvature. Consequently, emphasizing of regions with high curvature can contribute to a better perception of shape. Shading of virtual objects is discussed in Section 6.3 in detail.

Instead of shading an object, perception of its shape can also be achieved by texturing it with regular textures. Regular textures always provide texture gradients (see Section 5.2.2)

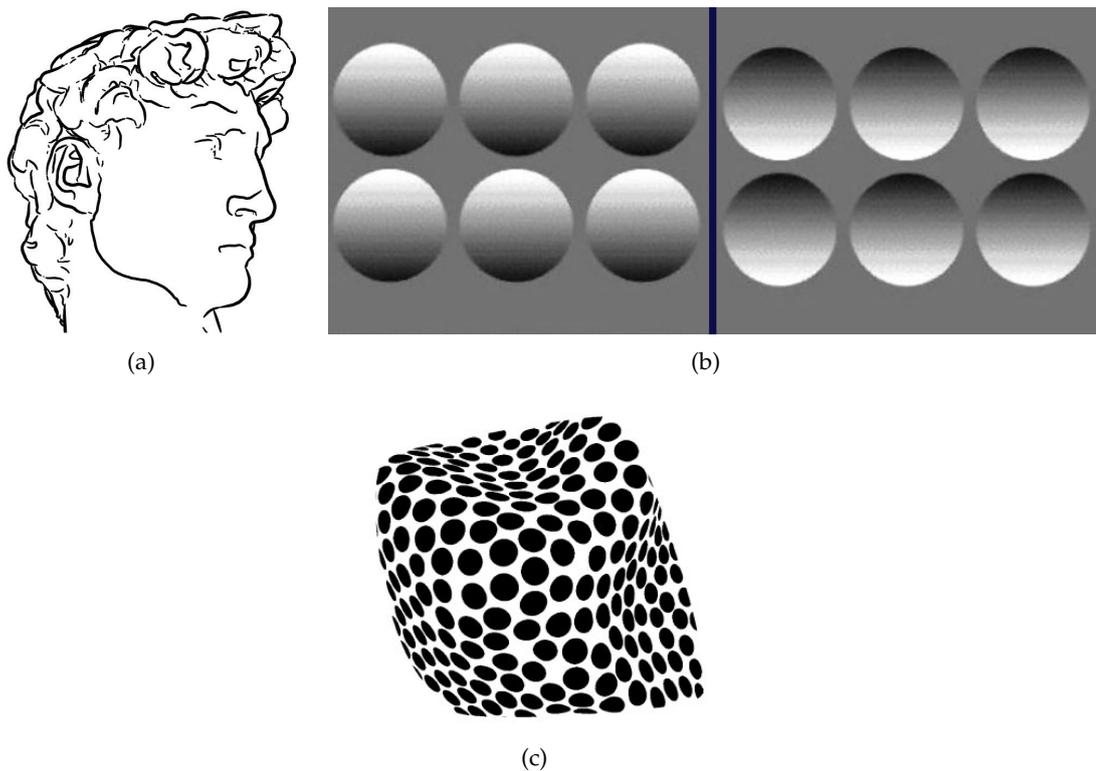


Figure 5.14: Perception of the shape of objects: (a) contours of a david bust [18], (b) provision of information about shape through highlights and shadowing and (c) the perception of shape through texture gradients [80].

as cues for the shape of the textured object. Todd et al. [80] examined the effect of regular textures, containing equal sized round black disks, which are positioned uniformly (like grid points) on a white background. The shape of the disks differs with the orientation of the surface position the disks are projected to. On the basis of the disk shapes, the visual system estimates their orientation and thus the shape of the object. Figure 5.14 (c) illustrates the effect of texture gradients for the perception of the shape of an object.

In addition, the visual system also exploits information gained from motion of objects for shape perception, which is denoted as kinetic depth information.

6 Principles of Designing Focus and Context Visualizations of Volumetric Data

This chapter discusses general principles of designing focus and context visualizations of volumetric (anatomic) data. The challenges of focus and context rendering are very similar to such of generating hand-made scientific illustrations (see Section 4.2). Generation of such visualizations means to automate illustration techniques and processes. However, especially in medical AR a correct perception of layout, distances and shape of objects, like discussed in Chapter 5, is of greater importance than in hand-made illustrations. When designing automated focus and context visualizations, correct perception of layout and distances has to be assured by remaining context fragments, whereas the correct perception of shape of virtual anatomy can be achieved by the use of specific shading methods. In general, automated focus and context visualization can be organized into the following building blocks:

- **Segmentation / Extraction**
 - of one or more **focus layers / objects**
 - of one or more **context layers / objects**
- **Shading** of the layers / objects: thereby generally a good perception of shape (see section 5.4) should be guaranteed, important features should be emphasized
- **Modulation of transparency or position of context layers / objects** in order to reveal the view onto the focus region and to provide cues for the perception of layout and distances (see sections 5.2 and 5.3); in general three techniques can be differentiated:
 - completely **cutting away** fragments hindering the view onto the focus (they are rendered completely transparent)
 - semi-transparent rendering of such fragments (**ghosting**), where the transparency of fragments can be adjusted by an importance measure, such that important features are rendered more opaque than unimportant regions
 - changing of the position of context fragments (**exploded views or deformations**)
- **Composition** of the layers / objects

Section 6.1 deals with the general possibilities of rendering volumetric data, respective the segmented focus and context layers. Section 6.2 presents possibilities for segmenting

volumetric data, respective the extraction of focus and context layer. Shading of virtual objects is discussed in Section 6.3. Section 6.4 discusses the three mentioned techniques for modulating of transparency respectively the position of context layers in more detail. The special challenges and tasks of designing focus and context visualization for medical AR are discussed in Section 6.5.

6.1 Visualization of Volumetric Data

This section focusses on the general methods for visualizing volumetric data. Generally, volumetric data sets are obtained by sampling, simulation, or modeling techniques. In the field of medicine, volumetric data can be obtained from different scanning technologies, such as CT or MRI. When scanning a human body 2D slices are generated, which can be three-dimensionally reconstructed into a volume model. Usually the slices are produced in regular displacements (e.g. one slice every millimeter) and consist of regular patterns of image pixels. The reconstructed volume model is thus arranged in a regular volumetric grid, where each voxel (volume element) is represented by a single data value (usually a value in a gray scale), which is denoted as iso-value. For example, in CT recordings bones have a lower data value than skin or muscles resulting in a brighter gray. CT images are usually stored in the "Digital Imaging and Communication in Medicine" (DICOM) format.

Volume visualization is a method of extracting and displaying meaningful information from volumetric data using computer graphics and imaging. It is concerned with volume data representation, modeling, manipulation and rendering. Volume visualization includes as well the extraction of (focus and context) layers as shading and transparency settings of the data. Different methods for rendering volumetric data have been developed. The following subsections provide a short overview of these methods.

6.1.1 Indirect Volume Rendering

Since methods for displaying geometric primitives have been well-established for a long time, most of the early methods approximate surfaces contained within the data by generating geometric primitives, like triangles or other polygons. Therefore an algorithm examines the volumetric data prior to rendering and generates a mesh of geometric primitives, corresponding to a given iso-value. Usually the marching cubes algorithm [59] is used for that purpose. It produces three dimensional vertices by detecting voxels, which are located at the boundary layer of homogeneous areas and connects them to a triangle mesh. After extracting this intermediate representation, hardware-accelerated rendering can be used to display the surface models. Related methods for indirect volume rendering include contour tracking [46], opaque cubes [36], dividing cubes [14] or marching tetrahedra [74]. In general, for every data sample these methods require to come to a decision whether the surface passes through the data sample or not. This can produce spurious surfaces or erroneous holes in surfaces. As information about the interior of objects is generally

not retained, a further drawback of these methods is, that one dimension of information is essentially lost.

6.1.2 Direct Volume Rendering

In response to the drawback of indirect volume rendering, direct volume rendering techniques were developed. These are techniques which do not require a prior surface extraction step, but generate the whole projection of the 3D data onto the 2D screen during rendering. A direct volume renderer samples the three-dimensional volumetric grid, whereby every sample value is mapped to an opacity value and a color value. This is performed by a *transfer function* which can be a simple ramp function, a piecewise linear function or an arbitrary table. Once converted to a RGBA (red, green, blue and alpha) value, the composed value is projected to corresponding pixels of the frame buffer. The way the sampling is performed and the nature of the transfer function depend on the rendering technique. Volume rendering comprises more information in a single image than traditional surface representations and is thus a valuable tool for the exploration and analysis of data. The disadvantages are costs of increased algorithm complexity and consequently a loss of rendering performance. The increasing programmability of the graphics hardware has enabled the application of several acceleration techniques for volume rendering [54, 69]. However, the performance of these approaches is heavily dependent on the hardware of the graphics processing unit (GPU). This section provides a brief overview of the different techniques of direct volume rendering. Combinations of these techniques are also possible.

Volume Ray Casting

Volume ray casting [56, 34] is an image-order algorithm that casts viewing rays through the volume. Image-order methods start from the pixels on the image plane and compute the contribution of the appropriate voxels to these pixels. The volume ray casting algorithm

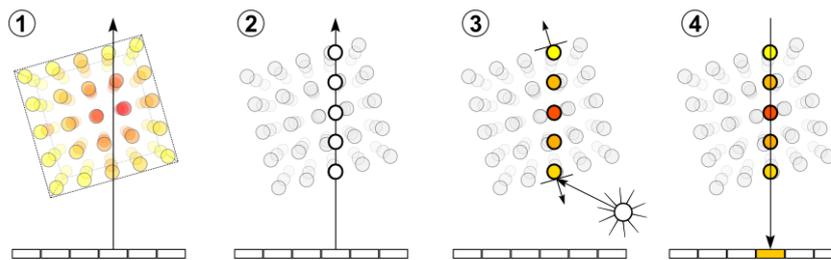


Figure 6.1: The four basic steps of volume ray casting [12]: (1) Ray Casting (2) Sampling (3) Shading (4) Compositing.

comprises four steps:

1. **Ray casting:** For each pixel of the final image, a ray of sight is cast through the volume, whereas it is useful to consider the volume being touched and enclosed within a bounding primitive - usually a cuboid - that is used to intersect the ray of sight and the volume.
2. **Sampling:** Along the ray of sight equidistant sampling points are selected. As in general the volume is not aligned with the ray of sight, sampling points usually will be located between voxels. Consequently, it is necessary to trilinearly interpolate the surrounding voxels to get the sample values.
3. **Shading:** For each sampling point, the gradient is computed. A gradient represents the orientation of a local surface position within the volume. Afterwards the samples are shaded, i. e. colored and lighted, according to their surface orientation and the source of light in the scene.
4. **Compositing:** After all sampling points have been shaded, they are composited along the ray of sight by blending the color values of all sampling points in back to front order. The composition results in the final color value for the processed pixel.

The four steps are illustrated in Figure 6.1. Ray casting provides results of very high quality at the price of long runtime and is usually considered as the best, but also as the slowest technique. One challenge in ray casting is the efficient skipping of non-contributing regions (i.e. regions that have been classified as transparent).

Splatting

Splatting [91] is an object-order technique for volume rendering, which trades quality for speed. In contrast to image-order techniques, object-order methods determine, for each data sample, how it affects the pixels on the image plane. Splatting is a technique that traverses and projects ("splats") voxels onto the image plane (like snow balls on a window) from in back to front order. Fully transparent voxels can thus be easily skipped. This is one of the greatest advantages of splatting, as it can tremendously reduce the amount of data that has to be processed. However, during the compositing step, undesired bleeding artefacts (i.e. the colors of hidden objects may "bleed" into the final image) can be produced.

Shear Warp

Image-order (like ray casting) and object-order algorithms (like splatting) have very distinct advantages and disadvantages. Therefore, some effort has been spent on combining the advantages of both approaches. Shear-warp [55] is such an algorithm. It is considered to be the fastest software-based volume rendering algorithm. It is based on a factorization of the viewing transformation into a shear and a warp transformation. The shear transformation transforms the nearest face of the cubic volume in a way that it becomes axis

aligned with an intermediate viewing image. It has the property that all viewing rays are parallel to the principal viewing axis in sheared-object-space. This allows volume and image to be traversed simultaneously. Once all slices of the volume have been rendered, the intermediate viewing image is warped into the desired orientation and scale and thus the final image is produced.

This technique is relatively fast in software at the cost of less accurate sampling and potentially worse image quality compared to ray casting.

Texture Mapping

A common and very fast volume rendering method exploits texture mapping capabilities of the GPU. Modern graphics hardware is fast at texturing and can efficiently render slices of a three-dimensional volumetric data. As well methods based on 2D texture mapping [68] as methods which apply 3D texture mapping [11, 25, 90, 61] exist. By using 2D texturing, a stack of two-dimensional textures, extracted from the volumetric data, is stored for each major viewing axis. The stack most parallel to the current viewing direction is chosen, whose textures are mapped on object-aligned proxy geometry which is rendered in back-to-front order using alpha blending. Object-aligned texturing produces images of reasonable quality, though there is often a noticeable transition when the volume is rotated.

3D texture mapping methods generate a single three-dimensional texture and upload it to the graphics hardware. The hardware is then used to map this texture onto polygons parallel to the viewing plane (view-aligned texturing), which are rendered in back-to-front order using alpha blending. 3D texture mapping allows to use trilinear interpolation supported by the graphics hardware and provides a consistent sampling rate. View aligned texturing can produce images of high quality. A problem of these volume rendering methods is the limited amount of video memory.

6.2 Extraction of Focus and Context Layers / Objects

This section discusses the extraction of focus and context layers (or objects). The focus layer contains the part of the data the observer is interested in. For the extraction of context layers, it has to be considered that parts of the data are chosen which are suitable for providing correct and useful context information. The extraction of focus and context layers or objects can be performed by segmenting the volumetric data prior to rendering or by extracting layers during rendering. These two possibilities are discussed in the following.

6.2.1 Segmentation of the Data Set prior to Rendering

One method for extracting the layers is to segment the volumetric data prior to rendering [82]. This can be performed by applying indirect volume rendering, which means extracting surface models in form of polygon meshes via the marching cube algorithm or other algorithms (see Section 6.1.1).

Instead of generating polygon models, the volumetric data can be segmented by just determining sets of voxels which belong to different objects. The objects then are usually in the form of segmentation masks. These masks can be used to selectively render only some of the objects contained in a single data set, or render different objects with different optical properties. An especially powerful approach is to apply different volume rendering methods to the pre-segmented objects and to combine these methods to a single volume rendering approach. The different pre-segmented objects can be rendered with individual and suitable per-object rendering modes, which allows an efficient separation of focus and context. A rendering method known as two-level volume rendering [35, 33] makes use of such segmentation information.

Viola et al. presented importance driven volume rendering [86]. This rendering method assigns importance values to pre-segmented objects. Thereby, objects with high importance values mark focus objects, whereas context objects are classified by a low importance value. For instance, a tumor and the vascular tree in close vicinity are chosen to be most important features, whereas the surrounding anatomy (bones, aorta, skin) is chosen to have a lower importance. During rendering by volume ray casting, the importance values of all objects, which are hit by a ray, are considered for color determination of a pixel.

6.2.2 Extraction of Layers during Rendering

Instead of pre-segmenting the volumetric data, the extraction of the different layers can also be performed during rendering. For instance, volume ray casting (see section 6.1.2) can be used to extract iso-surfaces corresponding to user a chosen iso-value or regions corresponding to a range of iso-values. Consequently, the user is able to adjust the visualized data during the application, which represents a higher degree of interactivity as pre-segmentation of data can offer. An approach for extracting focus and context layer via volume ray casting was introduced by Krueger et al. (*ClearView* an "interactive framework for texture-based volume ray-casting" [53]). They extract several 3D layers of volumetric data sets using texture-based ray casting and compose them. These layers are partitioned in a focus layer and several context layers, and stored in position maps, which are combined in a final compositing step. Detection of the focus can also be performed implicitly based on the distance to the observer's eyes, like shown by an approach of Bruckner et al. [9]. In this approach the distance is used as additional parameter for a special transfer function (see Section 6.1.2) and thus applied for transparency calculation of data fragments.

6.3 Shading

This section concerns shading of extracted focus and context layers of the volumetric data. The use of good and suitable shading methods is significant for the correct perception of shape (see Section 5.4) of focus layers (objects), but also of virtual context layers, whose transparency (or position) is also modulated (see Section 6.4).

Shading methods usually make use of the computation of normal vectors on the layer surface. During volume ray casting the normal vector of a surface position can be determined by first calculating the gradient of the sampled position. In sampled volume data, surfaces are implicitly represented as iso-surfaces of reconstructed continuous data values $f(\vec{x})$, which form a scalar field. "Assuming that the values of $f(\vec{x})$ increase as we move further inside objects of interest (e.g., a standard CT scan), the surface normal is defined as $\vec{n} = \frac{-\vec{g}}{|\vec{g}|}$, with the gradient

$$\vec{g} = \nabla f = \left[\frac{\partial f}{\partial x} \frac{\partial f}{\partial y} \frac{\partial f}{\partial z} \right] \text{ [49].} \quad (6.1)$$

The gradient represents the first partial derivatives of a position \vec{x} on the surface ($f(\vec{x})$). In the following several methods are described, which exploit information of local gradients, respectively normal vectors, for the shading of extracted iso-surfaces. These methods are Blinn-Phong shading (see Section 6.3.1), tone-based shading (see Section 6.3.2) and contour rendering (see Section 6.3.3). Physically based illumination models like ray tracing or radiosity are not taken into account, as they are too slow to achieve interactive framerates and are usually not used in volume rendering methods. By means of gradient information also curvature information can be computed and used for shading, which is discussed in Section 6.3.4. Finally texturing of surfaces (see Section 6.3.5) is discussed.

6.3.1 Blinn-Phong Shading

The Phong reflection model [67] is widely used in computer graphics. It represents an illumination and shading model for assigning shades to points on a surface. The Phong reflection model is not physically correct, but provides realistic looking illumination, which is computed very fast and efficiently. It is a local reflection model, i.e. it doesn't account for second-order reflections, as do ray tracing and radiosity. Only direct reflection is taken into account. The model describes the reflection of light as a combination of an ambient ($I_{ambient}$), a diffuse ($I_{diffuse}$) and a specular ($I_{specular}$) reflection term, whereas the ambient term compensates the loss of indirectly reflected light. The shaded color is computed by multiplying the input color (e.g. the color of a sample as obtained through the transfer function) by the sum of the three terms:

$$c_{out} = c_{in}(I_{ambient} + I_{diffuse} + I_{specular}) \quad (6.2)$$

The ambient component of the reflected light is constant and simulates the contribution of indirect reflections:

$$I_{ambient} = k_{ambient} \cdot I_a \quad (6.3)$$

I_a is a constant intensity value of the light in the environment and $k_{ambient}$ defines the fraction of ambient light reflection of the total reflection.

The diffuse term is based on Lambert's cosine law which states that the reflection of a perfect rough surface is proportional to the cosine of the angle between the light vector \vec{l}

and the surface normal vector \vec{n} .

$$I_{diffuse} = k_{diffuse} \cdot \max(\vec{l} \cdot \vec{n}, 0) \quad (6.4)$$

$k_{diffuse}$ defines the fraction of the diffuse light reflection of the total reflection.

The specular term adds an artificial highlight to simulate specular reflections. For computing the specular term, the phong model originally uses the angle between the ideal reflection direction \vec{r} and the view direction of the observer \vec{v} . Blinn proposed to replace this angle by the angle between the half-way vector \vec{h} - a vector halfway between the light vector \vec{l} and the view vector \vec{v} - and the normal vector \vec{n} , which leads to similar results and can be computed more efficiently [7]. The specular lighting intensity is proportional to the cosine of the angle between the half-way vector \vec{h} and the surface normal \vec{n} , which is calculated as dot-product of the two vectors. This dot-product is raised to the power of α , where α is called the specular exponent of the surface and represents its shininess. Higher values of α lead to smaller, sharper highlights, whereas lower values result in large and soft highlights.

$$I_{specular} = k_{specular} \cdot \max((\vec{h} \cdot \vec{n})^\alpha, 0) \quad (6.5)$$

$k_{specular}$ defines the fraction of diffuse light reflection of the total reflection. It holds that $k_{ambient} + k_{diffuse} + k_{specular} = 1$. The modified reflection model of Phong is denoted as Blinn-Phong shading model.

6.3.2 Tone-Based Shading

Alternatively to Blinn-Phong shading, tone-based shading [29] can be used for shading iso-surfaces. It replaces standard diffuse shading (compare the diffuse term in the Phong illumination model) with a blend between cool and warm colors, and is especially useful for non-photorealistic rendering and automated generation of technical illustrations. It is based on the model of artistic and illustrative painting, such as air-brush and pen, where often hue and luminance (greyscale intensity) shifts are applied. Instead of using reflectance information, this method uses hue changes to indicate surface orientation. Parts of the surface pointing away from the light source are colored in a cold, blueish hue (instead of a black color in the standard diffuse shading), whereas parts of the surface pointing towards the light source are colored in warm, yellow or red hue. Gooch et al. [29] proposed the following intensity equation for tone-based shading:

$$I = \left(\frac{1 + \vec{l} \cdot \vec{n}}{2} \right) k_{cool} + \left(1 - \frac{1 + \vec{l} \cdot \vec{n}}{2} \right) k_{warm} \quad (6.6)$$

Hereby the cosine term $(\vec{l} \cdot \vec{n})$ of the traditional diffuse light calculation is maintained. They recommend that the light vector \vec{l} should be perpendicular to the gaze direction. Since the human vision system assumes illumination coming from above (compare with Section 5.4), they positioned the light up and to the right and kept this position constant

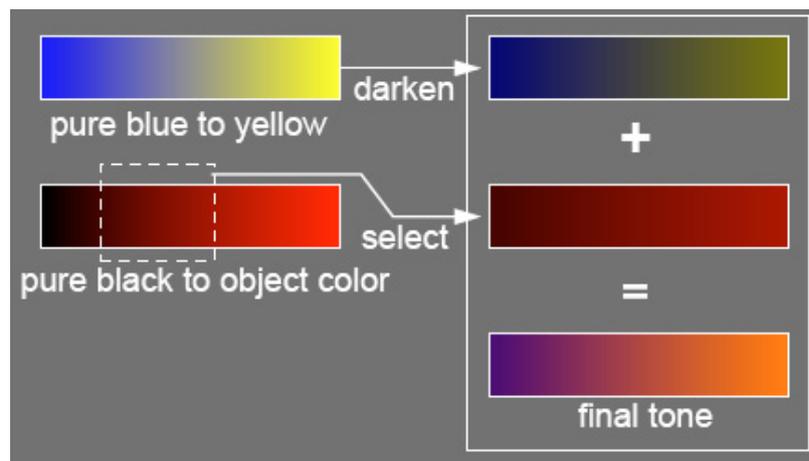


Figure 6.2: Gooch shading [29]: The creation of the tone for a pure red object, whereas a blue-to-yellow and a dark-red-to-red tone are summed up.

[29]. The equation blends between two different RGB-colors k_{cool} and k_{warm} . In order to generate these two colors, they propose a linear combination of blue-to-yellow and black-to-object-color tones. "Blue and yellow tones are chosen to insure a cool to warm color transition regardless of the diffuse color of the object" [29]. Therefore they specify a fully saturate blue $k_{blue} = (0, 0, b)$, $b \in [0, 1]$ in RGB space and a fully saturate yellow $k_{yellow} = (y, y, 0)$, $y \in [0, 1]$. They simulate undertones by a linear blend between the blue/yellow and black/object-color tones:

$$k_{cool} = k_{blue} + \alpha k_d, \quad k_{warm} = k_{yellow} + \beta k_d. \quad (6.7)$$

Here k_d represents the object's diffuse reflectance. Plugging these values into Equation 6.6 enables four free parameters: b , y , α , and β . "The values for b and y will determine the strength of the overall temperature shift, and the values of α and β will determine the prominence of the object color and the strength of the luminance shift" [29]. An example of a resulting tone for a pure red object is shown in Figure 6.2. This shading method is also denoted as *Gooch shading*.

A comparison of the Blinn-Phong shading method and the Gooch shading method is illustrated in Figure 6.3. By using Blinn-Phong shading, a loss of surface details occurs especially in dark regions, which lie in the shadow. Tone-based shading provides a clearer picture of shape, structure, and material composition than Blinn-Phong shading. However, this shading method has the drawback that shadowing and self-shadowing can hardly be applied additionally, which causes a loss of important depth information compared to Blinn-Phong shading (see Section 5.9). Gooch shading also uses black contour lines and white highlights. The highlights are rendered using the traditional exponential term from the Phong lighting model. Highlights and contour lines are drawn, because "then very

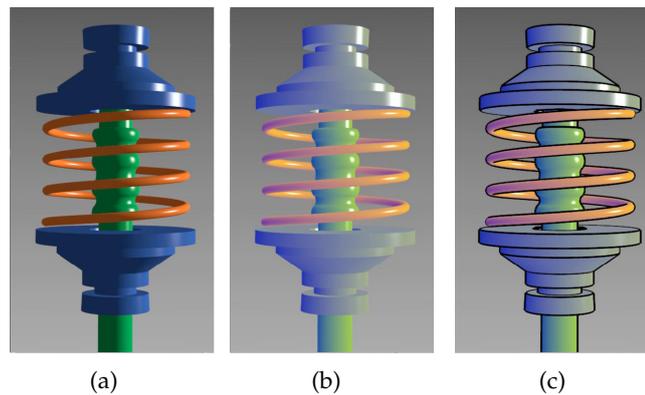


Figure 6.3: Comparison of shading models [29]: Blinn-Phong shading (a), tone-based shading with additional highlights (b) and tone-based shading with additional highlights and contour lines (c).

low dynamic range shading is needed for the interior” [29]. A gooch shaded model with additional highlights and contour lines, is shown in Figure 6.3 (c). The next subsection will discuss the rendering of contours in detail.

6.3.3 Contour Rendering

Surfaces can be emphasized by the additional drawing of contour lines (sometimes also denoted as phantom lines). The drawing of contours is also performed in line art and illustrations. Contours are very important descriptors for the shape of an object (see Section 5.4) and clarify sites of occlusion [51, 29]. Studies in the field of perception demonstrate that objects can be completely recognized and their shape perceived, when only contours are drawn [6, 13]. In 3D computer graphics they are defined as “2D projection of points on the 3D surface where the direction of the surface normal is orthogonal to the line of sight” [41] and are thus determined by a sign change of $\vec{v} \cdot \vec{n}$, the scalar product between the view vector \vec{v} and the face normal \vec{n} [30]. Consequently, contours are view dependent and emphasize the transition between front-facing and back-facing surfaces. Volume rendered contours are generally produced with an univariate function of $\vec{v} \cdot \vec{n}$, whereas the surface color is darkened when the result of the function is within a user-chosen range around zero. Contours are always rendered in a certain fixed thickness and are often used in conjunction with tone-based rendering, where they provide additional shape cues [29] (see Figure 6.3 (c)). Using them in conjunction with Blinn-Phong shading or traditional diffuse shading can also provide reasonable shape cues but is less effective, since the lines would be lost in the dark regions [29].

6.3.4 Curvature-Based Shading

A further method for shading iso-surfaces exploits local curvature properties of the surface. Intuitively, curvature can be considered as the amount by which a surface region deviates from being flat. Simplified, the curvature of a point on a surface can be determined by comparing the direction of the normal of this position with the directions of the normals of points in the near environment. Curvature information is only dependent on surface properties and not dependent on the position of the observer's eyes. In contrast to contours, curvature information is thus not view-dependent. Shading by means of curvature information is also able to emphasize shape properties of objects, since thereby distinctive sites, which are characterized by high curvature values, are more emphasized than flat and unimportant regions.

In addition to the gradient (respectively the normal vector), the first order partial derivatives of the volume, implicit principal curvature information can be computed from the second order partial derivatives of the volumetric data. Consequently, curvature generally means changes of the gradient in a scalar field, respectively an iso-surface. After Kindlmann et al., curvature at an arbitrary point in a scalar field (the parameterized surface $f(\vec{x})$) can be computed by applying the following steps [49]:

1. Measure the first partial derivatives comprising the gradient \vec{g} . Compute $\vec{n} = \frac{-\vec{g}}{|\vec{g}|}$, and $\mathbf{P} = \mathbf{I} - \vec{n}\vec{n}^T$, where \mathbf{I} is the identity matrix.
2. Measure the second partial derivatives comprising the Hessian matrix $\mathbf{H} = \nabla\vec{g}$. Compute the geometry tensor $\mathbf{G} = -\mathbf{P}\mathbf{H}\mathbf{P}/|\vec{g}|$.
3. Compute the trace T and Frobenius norm F of \mathbf{G} . Then,

$$\kappa_1 = \frac{T + \sqrt{2F^2 - T^2}}{2}, \kappa_2 = \frac{T - \sqrt{2F^2 - T^2}}{2}.$$

κ_1 and κ_2 are called the first and second principal curvature magnitudes of a point on the iso-surface. κ_1 is the curvature magnitude of principal direction associated with the largest, positive curvature and κ_2 is the curvature magnitude of the principal direction associated with the largest, negative curvature. As seen above, κ_1 and κ_2 can be computed directly from the gradient \vec{g} and the Hessian matrix \mathbf{H} , which is comprised of all second order partial derivatives of the scalar volume f . For instance, the second order partial derivatives can be computed by convolving the scalar volume with a combination of first and second order derivatives of a cubic B-spline kernel [85]. $(\kappa_1 + \kappa_2)/2$ is denoted as the *mean curvature* at a position on the surface, $\kappa_1\kappa_2$ as the *Gaussian curvature* and $\sqrt{\kappa_1^2 + \kappa_2^2}$ as the *total curvature* [49].

Color Mapping of Curvature Magnitudes

The principal curvature magnitudes κ_1 and κ_2 can be visualized on an iso-surface by mapping them to colors via one-dimensional transfer function lookup textures. The same

approach can be used to depict additional curvature measures directly derived from the principal magnitudes, such as mean curvature, Gaussian curvature or total curvature (see Figure 6.4). Even more powerful are two-dimensional transfer functions in the 2D domain

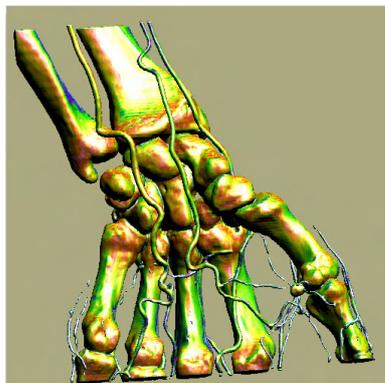


Figure 6.4: CT scan (256x128x256) with color mapping from a 1D transfer function depicting $\sqrt{\kappa_1^2 + \kappa_2^2}$ (total curvature) [85]

of both principal curvature magnitudes (κ_1, κ_2) , since color specification in this domain allows to highlight different structures on the surface. Figure 6.5 (a) shows a sequence of quadratic surface patches, whereas a two-dimensional transfer function was used to map the κ_1 and κ_2 values of the surface points to color values. Figure 6.5 (b) shows an ear, which was shaded with the same transfer function.

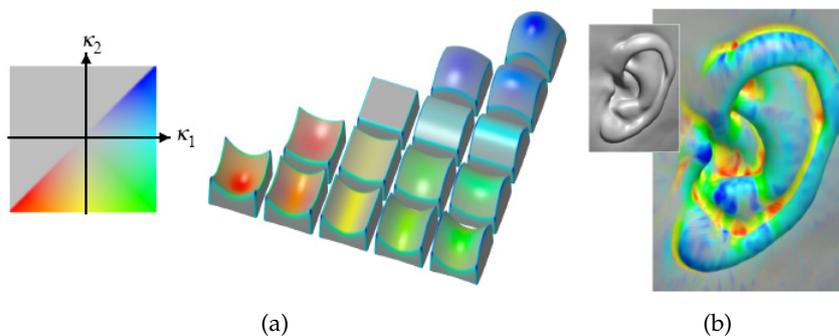


Figure 6.5: (a) Volume rendered diagram of (κ_1, κ_2) space. The colors in the (κ_1, κ_2) transfer function domain are mapped onto the patches with corresponding surface curvature. (b) Visualization of ear curvature using transfer function from (a).

Highlighting of Ridge and Valley Lines

Another possibility of exploiting curvature information, is the emphasis of ridge and valley lines [41, 49]. Valley lines are defined as “the locus of points on a surface at which the normal curvature assumes a local minimum in the principal direction associated with the largest, negative curvature” [41], which means a local minimum of κ_2 . Whereas ridge lines are “the locus of points at which the normal curvature in the principal direction associated with the largest, positive curvature assumes a local maximum” [41], which means a local maximum of κ_1 . While valleys generally correspond to darker portions of the surface, ridges are more likely to reflect a specular highlight [31]. Ridge and valley emphasis can be done by using special two-dimensional transfer functions in the domain of (κ_1, κ_2) . Such a transfer function is illustrated in Figure 6.6 (a), which follows the convention of darkening valleys and lightening ridges. Figure 6.6 (b) shows ridge and valley emphasis in conjunction with tone-based shading (see Section 6.3.2) and contour emphasis (see Section 6.3.3).

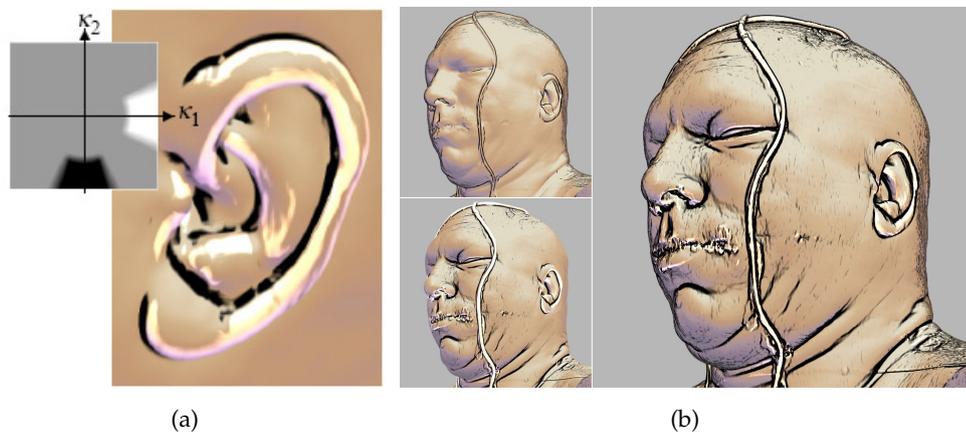


Figure 6.6: Ridge and valley emphasis [49] implemented with the illustrated two-dimensional transfer function (a). Ridge and valley emphasis in conjunction with tone-based shading and contour emphasis (the two small images depict the contribution of contours (top) and ridge and valley emphasis (bottom), whereas the large image shows the combination of the two effects)

Flow Visualization of Principal Curvature Directions

As visualizing principle curvature magnitudes (see section 6.3.4 and 6.3.4), principle curvature directions of an iso-surface can also be used for shading purposes. The principal curvature directions, which mean the directions associated with the largest, positive and negative curvature, are found as eigenvectors of the matrix \mathbf{G} [49] (see above). The curvature directions can be mapped directly to RGB colors, but such a mapping is hard to

interpret. An alternative approach is the visualization of principal curvature directions using image-based flow visualization [84]. An example of the advection of flow along principle curvature directions is shown in Figure 6.7.



Figure 6.7: Flow visualization of principal curvature directions [84].

6.3.5 Solid Texturing

A triangle mesh or a position image, which contains ray-isosurface intersection positions, can be used for the application of a solid texture onto an isosurface. Therefore a adequate parameterization has to be found. For a texture containing ray-isosurface intersection positions this is performed by specifying the transformation of object space to texture space coordinates, e.g. via an affine transformation. Texturing of a surface with a solid grid, like discussed in Section 5.4, is able to emphasize the shape of an object.

6.4 Modulation of Context Layers

The extracted context layers (or objects) usually occlude (parts of) the extracted focus layer, which thus might be not or only partly perceivable. In order to reveal the view onto the focus the transparency or the position of occluding context fragments has to be modulated. By doing that, it always has to be considered that enough context information remains, in order to maintain important cues for perceiving layout and distances of the focus (see Chapter 5). On the one hand as much context information as possible should be left, but on the other hand a preferable clear view onto the focus should be provided. Three techniques for handling this issue are discussed in the following subsections. All techniques were adapted from techniques used in traditional hand-made illustrations.

6.4.1 Cut-away techniques

One method to resolve the visibility problem is to cut away any fraction of the context layers which occludes parts of the focus. By applying this method first, a complete and

clear view on the focussed part of the data is provided. Second, interior parts of the data, which do not belong to the focus still remain occluded. Consequently the depth cue of occlusion (see Section 5.2.2) does not get lost. Context information are thus maintained on the side of the focus, but not in front.

Figure 4.1 (c) shows an example of a hand-made technical illustration, where this cut-away technique was applied. In computer graphics generally simple geometries like clipping planes or more complex three-dimensional clipping volumes are used to clip away the outer hull of an object. A clipping plane defines two half-spaces. The information that spatially belongs to one half-space is visible, while the other is not displayed. A clipping volume removes any part of the data which is inside of it and can be placed arbitrarily inside the visualized dataset. An automatic generation of cut-away views (and ghosted views, which are discussed in Section 6.4.2) for polygonal data was introduced by Feiner and Seligmann [23]. They propose a family of algorithms that automatically identify potentially obscuring objects and display them in a ghosted or cut-away view. The proposed algorithms exploit depth buffer rendering and are therefore suitable for real-time interaction achieved by hardware acceleration. Diepstraten et al. [20] discussed a further approach for cutting away focus obscuring parts of surface rendered polygon models. In their paper they distinguish between two different subclasses of the general notion of a cutaway drawing: *cutout* and *breakaway*. "A *cutout* in a technical illustration has not always to be smooth. For example, a sawtooth-like or jittering cutting is often applied to better distinguish between outer and inner objects and produce a higher level of abstraction" [20]. Applying a *cutout* is useful "when many objects or large objects are inside and cover a large portion of the interior of the girdling object" [20]. If only a few small inside objects lie densely to each other the *breakaway* method is used. Hereby "a virtual hole" is broken into the boundary to show the interior objects and which "should be just wide enough to see these objects" [20]. Figures 6.8 (a) and 6.8 (b) show the difference between the two subclasses. For creating *cutout* views they propose to follow five rules [20]:

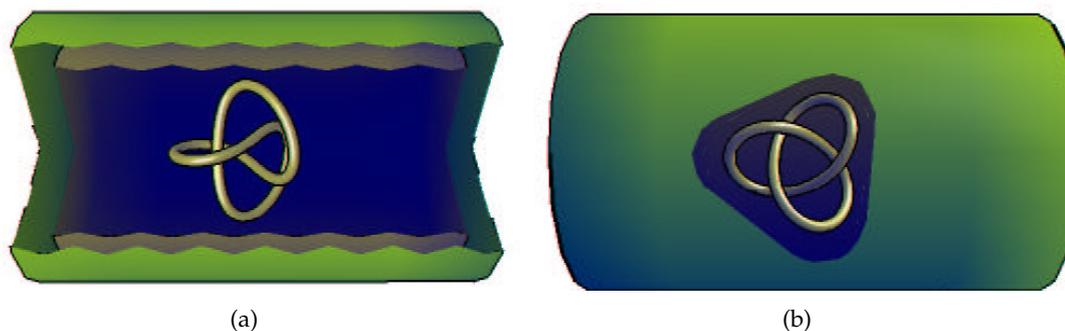


Figure 6.8: Comparison of computer-generated cutout and breakaway illustrations. Figure (a) demonstrates the cutout technique with a jittering boundary, in the Figure (b) the breakaway method is applied to the same scene. [20]

- Inside and outside objects have to be distinguished from each other.
- The cutout geometry is represented by the intersection of (a few) half spaces¹.
- The cutout is located at or around the main axis of the outside object.
- An optional jittering mechanism is useful to allow for rough cutouts.
- A possibility to make the wall visible is needed.

For *breakaway* illustrations the second and the third rule are substituted with [20]:

- The breakaway should be realized by a single hole in the outside object.
- All interior objects should be visible from any given viewing angle.

Weiskopf et al. [89] performed GPU-accelerated clipping for volume rendered data with arbitrary convex and concave clipping objects (see Figure 6.9 (a)). They proposed a number

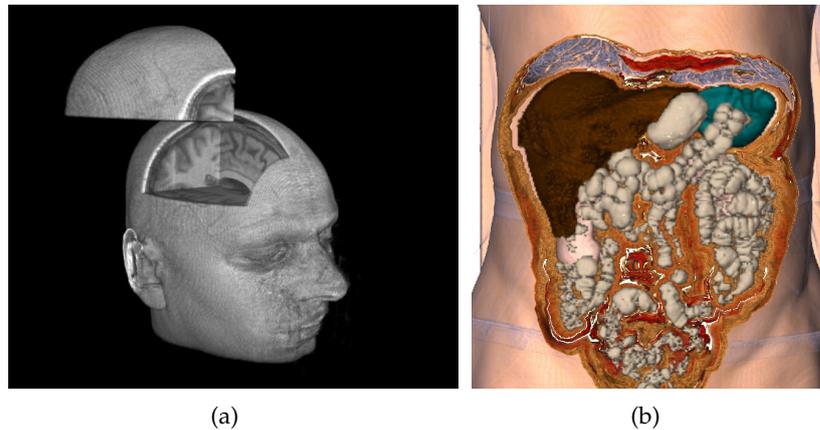


Figure 6.9: (a) Volume clipping [89]. (b) Cut-away by means of importance-driven volume rendering [86].

of effective techniques to increase performance and visual quality. Their implementation of clipping operations is mapped to commodity graphics hardware to achieve interactive framerates. Additionally to clipping all rendering computations are performed on the graphics hardware. Per-fragment operations estimate on-the-fly the visibility according to the clipping geometry and adjust the shading in areas where clipping occurs. Konrad-Verse et al. [52] use a mesh which can be flexibly deformed by the user with an adjustable sphere of influence. Zhou et al. [95] propose the use of distance to emphasize and de-emphasize different regions. They use the distance from the eye point to directly modulate

¹A half space can be represented by the plane that separates the space from its complement

the opacity at each sample position. Thus, their approach can be seen as a generalization of binary clipping. An automated way of performing clipping operations has been presented by Viola et al. [86]. They use importance-driven volume rendering, a framework which applies different compositing strategies to prevent an object from being occluded by a less important object (see Figure 6.9 (b)). Volume sculpting, proposed by Wang and Kaufman [88], enables interactive carving of volumetric data. A similar approach is presented by Pflesser et al. [66], who offer an interactive drill-like tool for carving away parts of the volume in order to perform surgical training. Owada et al. [65] extended volume cutting by incorporating two-dimensional textures that are mapped on the cut surface. This enhances the visualization with additional information of the internal arrangement of bones or muscles. Such a concept can be very useful for anatomy education for example. Straka et al. [77] are applying a cut-away technique for CT-Angiography.

Cut-away approaches could eliminate fragments also in regions, where it would not be necessary. Thus too much context information and therewith important positional and depth cues could be lost. The following chapter introduces a method which is able to avoid this problem.

6.4.2 Ghosting: Creating a Semi-Transparent Context

An alternative to cut away fractions of the data hindering the view onto the focus region is to render those fractions in a semi-transparent way (see Section 5.3). This technique is called ghosting. As clipping operations are generally not data-aware, they do not take into account features of the context objects, positioned in front of the focus. Consequently, clipping can also remove important context information which can lead to confusing and partly misleading result images, where certain visual cues for perception of distances and layout are absent. Instead of removing the occluding regions completely, ghosting reduces opacity selectively in a way which attempts to preserve features such as contour or curvature information, which offer important cues for the shape of exterior objects (see Section 5.4) and thus context information. In hand-made illustration, where artists want to visualize specific internal structures as well as the exterior ones, the use of ghosting is common. There less significant items are reduced to an illusion of transparency. For example, rather flat surfaces are faded from opaque to transparent to reveal the interior of an object. Detailed and prominent structures are still displayed with high opacity. The goal of this illustration technique is to provide enough hints to enable viewers to mentally complete the partly removed structures. This technique can thus avoid the drawback of missing context information and gives a better impression of the spatial location of the object in focus than the cut-away technique. However, ghosting has the drawback that there is always a certain disturbance of the view onto the focus. Figure 4.1 (a) and (b) show two examples for the application of ghosting in the field of technical illustration. Figure 4.1 (c) illustrates ghosting in connection with the cutaway technique. It contains three context layers and one focus layer, where ghosting is applied on the outer context layer and cut-away on the other context layers.

Diepstraten et al. [19] propose to adhere to an effective set of guidelines for using transparency for the automatic generation of illustrative visualizations. They focused on “illustrations consisting of smoothly shaded, colored surfaces” [19]. In order to avoid disturbed and misleading perception, they advise the consideration of the following three rules for the generation of semitransparent illustrative views:

- Faces of transparent context objects never shine through other transparent faces.
- Opaque objects which are occluded by two transparent objects do not shine through.
- Transparency falls off close to the edges of transparent context objects and increases with the distance to edges.

They thus recommend to use only one semi-transparent context layer, since several transparent objects can cause confusing result images (see also Section 5.3).

For rendering semi-transparent context fragments in the front of the focus, it is possible to simply assign an uniform transparency value to all context fragments, but this can cause a misleading and disturbed view of the focus objects, as it can be seen Figure 6.10. Thus it is



Figure 6.10: An uniform transparency setting of the context object can result in a misleading perception

helpful to adjust the transparency setting in such a way that important features and shape cues are emphasized. In order to preserve shape cues and thus context information of focus occluding fragments and to simultaneously guarantee a preferable clear view onto the focus, several techniques can be applied, which are discussed in the following subsections.

Regular Grid as Context on Transparent Surface

One method of highlighting the shape of a context layer, in order to provide positional cues for the focus objects, is the superimposition of a regular grid on the context layer. This

enables the depth and shape cue of texture gradients (see section 5.4). The grid lines or regularly spaced grid elements are rendered fully opaque, whereas the rest of the context layer remains as transparent so that it provides an unhindered view onto the focus objects. Among others, this technique was performed by Levoy et al. [57], who use solid grid textures on layered transparent surfaces of volume data to enhance the communication of shape and position information. The difference of rendering the context layer with and without the grid is illustrated in Figures 6.11. Despite of giving shape and positional cues,

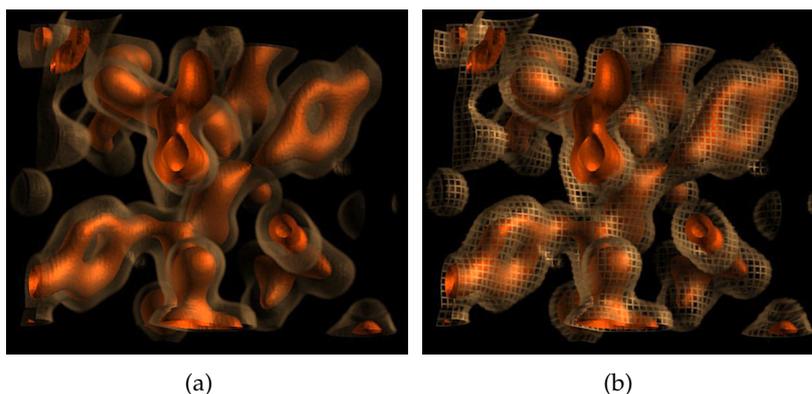


Figure 6.11: Regular grid on transparent surface providing context information: visualization of transparent isoelectron density surfaces around molecules [57] (a) without regular grid lines and (b) with regular grid lines.

this technique has the disadvantage that the view onto the focus might be disturbed by the grid lines and thus no clear observation of the focus is possible.

Contour-Based Modulation of Context Layers

Since contours are highly significant for the perception of shape (see Section 5.4), a more elegant method for preserving of context objects than the superimposition of regular grid lines, is the extraction and displaying of contour lines, as already discussed in Section 6.3.3. In addition to the rest of the context layer, which remain semi- or completely transparent, the contour lines are rendered fully opaque. Csebfalvi et al. [16] use non-photorealistic volume ray casting for visualizing object contours. There contours are visualized regardless of their iso-values. Instead of the original data values, a function of gradient magnitudes is used in order to characterize the "surfaceness" of a voxel. Data values, based on the magnitude of local gradient information as well as on the angle between viewing direction and gradient vector (compare with Section 6.3.3), are mapped to a color and a opacity value. For each pixel, rays are casted that collect all the color and opacity values along the ray. The values of each ray are combined using alpha blending or maximum intensity projection (MIP), where MIP assigns the highest sample intensity of the sample points along

a viewing ray. The result of the combination of all color and opacity values along a ray is that, 3D structures which are characterized by regions exhibiting significant changes of data values, become visible without being obstructed by visual representations of rather continuous regions within the 3D data set. The discussed contour rendering technique can be applied in combination with surface rendering (see Figure 6.12). By doing that the rendered surface model represents the focused object, whereas the contours represent the context information.

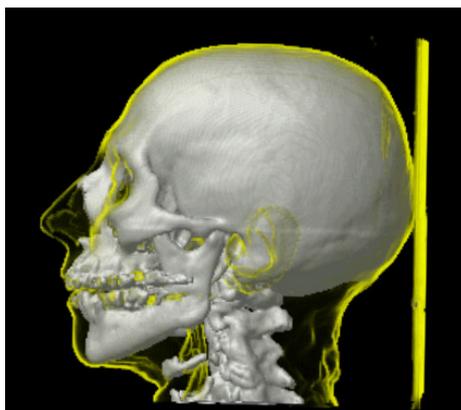


Figure 6.12: Ghosting by contour rendering [16].

Despite of providing a clear view onto the focus, contour rendering has the drawback that details of the transparent objects are lost and that it ignores material and surface information of the context objects [19]. Furthermore it only provides “two transparency states: fully opaque or fully non-opaque; semi-transparency cannot be visualized” [19]. However, it is also possible to produce a smooth transparency transition between contours and non-contour regions, which is presented in Section 7.3.2.

Curvature-Based Modulation of Context Layers

A further method for the modulation of context fragments is the exploit of local curvature (see Section 6.3.4) for transparency calculation. As the ratio of the normal directions and thus the curvature values remain equal during movement of the observer, curvature values are not view dependent (in contrast to contour lines). One approach is the usage of curvature values in order to compute and display ridge and valley lines [41]. Like mentioned in section 6.3.4, valleys generally correspond to darker portions of the surface, whereas ridges are more likely to reflect a specular highlight. Interrante et al. [41] suggested to augment semi-transparent iso-surfaces occluding the focus region by ridge and valley lines (see Figure 6.13 (a)). Their approach is based on the observation that self-shadowing causes valley lines normally to be less brightly lit than surrounding surface areas and valley lines thus gives visual cues for the shape of an object. They refer to psy-

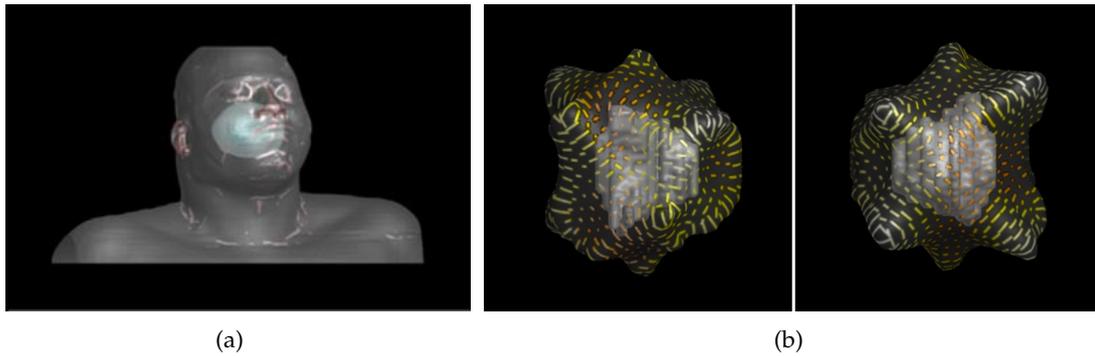


Figure 6.13: Enhancement of the context layer through (a) ridges and valleys [41] and (b) strokes at evenly distributed points over an isosurface which are directed by the local curvature to define the orientation and length of a polygonally-defined solid stroke texture [42].

chophysical studies [8], which have verified a theory stating that people tend to perceive objects as naturally partitioning along their valley lines. Further approaches of Interrante et al. enhance the context surface by using curvature-directed strokes [42] (see Figure 6.13 (b)) and 3D Line Integral Convolution [40] and are based on their previous approach. They visualize the 3D shape of smoothly curving transparent surfaces by utilizing the principal curvature directions (see Section 6.3.4) of surfaces. Generally, principal curvature magnitudes or curvature measures derived from the principal magnitudes could also be used for transparency modulation of context layers. Therefore the transfer functions has to map the curvature magnitudes, respectively the derived curvature measures, to a transparency value, instead to color value.

A method, which applies a smooth transition of transparency between regions of high curvature and regions of less curvature was presented by Krueger et al. [53]. During extraction of focus and context layers via volume ray casting, they store the ray-isosurface intersection positions in a position map. Afterwards, they approximate the curvature of every position in the map by comparing the direction of a texel's (texture element) normal with the normal directions of its four neighboring texels. This results in curvature values close to zero in regions of low normal variation and large values otherwise. These values are used for transparency determination of all context fragments. Consequently, a smooth transition between sites having a high curvature, which are necessary for shape perception, and unimportant sites of the context layer, having a low curvature value, is enabled. This effect is shown in Figure 6.14.

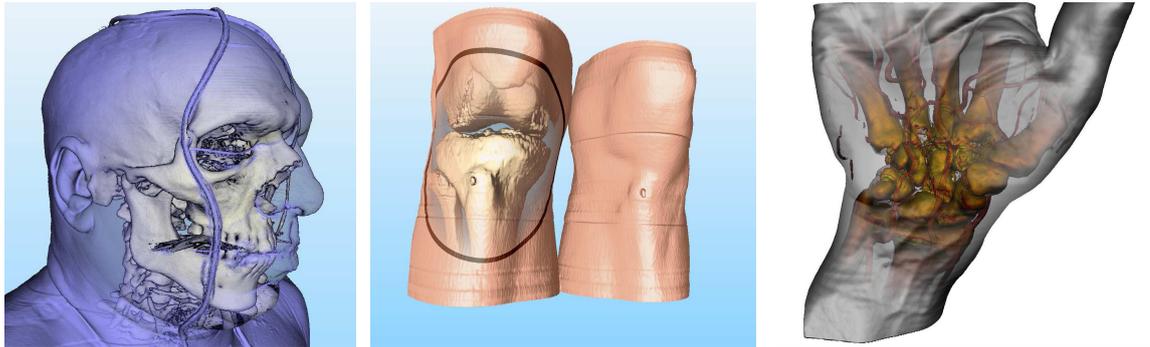


Figure 6.14: Transparency modulation by the exploit of local curvature [53].

Fading-out Transparency by Distance to Focus

In focus and context illustrations the opacity of context fragments often increases by increasing distance to the focus point. This fading-out of the context transparency attracts the gaze of the observer to the focus by forming a hotspot region. Thus, the context layer is divided in a fully opaque and a transparent region. This illustration technique can also be adapted from automated focus and context visualization and was additionally implemented by the visualization method of Krueger et al. [53], which was already discussed in the previous subsection (see Figure 6.14).

Distance-Based and View-Distance-Based Transparency

A further method for handling the transparency of the context layer is the usage of an distance-based or a view-distance-based importance measure [53]. For determining the transparency, the distance-based importance measure considers the distance from the context surface to the focus surface in the direction of the context normal (see Figure 6.15 (a)). Whereas the view-distance-based importance measure computes the distance from the context surface to the focus surface into the direction of a fragment's normal (see Figure 6.15 (a)).

Transparency Settings by means of Pre-Segmented Importance Areas of the Volumetric Data

The approach on importance-driven ray casting of Viola et al., already discussed in Section 6.2.1, assigns importance values to pre-segmented data regions. During ray-casting the transparency of the hit regions are adjusted by their assigned importance values. An example of this technique is shown in Figure 6.16.

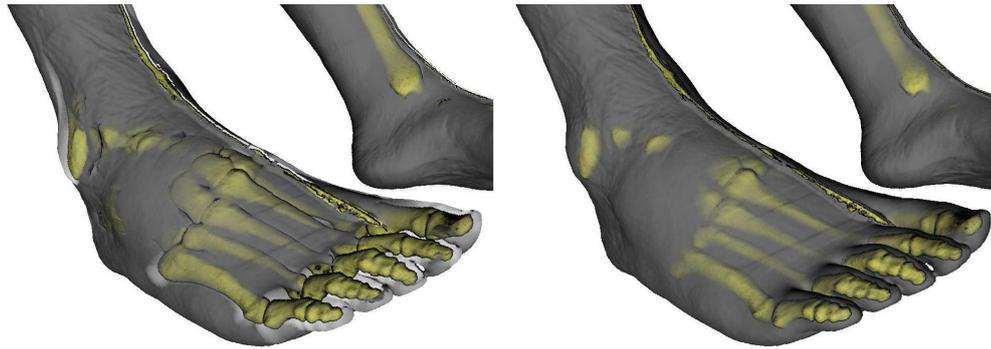


Figure 6.15: (a) Distance-based and (b) view-distance-based transparency modulation of context layer [53].



Figure 6.16: Transparency adjustment by importance values of pre-segmented data regions [86].

Transparency Settings by Calculation of Light Reflection

A rather new approach for transparency modulation of context fragments is denoted as *context-preserving volume rendering*, introduced by Bruckner et al. [9]. For this method a directional light source is placed at the position of the viewer's eye and the Blinn-Phong shading model (see Section 6.3.1) is used to calculate light values of every context fragment, which is hit by a ray during ray-casting. However, these light values are not used for manipulating the color values of the fragments, but for their transparency determination. Their "idea is based on the observation that large regions of high lighting intensity usually correspond to rather homogenous areas which do not contain characteristic features. While the position and shape of specular highlights, for example, give good cues for perceiving the curvature of a surface, the area inside the highlight could also be used to display other information" [9]. Large regions of highly illuminated material normally correspond to rather flat surfaces that are oriented towards the light source. In these regions

opacity is reduced. Regions, on the other hand, which receive less lighting, for example light silhouettes, remain visible. Additionally to the light intensity value this approach considers the gradient magnitude of a fragment, which is hit by a ray, and the distance of the fragment's position to the eye point (like already discussed in Section 6.2.2) for transparency calculation of the fragment. Resulting images of this technique are shown in Figure 6.17.

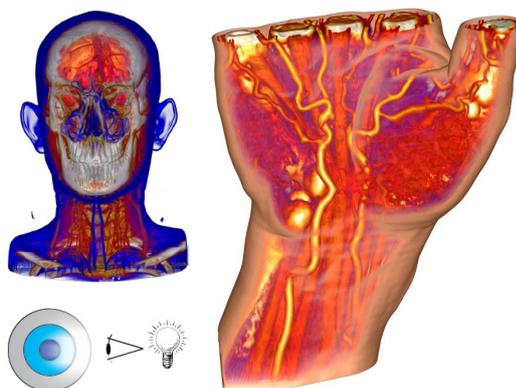


Figure 6.17: Context-preserving volume rendering using the calculation of light reflection for transparency modulation [9].

6.4.3 Exploded Views and Context Deformations

Instead of changing the transparency of a context layer, which occludes the underlying focus region, exploded views and deformations modify the spatial arrangement of the context fragments in order to unveil the view onto the focus. This technique is also adapted from hand-made illustrations or assembly instructions (see Figure 4.1 (d)). Exploded views enable a clear view on the focus by pushing aside parts of the context. The information about the original spatial location thereby can be conveyed by helpers such as lines or arrows. Agrawala et al. [1] proposed design principles for creating effective assembly instructions based on exploded views and presented a system for the automatic design of assembly instructions and a system that semi-automatically generates exploded views from two-dimensional images [58]. Some methods in visualization of volumetric data are based on the model of such assembly instructions and illustrations.

Volume splitting [43, 32, 10] is a visualization technique of scientific data, which enables an exploded view of the data. This technique is intended for displaying multiple enclosed iso-surfaces within the volumetric data. Each iso-surface, except the inner focus surface, is split into two or several parts and moved slightly apart (see Figure 6.18 (a)).

Another visualization technique enables browsing within the data by applying deformations like leafing, peeling, or spreading. McGuffin et al. [60] propose an elaborate

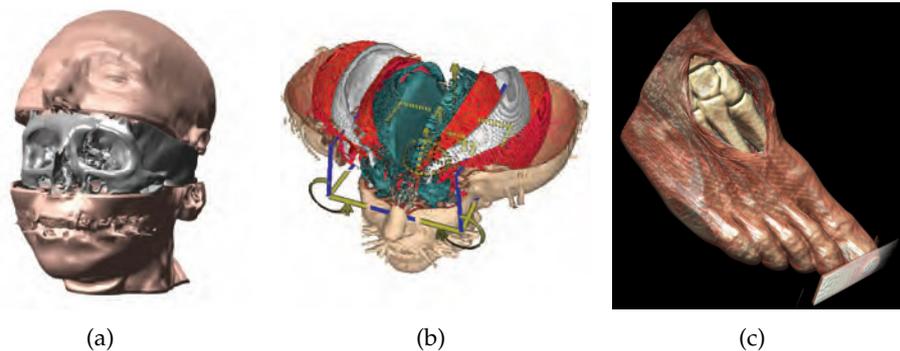


Figure 6.18: Examples of exploded Views and context deformations: volumetric splitting [43] (a) , browsing in the features through leaflet deformation [60] (b), skin distortion of CT foot dataset [15] (c).

framework featuring a set of advanced deformations for an understandable visual presentation of complex three-dimensional information. The operation for investigating the interior of a volume is browsing. The browsing is realized on pre-segmented data decomposed into several semantic layers (e.g., skin, muscle, skull, brain). The user can cut into and open up, spread apart, or peel away parts of the volume in real time. This makes the interior visible while still retaining surrounding context. This technique is illustrated in Figure 6.18 (b).

Viewpoint-dependent distortion of volumetric data [73, 15] is another technique, which applies distortions of occluding fragments in order to have a clear view onto the focus. This technique highlights data by dedicating more display space to it. All context parts originally occluding the focus node are moved apart to uncover the most relevant information as shown in Figure 6.18 (c).

6.5 Designing Focus and Context Visualization for medical AR

Focus and Context Visualization for medical AR systems slightly differs from purely virtual focus and context visualization, as discussed in the previous sections. It has to combine focus and context methods of computer graphics with the two-dimensional image of the real scene. The camera image has to be considered as an additional context layer. Consequently, the following layers exist or can be extracted in medical AR applications:

- a “real” context layer (the camera image of the patient’s body)
- zero or several virtual context layers/objects
- a virtual focus layer/object

The challenge in medical AR is to integrate the "real" context as if it was a virtual layer. Real and virtual objects have to be fused to one image, in which the virtual objects are perceived at the locations where their real counterparts are positioned.

As mentioned in Chapter 5 the perception of object positions, which means layout and distances, has to be as exact as possible. The most important issue is to solve the misleading depth perception when superimposing the virtual objects onto the camera image. This can be done by cutting a window into the camera image (see Section 5.2.2) and thus realising a cut-away view (see Section 6.4.1). This window can also be displayed semi-transparent, which means the application of ghosting (see Section 6.4.2). Thereby features can be highlighted by means of a virtual model of the skin, which was implemented in the focus and context framework for medical AR, described in the following chapter. Of course the techniques of cut-away and ghosting of the context, which was discussed in this chapter, can also be applied to virtual context layers. The possibility for the application of exploded views and deformations of context layers is only limited in medical AR systems, since cues for perceiving distances become lost in such methods. Exploded views and deformations were thus not implemented in the framework presented in the next chapter.

The problem of misleading depth perception was already identified by Bajura et al., in one of the first publications [4] about medical AR. To handle the problem, they rendered a "synthetic hole [...] around ultrasound images in an attempt to avoid conflicting visual cues" and thus created one of the first focus and context visualizations for medical AR. Their method provides visual cues to perceive anatomical objects inside the patient, however their visualizations utilize highlighted, abstract objects, which cannot be smoothly integrated into the scene to create an intuitive view on the operation site. The focus and context visualization framework presented in the following chapter overcomes this problem and offers a smooth fusion of the real and virtual parts of the scene. Furthermore, it considers the cues for the perception of layout and distances presented in Chapter 5, and tries to artificially provide depth cues where they are needed.

7 Results: A Focus and Context Visualization Framework for Medical Augmented Reality

This chapter presents the results of the practical work of this diploma thesis. It explains new techniques, which were developed to generate a focus and context visualization framework for medical augmented reality. These techniques are based on the model of methods described in Chapter 6, whereas chief attention lies on the correct fusion of the two-dimensional camera images of the real part of the scene with the virtual scene. The present framework is capable of exploiting positional and depth cues efficiently to provide an intuitive perception of layout (arrangement), distances and shape of objects within a medical AR scene. Therefore cues are intensified or artificially provided, where they are needed. The framework was implemented and tested by means of the medical AR system of CAM-PAR, described in Chapter 3. It was evaluated by two studies - a cadaver and an in-vivo study - and by generating visualizations of a thorax phantom (see Section 7.2).

In Section 7.3 a ghosting technique is described, which adjusts the transparency of the video image, recorded by the color cameras of the HMD, according to the shape of the patient's skin. The modified video image is blended with previously rendered virtual anatomy. This technique enables a correct fusion of real and virtual anatomy by overcoming the problem of misleading depth perception in medical AR and the provision of important context information. Section 7.4 presents a cut-away technique, which applies volume clipping for cutting away parts of objects, which occlude the focus region. Clipping volumes are able to clip as well parts of virtual objects as parts of the camera image. It is thus able to cut a virtual hole into the body of the patient. Section 7.6 describes the implementation of a shadow mapping algorithm in order to provide depth cues enabled by shadow casts.

The present work also aims on a smooth integration of surgical instruments. In surgeries, especially keyhole surgeries, it can be very helpful, when the instruments can always be seen inside the body. Therefore the instrument has to be extended virtually as soon as it penetrates into the patient's body. Furthermore, instruments can be exploited to provide visual feedback and thus additional positional cues for the virtual scene. Visual feedback includes particularly the casting of shadows onto other virtual objects. A further method is the use of light surrounding the instrument axis, enabling a spotlight in the direction of the axis. The implementation of a method which enables the virtual extension of instruments and the use of shadows and the instrument light are discussed in Section 7.7. Moreover, this section presents a method for highlighting the penetration port of an instrument. The first section of this chapter discusses rendering and shading of virtual anatomy.

7.1 Rendering and Shading of the Virtual Anatomy

All techniques, described in this chapter, were used and tested only in conjunction with indirect volume rendering (see Section 6.1.1). That means that all figures of this chapter, illustrating the presented techniques for focus and context visualization, show surface models, which were segmented, triangulated and smoothed before the visualization. However, all techniques described in this chapter can also be realized using direct volume rendering (see Section 6.1.2). Due to time and performance constraints, it was not possible to integrate direct volume rendering into the framework yet. Polygonal surface models have the advantage of allowing real time rendering of volumetric data. Using a direct rendering technique requires powerful hardware to realize real time visualization. However, future work will include the integration of the framework into a ray cast based volume renderer designed for comparable medical AR scenarios (see Section 6.1.2).

The framework is implemented in C++ using the OpenGL¹. All shader programs were realized in CG².

For illumination and coloring of the datasets the Blinn-Phong model is applied (see Section 6.3.1). The diffuse and specular fraction can be adjusted by means of the data set. Tone-based shading (see Section 6.3.2) was not implemented, because the application of shadows (see Section 7.6) would then not be as effective and realistic as with the Blinn-Phong model. Additional contour emphasis (see Section 6.3.3) and curvature-based rendering (see Section 6.3.4) was not implemented due to time constraints, but would be definitely effective for a better perception of the shape of objects.

7.2 Anatomy and Studies

All figures of this chapter, which illustrate the presented techniques for focus and context visualization, show either visualizations of a phantom model, resulting images of a cadaver study or resulting images of an in-vivo study. The two evaluation studies and the usage of the phantom model are discussed in the following.

7.2.1 Phantom

The thorax phantom (see Figure 3.3) originally consisted only of a spinal column installed inside the phantom. However, the phantom's virtual anatomy was extended by surface models segmented from the Visible Korean Human (VKH)³ data set. The virtual models were manually registered to the thorax phantom. The skin model of the thorax phantom

¹Open Graphics Library - <http://www.OpenGL.org>

²C for Graphics: a high-level shading language created by NVIDIA for programming vertex and pixel shaders - http://developer.nvidia.com/page/cg_main.html

³The Visible Korean Human project provides different full body data sets: CT, MRI and anatomical images <http://vkh3.kisti.re.kr/new>

was generated from a CT-scan of the phantom. For tracking nine fiducial markers are attached to the phantom.

7.2.2 Cadaver Study

Within the scope of a study, a cadaver was provided in order to test the developed techniques. The cadaver was scanned with a CT scanner. The resulting volumetric data set includes imaging data starting from the pelvis and ending at cervical vertebrae. Before data acquisition, CT markers were attached to the skin around the region of the spinal column to automatically register the visualization to the body. Figure 7.1 shows the cadaver and the observer wearing a HMD.



Figure 7.1: Cadaver Study

7.2.3 In-Vivo Study

Additionally an in-vivo study was accomplished. Susanne⁴ had a ski accident and got a CT scan of parts of her head. She is fine again and agreed to participate in the study.

The face structure provides high curvature due to nose, mouth and eyes and therefore is perfect to showcase the effectiveness of the present visualization method, which was important to test the technique of adjusting the transparency of the video image (see Section 7.3). At the day of the accident, no CT markers were attached before the CT scan.

⁴name changed

Therefore, visualized anatomy was registered manually with her head using natural landmarks. Figure 7.2 shows Susanne together with an observer and a tracked spoon in her mouth, which was used in this study to simulate the augmentation of a surgical instrument (see Section 7.7).



Figure 7.2: In-Vivo Study

7.3 Ghosting of the Camera Image: Fusion of Real and Virtual Anatomy

This section describes a new ghosting technique (see Section 6.4.2) of the camera image as the "real" context layer (see Section 6.5) of the medical AR scene (see Figures 7.3). The transparency of the camera image is manipulated as a function of the geometry of the skin and the view direction of the observer (surgeon). For determination of the degree of transparency as well the local curvature as contour information are considered. The technique is capable of solving the problem of misleading depth perception in medical AR by providing partial occlusion of the virtual anatomy (compare with Section 5.2.2). Furthermore it supplies positional cues for the virtual anatomy located underneath the real skin (compare with Section 5.3). By using this approach, the anatomy of interest becomes totally visible, except at positions where partial occlusion by the skin results in better perception of the focussed anatomy. The visualization method divides the skin surface into two domains:

- transparent and semi-transparent skin within the vision channel - *TransDom*



(a)



(b)

Figure 7.3: Ghosting of the "real" camera image: (a) in-vivo study and (b) cadaver study.

- opaque skin outside the vision channel - *OpaDom*

The vision channel to the inside of the patient can be interactively defined by the view direction of the observer wearing the HMD. Within *TransDom* only parts of the skin providing important contextual information are shown. Visualized anatomy within *OpaDom* are completely occluded by the skin. Selection of the remaining fragments on the skin is performed by three parameters (described in sections 7.3.1, 7.3.2, 7.3.3) calculated from geometrical properties of the skin surface and the view direction of the observer to provide visual cues for correct perception of depth and spatial layout.

The transparency value of each single pixel of the video image is determined by off-screen-rendering of a skin model of the human body. This transparency calculation is done by a fragment shader program exploring the geometrical properties of the skin. The

computed transparency values are stored to a buffer, a designated texture called transparency map. This map is used to blend the video image with the previously rendered virtual entities of the scene, which is performed by alpha blending. The alpha values of the video image are determined by lookup tables, respectively the generated transparency maps. Therefore a second shader program is used. Three different parameters are taken into account to calculate the transparency (respectively opacity) of a pixel in the video image (see Figure 7.4 in the context of the in-vivo study):

1. The curvature within the region around a fragment's position on the skin surface (see Section 7.3.1),
2. the angle between the fragments normal and the view vector (the vector from the fragment's position to the eyes, respectively the cameras of the HMD), which offers contour information (see Section 7.3.2),
3. and the distance of a fragment's position on the skin surface to the intersection point of the view direction with the skin surface (see Section 7.3.3).

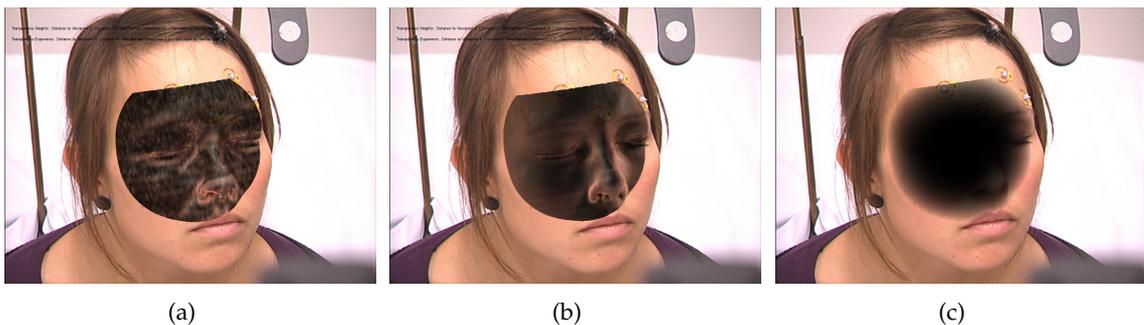


Figure 7.4: Figures show the three parameters, (a) *curvature*, (b) *angle of incidence factor*, (c) *distance falloff*, without any virtual objects in the black background. According to the particular parameter, different regions of the skin on the video images remain.

Finally these three parameters are combined to calculate the transparency of pixels of the camera image. The following sections describe the method in detail.

7.3.1 Curvature

The first parameter *curvature* (see Figure 7.4 (a)) refers to the curvature around a particular 3D position of a fragment on the surface model of the skin (see Section 6.4.2). Two different methods for determining the curvature of a fragment were developed:

- The first method calculates the curvature once for every vertex of the skin model, when the skin model is loaded into memory. Therefore the normals of the neighboring vertices are taken into account. During the rasterization process in the rendering pipeline, the curvature is interpolated for every fragment.
- The second method generates a normal map of the skin from the eye's point of view in every rendering step. This normal map is used to calculate the curvature of a fragment in every rendering step by means of the normals of the neighboring fragments.

The first method is faster than the second one, because the curvature calculation has to be done only once. However, the advantage of the second method is a smoother looking result and the ability of using it in conjunction with direct volume rendering methods like volume ray casting (see section 6.1.2), since these methods do not use vertex models. Consequently, the second method is chosen as the default one.

The functions for calculating the curvature of these two methods are equal, except that the first method considers the normals of the neighboring vertices and the second method the normals of the neighboring fragments. The curvature value for a vertex (respectively fragment in the second method) is calculated by comparing the normal vector \vec{n} of the vertex (respectively fragment) with the normal vectors \vec{n}_i of all neighboring vertices (respectively fragments). N represents the set of normals of the neighboring vertices (respectively fragment).

$$curv = 1 - \left(1 - \frac{\sum_{n_i \in N} \|\vec{n} - \vec{n}_i\|}{2 \cdot |N|} \right)^\alpha \quad (7.1)$$

The sum of length values resulting from $\|\vec{n} - \vec{n}_i\|$ are calculated for every neighboring normal \vec{n}_i , whereas one length value corresponds to the angle between two normals.

Therefore high length values implicate a high curvature value within the region of the vertex position. The division of the sum by two-times $|N|$ clamps the curvature value to the range of $[0, 1]$. $\alpha \in \mathbb{R}^+$ can be modified interactively by the user to adjust the parameter *curvature*. Increasing α approximates the parameters *curvature* to 1, decreasing approximates it to 0. While increasing α , regions with lower curvature contribute more to the opacity of the pixel.

The parameter *curvature* makes those regions of the skin, providing high curvature, become more opaque than regions with low curvature. For this reason wrinkled, bumpy and sinuous regions on the skin can still be seen, while flat parts become transparent. A high weighting of the parameter can be suitable for the visualization within the face regions as anatomy provides high curvature due to nose, eyes and mouth.

7.3.2 Angle of Incidence Factor

The second parameter *angle of incidence factor* (see Figure 7.4 (b)) is defined as a function of the view vector \vec{v} , which points from the fragment's position to the eye (the cameras), and the normal \vec{n} of a point on the skin surface. It provides contour information (see

Sections 6.3.3 and 6.4.2), since it calculates the dot-product of the normal and the view vector.

$$angOfInci = 1 - (\vec{n} \cdot \vec{v})^\beta \quad (7.2)$$

For approximately parallel vectors \vec{n} and \vec{v} , this parameter is close to 0 and results in low opacity. For almost perpendicular vectors \vec{n} and \vec{v} the parameter converges to 1 resulting in high opacity. It thus highlights the contours, because contours are defined as sites, where the direction of the surface normal is orthogonal to the line of sight (see Section 6.3.3), which means sites, where the described dot-product and thus the parameter have the value 1. Similar to the curvature, this parameter can be adjusted interactively by $\beta \in \mathbb{R}^+$.

The parameter *angle of incidence factor* is modified by the observer wearing the HMD and defining the view direction. It has the effect, that faces of the skin which are oriented towards the eyes of the observer (the hmd-cameras) appear more transparent than faces which "look" away from the eyes. In contrast to the curvature it is thus view-dependent.

7.3.3 Distance Falloff

The third parameter *distance falloff* (see Figure 7.4 (c)) is a function of the distance between each surface fragment and the intersection point of the view direction and the skin surface. The parameter can be calculated using any monotonically increasing falloff function. The transparency of skin fragments decreases with respect to the distance to the intersection point. Skin fragments having a greater distance to the intersection point than a chosen maximum distance are rendered completely opaque. Thus this parameter divides the skin in the *TransDom* and *OpaDom*, whereas the transparent region can be considered as the vision channel to the region of interest inside the body. The falloff function leads to a smooth border between the domains *TransDom* and *OpaDom*.

$$distFalloff = saturate \left(\frac{distToViewIntersecPoint}{radiusOfTransparentRegion} \right)^\gamma \quad (7.3)$$

Saturate is a function that clamps its input parameter to $[0, 1]$. The falloff is calculated by a division of the distance to the intersection point of the view direction with the skin surface by the defined radius of the *TransDom*.

The parameter *distance falloff* implements a restriction of the observer's vision channel to the important region of the virtual entities. The adjustable exponent $\gamma \in \mathbb{R}^+$ defines the smoothness of the transition between the domains *TransDom* and *OpaDom*. A smaller value for γ results in a smoother transition and a higher degree of opaqueness close to the intersection point of the view direction with the skin surface. Vice versa, a high value for γ results in a sharp transition of the domains and a low degree of opaqueness close to the intersection point. The position of the vision channel is interactively defined by the head motion of the observer.

7.3.4 Final Transparency/Opacity Determination

In order to calculate the opacity value of a pixel on the video image, the maximum of the three parameters is determined. In addition to the respective exponents α , β and γ , the parameters can be interactively weighted by $w_1, w_2, w_3 \in \mathbb{R}^+$.

$$\text{opacity} = \text{saturate}(\max(w_1 \cdot \text{curv}, w_2 \cdot \text{angOfInci}, w_3 \cdot \text{distFalloff})) \quad (7.4)$$

The resulting opacity value is then stored to the mentioned buffer called transparency map. When rendering the camera image, a second fragment shader uses this map to define the alpha values of the pixels in the camera image. Finally the colors of the previously rendered virtual anatomy is blended with the manipulated video image.

Results of this technique are illustrated in Figures 7.3, whereas expedient combinations of the three transparency parameters were chosen. Figure 7.3 (a) shows its evaluation in the in-vivo study and Figure 7.3 (b) in the cadaver study. The sequence in Figure 7.5 shows different combinations of the three parameters to calculate the transparency map.



Figure 7.5: High "angle of incidence factor" (a). Combination of "angle of incidence factor" with the "curvature" parameter (b). High weighting of the "distance falloff" parameter (c).

7.3.5 Highlighting the Border of the Semi-Transparent Domain

In order to enhance the border between *TransDom* and *OpaDom*, and thus to attract the attention of the user, the border can be drawn with a certain thickness aligned to the skin surface. Therefore the distance of fragments to the intersection point of the view direction and the skin surface (see section 7.3.3) is used. The border is realized by setting the red fraction of colors of those fragments to 1, whose distance to the intersection point is within the range $[(1 - \text{thickness}) \cdot \text{maxdist}, \text{maxdist}]$, where *thickness* denotes a user chosen thickness value for the border and *maxdist* denotes the user chosen radius of *TransDom*. This method is equivalent to intersecting a sphere around the intersection point with the surface. The colored border further attracts the user's attention to the focus region (see Figure 7.6).

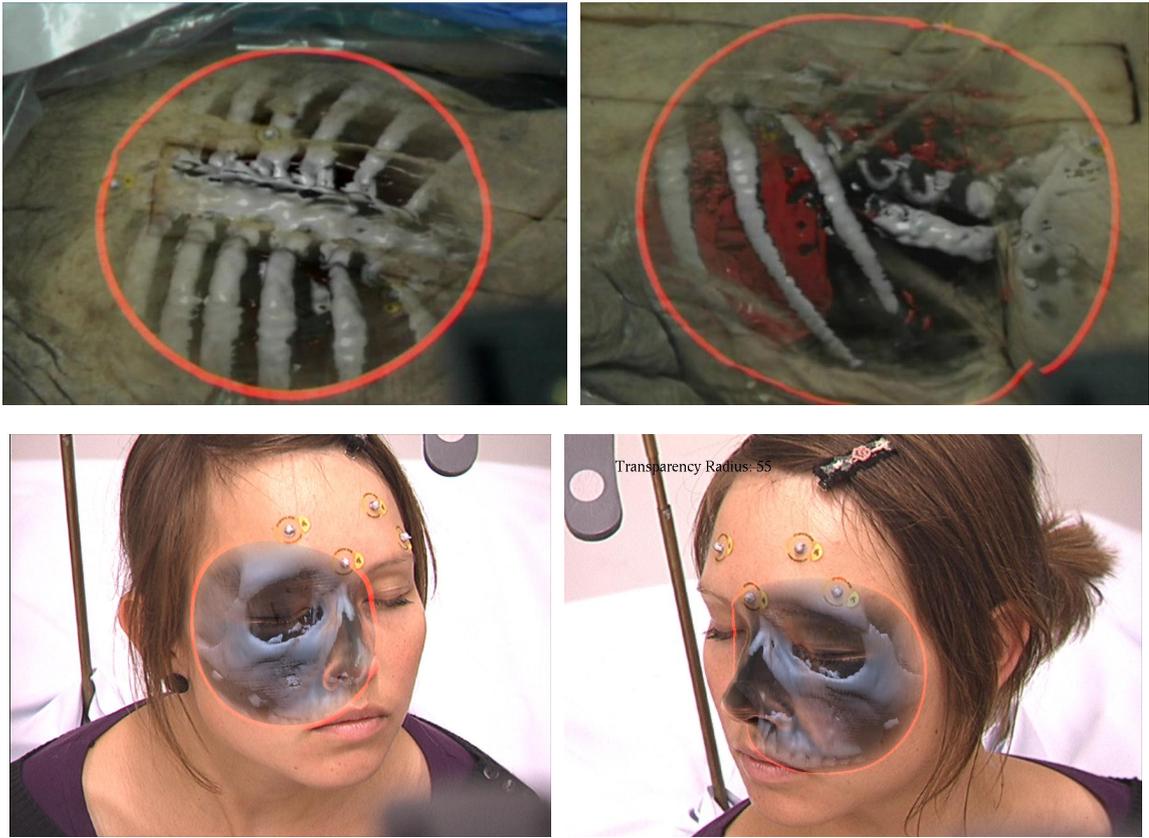


Figure 7.6: Highlighted borderline of the semi-transparent domain.

7.3.6 Virtual Window to the Inside of the Patient

Initially the center of *TransDom* on the skin surface is the intersection point of the view direction with the skin model. The observer wearing the HMD is able to navigate this domain to a region of interest by moving his/her head to get the desired vision channel into the patient. The vision channel can be fixed at a particular region. Enhancing the transition between *OpaDom* and *TransDom* by a colored borderline results in a round window overlaid on the skin surface (see Figure 7.7) providing the depth cue of occlusion (see Section 5.2.2). While moving relative to the window, the observer gets an intuitive view on the anatomy of the patient and depth cues occlusion and motion parallax (see Section 5.2.3) correct depth perception of objects within the field of view.

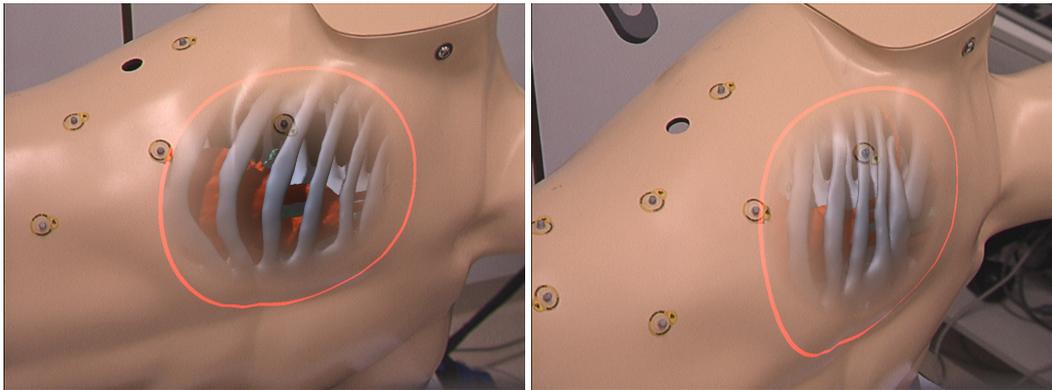


Figure 7.7: Moving around the fixed virtual window enables to exploit the depth cue motion parallax and improves depth perception of objects within the medical AR scene.

7.4 Cut-away by Volume Clipping

Additional to the ghosting of the real skin, a cut-away method (see Section 6.4.1) was developed, which enables the user to clip parts of the virtual anatomy and the camera image. A virtual hole is thereby created in the body of the patient. The clipping method can be applied to objects hindering the view on focus objects. Focus objects are not clipped and thus remain inside the hole. The outside and the borders of the volume provide context information and thus positional and depth cues.

The user is able to define and position a volume, which clips away any parts of context objects lying inside of it. This volume can be a box, a sphere or a cylinder and the user is able to adjust it individually in size and space. The user defines the focus objects of the data set of the human body by determining for every segmented object, if it should be clipped by the clipping volume or not. Theoretically several clipping volumes can be defined to adjust the view on the focus objects even more precisely, but this was not implemented due to time constraints.

The usage of clipping volumes is illustrated in Figures 7.8. Figures 7.8 (a), (b) and (c) show the application of a box as clipping volume and Figure 7.8 (d) the application of a sphere as clipping volume. Clipping volumes are intended to be used in conjunction with the ghosting of the “real” skin (see Section 7.3). Figure 7.8 (a) shows the application of a clipping box without ghosting of the skin, whereas Figures 7.8 (b), (c) and (d) illustrate the use of clipping volumes in addition to ghosting of the skin.

The presented technique generates a depth map by grabbing the depth buffer after off-screen rendering of the skin and the chosen clipping volume. This clipping map is used to clip objects in the scene by performing a depth test in a shader program. The technique is discussed in the following subsections in more detail.

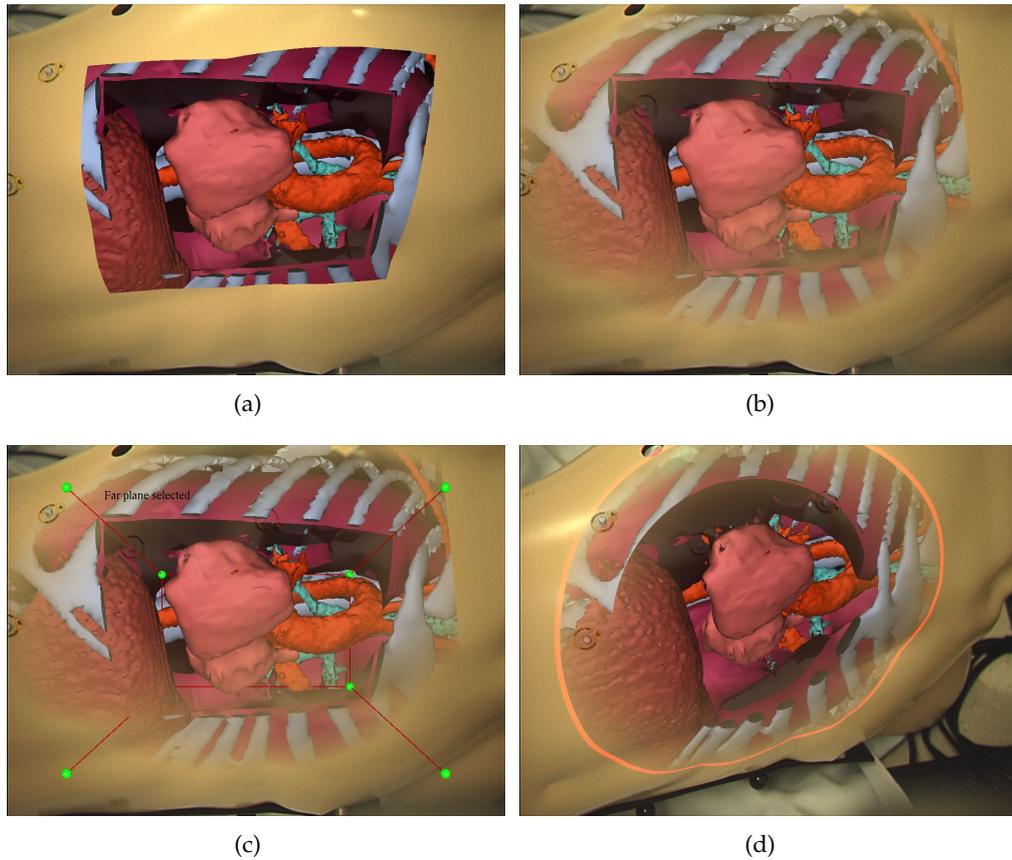


Figure 7.8: Cut-away by Volume Clipping: (a) the usage of a box as clipping volume, (b) the clipping box in conjunction with ghosting of the skin, (c) setting up the clipping volume and (d) the usage of a clipping sphere in conjunction with ghosting of the skin.

7.4.1 Setting up Form, Size and Position of the Clipping Volume

When using a box for clipping, the user is able to interactively define the coordinates of each side of the box by key-pressing, mouse movement or movement of his head (see Figure 7.8 (c)). If a sphere or a cylinder has been chosen, the user can define positioning and radius of the volume. When the user found the right position and size of the volume, it can be fixed to this position. The volume will then remain on the chosen position with respect to the human body.

7.4.2 Generation of the Clipping Depth Map

In order to generate the depth map, first all buffers are cleared and the skin is rendered to the depth buffer. Afterwards the depth test is flipped, which means that fragments pass the depth test, that have a higher depth value than those which are already in the depth buffer. The clipping volume is then rendered to the depth buffer. The depth values of the clipping volume are stored in the depth buffer at those pixel positions, where they are greater than those of the previously rendered skin. Additionally the stencil bit⁵ is set at these screen positions. The resulting stencil mask is used for clipping of the camera image (see Section 7.4.4) and displaying of the volume (see Section 7.4.5). At positions where the new depth values are smaller than those of the skin, the depth values of the skin remain in the buffer. The resulting depth buffer is stored to a depth map.

7.4.3 Clipping of Virtual Objects

A render mode chosen by the user determines if a virtual object should be clipped or not and thus whether it should be a focus object or not. When rendering an object that should be clipped, a fragment shader program is performing an additional depth test to the standard depth test, using the previously created depth map as parameter. This shader performs a lookup into the depth map at the screen coordinates of the fragment and compares the received depth value with the depth value of the fragment. If the depth value is smaller (nearer to the eye), the fragment is discarded. That means that every fragment which has a smaller depth value than the clipping volume is discarded and thus the part of the dataset lying inside the clipping volume is clipped.

7.4.4 Clipping of the Camera Image

The camera image gets clipped only if the clipping volume is not used in conjunction with ghosting of the skin. If applied without ghosting of the skin, the generated stencil mask (see Section 7.4.2) can be utilized to display the camera image only at those screen positions where the stencil bit is not set. In this way, every part of the skin lying inside the clipping volume is clipped and a window on the skin is generated, offering the depth cue of occlusion for the virtual objects (see Section 5.2.2 and Figure 7.8 (a)).

7.4.5 Displaying and Texturing of the Clipping Volume

In order to provide further depth cues, the borders and planes of the clipping volume can also be displayed. Therefore fragments of the clipping volume lying in front of the skin are removed by using the generated stencil mask (see Section 7.4.2). The sides of the clipping volume can be displayed in a semi-transparent or opaque way. Shading and texturing the

⁵The stencil buffer is an additional buffer beside the color buffer and depth buffer, and available on usual modern computer graphics hardware.

clipping volume is able to provide further context information for the focus objects lying inside. Additional displaying of the edges of the volume provides linear perspective (see Section 5.2.2). Figure 7.9 illustrates these effects. Here a semi-transparent texture showing a linear gradient from white (near to the observer) to dark (distant to observer) was chosen to improve depth estimation of the focus.

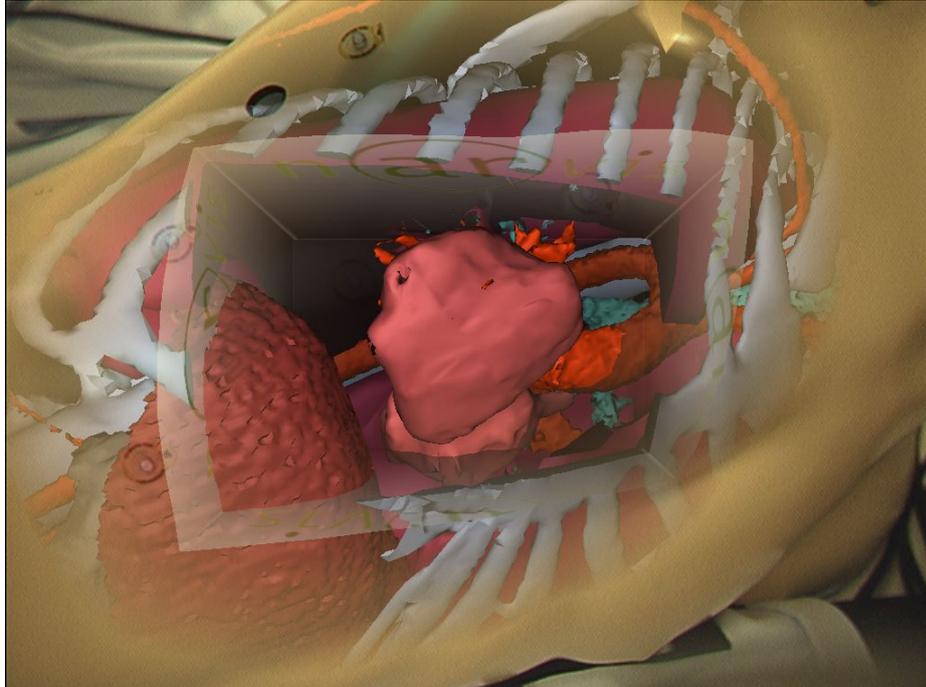


Figure 7.9: Texturing of the clipping volume.

7.4.6 Clipping by a Clipping Depth Map vs. Standard Clipping Planes

The clipping could also be performed by the use of clipping planes, provided by a graphics library like OpenGL. But this approach would have two major disadvantages compared to the shader approach, discussed in this section. The first one is a loss of flexibility in choosing the volume. With the shader approach it is theoretically possible to use any kind of volume for clipping, although for complex volumes it might be necessary to use several depth maps. When using clipping planes, only a simple volume, like a box can be defined for clipping. The second, and bigger, disadvantage concerns the performance. By using OpenGL-clipping planes a clipped data set has to be rendered once per clipping plane. When using a box as clipping volume, this would mean that the data set has to be rendered five times (for every side of the box lying inside the dataset), instead of one time using the shader approach.

7.5 Ghosting of Virtual Anatomy

Alternatively to cut away parts of the virtual anatomy hindering the view onto the focus, virtual entities can be rendered semi-transparently. This can be performed by assigning an uniform transparency value to every fragment of the object, but this would have the disadvantage of a too great disturbance of the view onto the focus (compare with Section 6.4.2). Instead parts with high curvature are rendered more opaque, whereas flat regions are rendered more transparent, like it is performed for adjustment of the transparency of the camera image. The calculation of the curvature and thus the transparency value is performed analogous to Section 7.3.1. However, simultaneous rendering of a transparent skin (camera image) and the transparent virtual anatomy can nonetheless disturb the view onto focus (compare with Section 6.4.2). Consequently, cutting away this virtual anatomy is recommended instead of the application of ghosting, if the skin is already rendered transparent.

7.6 Shadows

In addition to phong shading a shadow mapping algorithm was integrated to provide improved spatial information of virtual objects. Shadows are useful cues for the perception of spatial depth and layout. Their effectiveness was demonstrated in Section 5.2.2. The method of shadow mapping was introduced by Lance Williams in 1978 [92]. It first renders the scene from the position of the light source and stores the resulting depth buffer into a texture, called shadow map. In a second pass the scene is rendered from the usual camera viewpoint. For every generated fragment it is thereby tested, if it is visible from the light source, by projecting it into the shadow map and comparing the received depth value with the depth value of the projected fragment.

The implemented shadow mapping algorithm was presented by Yury Uralsky [83], which improves the original algorithm by approximating real soft shadows at interactive frame rates. It does not produce physically correct soft shadows, but the visual effect is similar to real soft shadows. The method is based on multiple and adaptively sampling of a created shadow map in a fragment shader. It avoids artifacts at the shadow edges, such as rough and angular shadow edges, due to creating a penumbra region at the shadow edges. Instead of only sampling the shadow map at the projected position of the fragment, here several irregular samples of the shadow map are performed in the area around the projected position of the fragment. The samples are then combined to a shadow value, which states the degree of shadowing of the fragment. If this shadow value is equal to 1, the fragment lies fully in the shadowed area, whereas if the value is equal to 0, the fragment does not lie in the shadow. A value between 0 and 1 means, that the fragment is positioned inside the penumbra region. The fragment is darkened by means of its shadow value. Furthermore, the sampling is stopped, if all of a defined number of test samples either lie in the shadow or outside the shadow. These test samples allow the skipping of

unnecessary computation and thus increase the performance.

An algorithmic overview of the implemented shadow mapping algorithm is presented in the following (certain OpenGL-commands are provided in brackets):

1. Creation of the Shadowmap: Rendering the scene from the position of the light source
 - a) Transformation of the "eye" to the position of the light source with view onto the virtual scene (*gluLookAt*)
 - b) Properly adjusting of the field of view (*gluPerspective*)
 - c) Adjusting of the view port to the size of the shadow map (*glViewport*)
 - d) Enabling of the front face culling in order to resolve stitching problems, which means self shadowing of fragments
 - e) Storage of the modelview-projection-matrix
 - f) Rendering of the virtual scene, which means virtual anatomy and models of surgical instruments
 - g) Storage of the depth buffer into a depth map
2. Drawing of the scene from the original camera viewpoint, application of the shadow map
 - a) Vertex shader: transformation of the vertex's model position into the shadow map by means of the stored modelview-projection-matrix, storage of the resulting shadow map position as texture coordinate
 - b) Rasterization: shadow map position gets interpolated for every fragment
 - c) Fragment Shader: usage of the x- and y-coordinate of the shadow map position for performing samples of the shadow map
 - Adaptive and irregular sampling of the area around the projected shadow map position in order to compute a final shadow value
 - Darkening of the fragment's color on the basis of the computed shadow value

Please read the paper of Yury Uralsky [83] for detailed explanation of the algorithm.

Figure 7.10 illustrates the perceptive gain of displaying shadows. Figure 7.10 (a) shows a scene without shadows, while Figure 7.10 (b) shows the same scene with shadows. The shadow cast enables a more realistic perception of the scene and provides helpful visual feedback for navigation of augmented instruments inside the body (see Section 7.7).

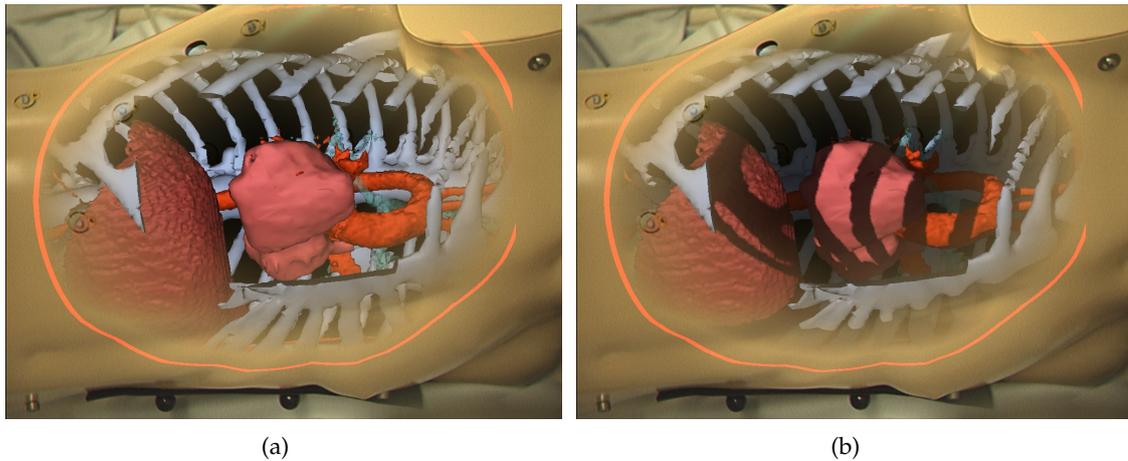


Figure 7.10: (a) Scene without shadows and (b) with shadows.

7.7 Integration of Surgical Instruments

In keyhole surgery, appropriate surgical instruments are inserted into the patient's body through a trocar serving as a port to the operation site. In state-of-the-art systems, video images recorded by an endoscopic camera device are presented on a monitor viewing the operation site inside the patient and involved instruments. However, surgeons have to deal with visual cues from an additional environment presented on the monitor and integrate this spatial information into their mental model of the operation site.

Thanks to an HMD for in-situ visualization in minimal invasive surgery the fields of view can be reduced to only one workspace namely the patient lying on the operation table. Furthermore such a system does not require the mental mapping of medical imaging data with the patient, which is necessary in traditional presentation systems where imagery is shown on a 2D display or attached to the wall. The present system integrates such data three dimensionally into the mentioned field of view. However, this requires new techniques for a smooth integration of such surgical tools into the AR scene to enable the surgeon to perceive relative distances between instruments and the operation site. Therefore, the augmentation of the instrument is presented only inside the patient's body. For that reason an exact virtual copy of an instrument has to be constructed and overlayed onto the real instrument, when it enters the body. A methods for generating the virtual extension of an instrument is presented in Section 7.7.1. Furthermore, visual feedback through shadows casts (see Section 7.7.2), highlighting the penetration port (see Section 7.7.3) and a method for the creation of light surrounding the instrument axis (see Section 7.7.4) are explained.

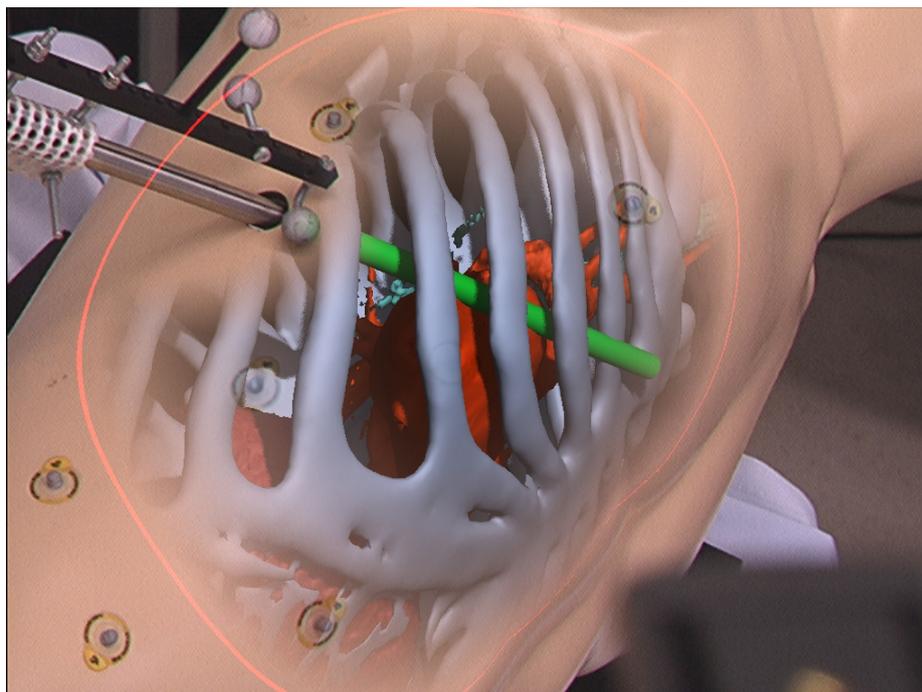


Figure 7.11: The instrument is virtually extended inside the thorax phantom.

7.7.1 Virtual Extension of Instruments

In this framework only the visually hidden parts of an instrument are augmented. This is performed by rendering an exact model of an instrument in two rendering passes after rendering the model of the skin surface. In the first pass the stencil bit is set for those fragments of the instrument that fail the depth test. During the first rendering pass, drawing to the color buffer is disabled. In the second pass, fragments are stored to the color buffer if the stencil bit is set on their respective screen position. The virtual extension of an instrument inserted into the thorax phantom, is shown in Figure 7.11.

7.7.2 Visual Feedback due to Shadow Effects

Thanks to the implementation of the shadow mapping algorithm (see Section 7.6) the surgical instrument throws shadow onto the anatomy and vice versa. When the surgeon guides a surgical instrument close to the visualized anatomy, the shadow can be recognized on the virtual bones and tissue. According to Kersten et al. (see Section 5.2.2), the human visual system is able to estimate the position of the light source and the relative distances and in this case, between the instrument and the bones and tissue. A shadow thrown by the surgical instrument onto the anatomy provides helpful visual feedback due to user interaction, which can be used to navigate the tip of the instrument to the opera-

tion site. Figures 7.12 illustrate this effect within the in-vivo study. A tracked spoon was integrated into the scene to interact with the in-situ visualization. Therefore, an exact virtual counterpart of the spoon was created to visualize it also beneath the skin surface and to throw a shadow on visualized anatomy. The spoon was not inserted into the mouth because the data set did not include this region. Figure 7.13 shows a result of the cadaver



Figure 7.12: Shadow is thrown by a tracked spoon onto the virtual skull.

study. The surgical instrument was not inserted into the body, however the shadow was thrown onto the visualized ribs, caused by the user guided surgical instrument and a virtual light source positioned above the scene.

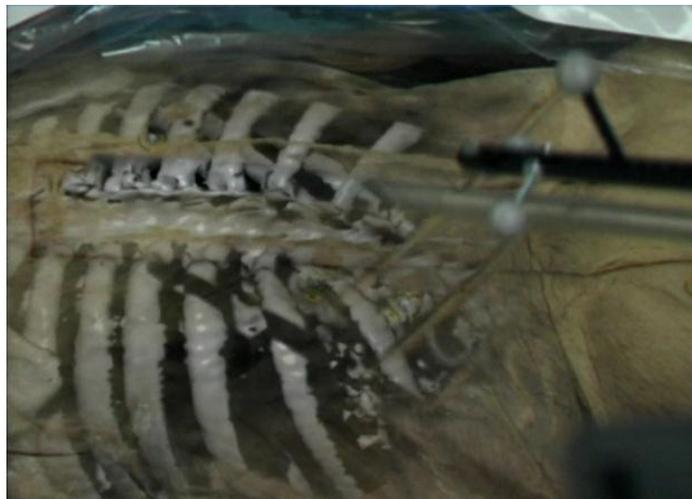


Figure 7.13: Shadow is thrown by a surgical instrument onto the virtual ribs of the cadaver.

7.7.3 Highlighting the Penetration Port of the Instrument

This section discusses the highlighting of the region on the skin around the port, where a surgical instrument is inserted into the body. The port represents a transition between the real and the virtual scene. Beside the virtual extension of the instrument inside the body, as well the real outer part of the instrument as the port should be seen in the resulting AR images. In this section first a method is presented for coloring the port. This method can further be extended in order to calculate the position of the port. The position is necessary to realise a technique enabling a smooth transition between the outer and the inner part of an instrument by adjusting the transparency of the camera image around the port. The region of the skin around the port gets opaque when the instrument is penetrating the body. Thanks to this effect, the view on the port is not restricted, since it is not occluded by the virtual anatomy.

Rendering of the Intersecting Area of Instrument and Skin

One method for highlighting the port is coloring the intersecting area of instrument and skin. It can be realized without knowing the coordinates of the intersection area. The implemented algorithm uses the stencil buffer to mask the intersection area and is similar to the algorithm, which is applied for drawing shadows using shadow volumes⁶. It requires one rendering pass for the skin model and three rendering passes for the instrument model. Rendering of the skin model and the first two passes for the instrument model do not affect the color buffer (they are off-screen passes). These passes are only required to set stencil values. The third pass renders the intersection area to the color buffer. The three rendering passes for the instrument are presented in the following:

1. **First pass** (off-screen):
 - a) Enabling backface culling
 - b) Rendering of the instrument model
 - c) Increasing of stencil value, when depth test passes
2. **Second pass** (off-screen):
 - a) Enabling frontface culling
 - b) Rendering of the instrument model
 - c) Decreasing of stencil value, when depth test passes
3. **Third pass**: Rendering of the instrument model in a specific color (e.g. red) where the stencil bit is set

⁶Besides shadow mapping, one of the most popular algorithms for creating shadows.

Determination of the 3D-Coordinates of the Intersecting Area of Instrument and Skin

Rendering of the intersecting area, presented in the previous section, can be used to determine the 3D-position of the intersecting area, respectively of the port. Therefore a suitable small intersection area is drawn in a specific color on the black screen, applying the algorithm described above. The intersection point has to be as small as possible, which means that it should only contain a few screen pixel. This can be realized by scaling the size of the instrument model before rendering the intersection area. Afterwards the pixels of the screen are read and iterated to find the few pixels, which have been drawn previously. By doing that, the color of each pixel is compared with the color of the intersection area. If the first colored pixel is found, its screen coordinates are read and transformed back to model- or view-space. In detail, the algorithm requires the following steps (the color buffer has to be cleared previously):

1. Rendering the skin to the depth buffer (off-screen)
2. Drawing a scaled, which means suitable small intersection area using the algorithm described in the previous subsection. Thereafter, only a few colored pixel, belonging to the intersection area, are localized on the black screen.
3. Reading all pixels of the screen from the color buffer (*glReadPixels*)
4. Running through the pixels and searching for the first colored pixel
5. Transforming the screen coordinates and the depth value of the found pixel back to model- or view-space (*gluUnProject*)

The algorithm exploits functions of the standard graphics pipeline on the graphics hardware. The found position is not the totally exact midpoint of the intersection area, but is sufficient for the used purpose of adjusting the transparency of the camera image around it (see next subsection). The smaller the drawn intersection area the better results are provided by the algorithm. The scaling amount of the instrument model has to be chosen by means of its size and the distance of the observer to the body. The advantage of this method over a classic collision detection is its low runtime complexity. In this application the skin model has to be rendered anyhow, in order to manipulate the transparency of the skin (see Section 7.3) and thus no further rendering pass is required. The complexity is hence independent of the size of the skin model. It depends only on the size of the instrument model (which is usually very small and thus negligible) and the screen solution.

Modifying the Transparency Settings of the Camera Image around the Port

The determination of the 3D-position of the port (see previous subsection) can be used for modifying the transparency of the camera image around the region of the trocar. Within the video images, the region around the port remains opaque, so that the real instrument entering the patient is not occluded by the visualized anatomy. The transparency of pixels

within this region depends on the distance to the determined intersection point. Therefore a fourth parameter is used for transparency calculation of the skin described in Section 7.3.4, which is decreasing linearly with respect to the distance to the intersection point:

$$instrumentIntersectionFactor = 1 - \frac{distanceToInstrIntersecPoint}{maxdist} \quad (7.5)$$

with $maxdist$ as a constant value representing the maximum distance to the intersection point of fragments, that are influenced by this parameter. The effect of this method is illustrated in Figure 7.11.

7.7.4 Light surrounding the Instrument Axis

A further feature concerning the interaction of instrument and virtual scene is the implementation of a light surrounding the instrument axis up to a distance of several centimeters. If the instrument is sticking in a virtual entity (e.g. in an organ or tissue), the light highlights the surrounding fragments with the color of the light (e.g. green). This highlighting of the surrounding fragments helps to perceive the exact positioning of the instrument, although it is hidden by the virtual object. This method furthermore causes the throw of a spotlight on the anatomy in the direction of the instrument axis. The spotlight provides visual cues to perceive of the orientation of the instrument respective the virtual anatomy. Thus it allows a better handling and navigation of the instrument to the operation site. Figures 7.14 illustrate the effect of light surrounding the instrument axis.

The light is implemented by a fragment shader. For every fragment the distance of the 3D-position of fragment to the instrument axis, which represents a straight line, is calculated. Therefore the perpendicular has to be dropped from the position of fragment to the axis. The distance of the fragment's position and the axis is equal to the distance between the fragment's position and the intersection point of the perpendicular and the axis. Consequently this intersection point \vec{l} is calculated first. It is given by

$$\vec{l} = \vec{x}_0 + \lambda \vec{t}, \quad (7.6)$$

with \vec{x}_0 as arbitrary point on the axis, \vec{t} as tangent vector pointing in the direction of the axis and λ as real number. For \vec{x}_0 the position of the instrument tip is used and \vec{t} can be easily calculated by means of a second point on the axis. λ can be determined by

$$\lambda = \vec{t} \cdot (\vec{p} - \vec{x}_0), \quad (7.7)$$

with \vec{p} as the position vector of the fragment. Finally, the distance can be calculated as the length of the vector $(\vec{l} - \vec{p})$.

For enabling the light, a term which is decreasing linear with respect to the distance d , can be added to the color intensity:

$$c_{out} = c_{in} + 1 - saturate\left(\frac{d}{maxdist}\right), \quad (7.8)$$

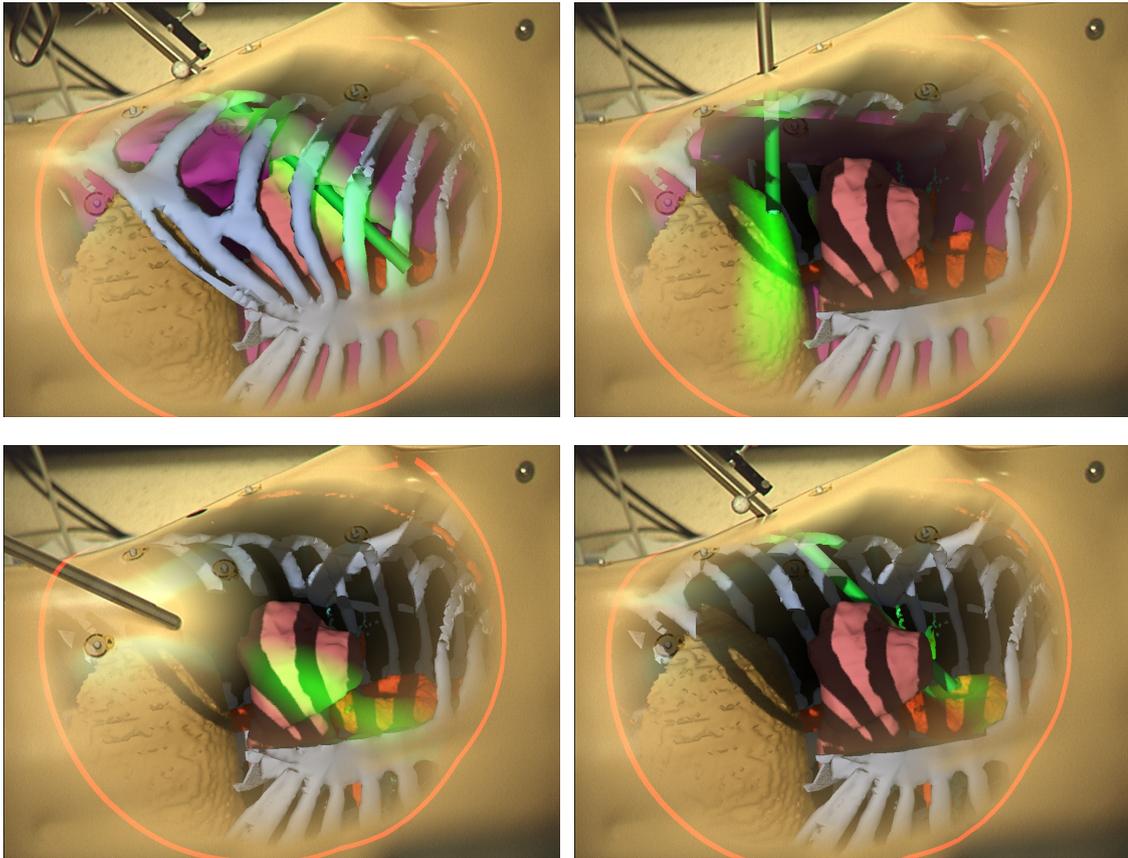


Figure 7.14: Light surrounding the instrument axis.

with *maxdist* as a constant value representing the maximum distance of fragments, which are influenced by the light. *saturate* clamps the value to the range $[0,1]$.

7.8 Visualization Pipeline

This section summarizes and combines the most important rendering steps of all presented techniques of the present focus and context visualization framework. They are enumerated in the correct rendering order:

1. Generation of curvature values for each vertex of the skin model - this has to be done only once (see Section 7.3.1)
2. Determination of the intersection area between skin and instrument - if an instrument is used (see Section 7.7.3)

3. Determination of the intersection point of the view direction with the skin - if the vision channel is not fixed (see Section 7.3.3)
4. Calculation of the transparency map by rendering the skin model - parameters: center point of transparent region, maximum distance to the center point of the transparent region, "falloff distance" weight, "curvature" weight, weight of "angle of incidence factor", "falloff distance" exponent, "curvature" exponent, "angle of incidence factor" exponent, intersection point of the skin and the instrument, thickness of focus border (see Sections 7.3 and 7.7.3)
5. Generation of the clipping depth map by rendering the clipping volume (see Section 7.4.2)
6. Rendering the instrument model and increasing the stencil bit when depth test fails (see Section 7.7.1)
7. Calculation of the shadow map - rendering instrument model and virtual anatomy from the view of the light source (see Section 7.6)
8. Rendering the virtual anatomy:
 - a) Blinn-Phong illumination (see Section 7.1)
 - b) Shadow calculation: projection of the position of the fragment into the shadow map (see Section 7.6)
 - c) Clipping the fragment by means of the clipping depth map (see Section 7.4.3)
 - d) Modifying the transparency of fragments (see Section 7.5)
 - e) Implementation of light surrounding the instrument axis (see Section 7.7.4)
9. Drawing the instrument where stencil bit is set (see Section 7.7.1)
10. Displaying and texturing the clipping volume (see Section 7.4.5)
11. Rendering the camera image with alpha blending, manipulated with the transparency map (see Section 7.3.4)

8 Conclusion and Further Work

This diploma thesis deals with focus and context visualization of medical imaging for medical AR. Chief attention lies on the correct perception of a region of interest inside the in-situ visualized anatomy within the context of video images representing the real part of the scene. Medical AR systems, like the introduced system of CAMPAR (see Chapter 3), are able to realize in-situ visualizations with desired accuracy and speed. However, superimposed visualizations of virtual and real anatomy can still lead to misleading depth perception of the virtual anatomy. This problem can be solved by generating visualizations, where the real body partly occludes the virtual objects, for instance by cutting a window into the real body of the patient. Misleading depth perception in medical AR is discussed Chapter 5 in conjunction with a general analysis of the perception of layout and distances. The provision of a guarantee for correct perception of layout and distances of virtual objects in the AR scene is a major interest in medical AR visualizations. In a possible surgical application of medical AR, incorrect and misleading perceptions could be life-threatening. Chapter 5 moreover concerns the use of transparency for providing contextual information and the perception of shape of objects. Chapter 6 discusses general principles and concrete methods for realising focus and context visualizations, which are mainly adapted from hand-made illustration techniques.

Cognitions of Chapters 5 and 6 were used to implement a focus and context visualization framework, which enables a smooth fusion of real and virtual anatomy (see Chapter 7). Consequentially new methods for medical in-situ visualization are presented that allow for improved perception of medical imaging data and navigated surgical instruments relative to the patient's anatomy. Section 7.3 presents a new technique for modifying the transparency of camera images, recorded by the color cameras mounted on a stereoscopic, video see-through HMD. The transparency modulation of the video images thereby depends on the topology of the skin surface of the patient and the viewing geometry of the observer wearing the HMD. The presented method allows for an intuitive view on the deep-seated anatomy of the patient by providing important context information and thus visual cues to correctly perceive absolute and relative distances of objects within an AR scene. Another result of this thesis represents a technique (see Section 7.4), which uses volume clipping for cutting away fractions of objects occluding the view onto the focus region. The technique is able to clip parts of virtual objects and parts of the camera image lying inside the clipping volume. It thus generates a virtual hole in the body of the patient, providing context information as well as a clear view on focussed objects. Finally methods for integrating surgical tools into the medical AR scene are described (see Section 7.7). An implemented shadow mapping algorithm is thereby used to provide visual feedback due

to shadow effects, which enables additional depth cues resulting in improved navigation of the instrument. Furthermore, surgical instruments are virtually extended as soon as they penetrate into the patient's body and the penetration port is highlighted. The effectiveness of the developed techniques has been demonstrated with a cadaver study and a thorax phantom, both visualizing the anatomical region of the upper part of the body, and an in-vivo study visualizing the head.

The paper "**Contextual Anatomic Mimesis: Hybrid In-Situ Visualization Method for Improving Multi-Sensory Depth Perception in Medical Augmented Reality**" (authors: C. Bichlmeier, F. Wimmer, S.M. Heining and N. Navab) concerning the ghosting of the real skin and the integration of surgical tools presented in this thesis was submitted to the conference *International Symposium on Mixed and Augmented Reality (ISMAR) 2007, Nara, Japan, Nov 13-16, 2007*.

Further work has to concentrate on as well improvements of the developed techniques, including a better usability, as the realisation and integration of advanced rendering methods for anatomical data, e.g. the application of volume ray casting (see Section 6.1.2). A ray cast based volume renderer was already developed at CAMPAR for comparable medical AR scenarios. However, such a method requires powerful hardware to realize real time rendering. Another issue of future work might be an improvement of shape perception of virtual objects by applying advanced shading methods as discussed in Section 6.3.

9 Acknowledgements

I would like to thank my supervisor Christoph Bichlmeier, my professor Prof. Dr. Nassir Navab, our surgeon Dr. Sandro Heining and all members of the NARVIS crew for their support. I want to express my gratitude to Mohammad Rustae for generating surface models, Hermann Gruber for reviewing this work and Susanne Stettner for taking her time for the in-vivo study. Thanks also to our clinical partner and SCR for the courtesy of their state of the art AR system.

Bibliography

- [1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 828–837, New York, NY, USA, 2003. ACM Press.
- [2] T. Alt and S. Nölle. Augmented reality in der automobilindustrie. *Magazin für professionelle Datenverarbeitung*, 2001.
- [3] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 1997.
- [4] M. Bajura, H. Fuchs, and R. Ohbuchi. Merging virtual objects with the real world: seeing ultrasound imagery within the patient. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 203–210, New York, NY, USA, 1992. ACM Press.
- [5] C. Bichlmeier and N. Navab. Virtual window for improved depth perception in medical ar. In *AMIARCS - Virtual Window for Improved Depth Perception in Medical AR*, Copenhagen, Denmark, Oct. 2006. MICCAI Society.
- [6] I. Biederman and G. Ju. Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64, 1988.
- [7] J.F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 192–198, New York, NY, USA, 1977. ACM Press.
- [8] M. Braunstein, D. Hoffman, and A. Saidpour. Parts of visual objects: an experimental test of the minima rule. *Perception*, 18:817–826, 1989.
- [9] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 11 2006.
- [10] S. Bruckner and M.E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 9 2006.
- [11] B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94: Proceedings of the 1994 symposium on Volume visualization*, pages 91–98, New York, NY, USA, 1994. ACM Press.

- [12] MediaWiki: Volume Ray Casting. http://en.wikipedia.org/wiki/volume_ray_casting, 2007.
- [13] C. Christou, J.J. Koenderink, and A.J. van Doorn. Surface gradients, contours and the perception of surface attitude in images of complex scenes. *Perception*, pages 25:701–713, 1996.
- [14] H.E. Cline, W.E. Lorensen, S. Ludke, C.R. Crawford, and B.C. Teeter. Two algorithms for the reconstruction of surfaces from tomograms. *Medical Physics*, pages 320–327, 1988.
- [15] C. Correa, M. Chen, and D. Silver. Feature aligned volume manipulation for illustration and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1069–1076, 2006.
- [16] B. Csebfalvi, L. Mroz, H. Hauser, A.K., and M.E. Gröller. Fast visualization of object contours by non-photorealistic volume rendering. Technical report, 2001.
- [17] J.E. Cutting and P.M. Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In W. Epstein & S. Rogers (Eds.), *Perception of Space and Motion*, pages 69–117, 1995.
- [18] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2003.
- [19] J. Diepstraten, D. Weiskopf, and T. Ertl. Transparency in interactive technical illustrations. *Computer Graphics Forum*, 21(3):317–326, 2002.
- [20] J. Diepstraten, D. Weiskopf, and T. Ertl. Interactive cutaway illustrations. In *Proceedings of EUROGRAPHICS '03*, pages 523–532, 2003.
- [21] W. Epstein. Nonrelational judgments of size and distance. *The American Journal of Psychology*, 78(1), 1965.
- [22] MediaWiki: Eye. <http://en.wikipedia.org/wiki/eye>, 2007.
- [23] S. Feiner and D.D. Seligmann. Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations. *The Visual Computer*, 8(5&6):292–302, 1992.
- [24] J. Freund, C. Geiger, M. Grafe, and B. Kleinjohann. The augmented reality personal digital assistant. In *Second International Symposium on Mixed Reality (ISMR2000)*, 2001.
- [25] A. Van Gelder and K. Kim. Direct volume rendering with shading via three-dimensional textures. In *VVS '96: Proceedings of the 1996 symposium on Volume visualization*, pages 23–ff., Piscataway, NJ, USA, 1996. IEEE Press.

- [26] W. Gerbino, C.J. Stultiens, and J.M. Troost. Transparent layer constancy. *Journal of Experimental Psychology: Human Perception and Performance*, 16:3–20, 1990.
- [27] J.J. Gibson. *The perception of the visual world*. Houghton Mifflin, 1950.
- [28] E.B. Goldstein. *Sensation and Perception*. Wadsworth Publishing Company, 2002.
- [29] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 447–452, New York, NY, USA, 1998. ACM Press.
- [30] B. Gooch and A. Gooch. *Non-Photorealistic Rendering*. AK Peters, Ltd., 2001.
- [31] B. Gooch, P. J. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. Interactive technical illustration. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 31–38, New York, NY, USA, 1999. ACM Press.
- [32] S. Grimm, S. Bruckner, A. Kanitsar, and M.E. Gröller. Flexible direct multi-volume rendering in interactive scenes. In *Vision, Modeling, and Visualization (VMV)*, pages 386–379, October 2004.
- [33] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 40, Washington, DC, USA, 2003. IEEE Computer Society.
- [34] M. Hadwiger, C. Sigg, H. Scharsach, K. Buehler, and M. Gross. Realtime ray-casting and advanced shading of discrete isosurfaces. In *EUROGRAPHICS 2005*, 2005.
- [35] H. Hauser, L. Mroz, G.I. Bisch, and E. Groeller. Two-level volume rendering - fusing mip and dvr. In *Proceedings of IEEE Visualization 2000 (VIS'00)*, pages 211–218, 2000.
- [36] G.T. Herman and H.K. Liu. Three-dimensional display of human organs from computed tomograms. *Computer Graphics and Image Processing*, pages 1–21, 1979.
- [37] E. Hodges. *The Guild Handbook of Scientific Illustration*. Wiley, 2003.
- [38] W. A. Hoff and T. L. Vincent. Analysis of head pose accuracy in augmented reality. *IEEE Trans. Visualization and Computer Graphics*, 6, 2000.
- [39] K. Hulsey. Technical illustration web page, 2005.
- [40] V. Interrante. Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 109–116, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

- [41] V. Interrante, H. Fuchs, and S. Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 52, Washington, DC, USA, 1995. IEEE Computer Society.
- [42] V. Interrante, H. Fuchs, and S. Pizer. Illustrating transparent surfaces with curvature-directed strokes. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 211–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.
- [43] S. Islam, S. Dipankar, D. Silver, and M. Chen. Spatial and temporal splitting of scalar fields in volume graphics. In *VV '04: Proceedings of the 2004 IEEE Symposium on Volume Visualization and Graphics (VV'04)*, pages 87–94, Washington, DC, USA, 2004. IEEE Computer Society.
- [44] G.A. Kaplan. Kinetic disruption of optical texture : The perception of depth at an edge. *Perception and Psychophysics*, 6:193–198, 1968.
- [45] G. Kebeck. *Wahrnehmung: Theorien, Methoden und Forschungsergebnisse der Wahrnehmungspsychologie*. Juventa Verlag GmbH, 1994.
- [46] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM Journal of Research and Development*, 19(1):2–11, 1975.
- [47] D. Kersten, P. Mamassian, and D. Knill. Moving cast shadows and the perception of relative depth. Technical report, 1994.
- [48] D. Kersten, P. Mamassian, and D.C. Knill. Moving cast shadows induce apparent motion in depth. *Perception*, 26(2):171–192, 1997.
- [49] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 67, Washington, DC, USA, 2003. IEEE Computer Society.
- [50] G. Klinker, D. Stricker, and D. Reiners. Augmented reality for exterior construction applications. *Fundamentals of Wearable Computers and Augmented Reality*, 1(1):379–427, 2001.
- [51] J. Koenderink. What does the occluding contour tell us about solid shape? pages 321–330, 1984.
- [52] O. Konrad-Verse, B. Preim, and A. Littmann. Virtual resection with a deformable cutting plane. *Simulation und Visualisierung*, pages 203–214, 2004.
- [53] J. Krueger, J. Schneider, and R. Westermann. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, 2006.

-
- [54] J. Krueger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 38, Washington, DC, USA, 2003. IEEE Computer Society.
- [55] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 451–458, New York, NY, USA, 1994. ACM Press.
- [56] M. Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37, 1988.
- [57] M. Levoy, H. Fuchs, S.M. Pizer, J. Rosenman, E.L. Chaney, G.W. Sherouse, V. Interrante, and J. Kiel. Volume rendering in radiation treatment planning. In *Proc. First Conference on Visualization in Biomedical Computing*, pages 4–10. IEEE Computer Society Press, 1990.
- [58] W. Li, M. Agrawala, and D. Salesin. Interactive image-based exploded view diagrams. In *Graphics Interface'04*, 2004.
- [59] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Computer Graphics (Proceedings of SIGGRAPH 87)*, volume 21, pages 163–169, July 1987.
- [60] M.J. McGuffin, L. Tancu, and R. Balakrishnan. Using deformations for browsing volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 53, Washington, DC, USA, 2003. IEEE Computer Society.
- [61] M. Meißner, U. Hoffmann, and W. Straßer. Enabling classification and shading for 3d texture mapping based volume rendering. In *VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, Washington, DC, USA, 1999. IEEE Computer Society.
- [62] F. Metelli. The perception of transparency. *Scientific American*, 230(4):90–98, 1974.
- [63] P. Milgram and H. Colquhoun. A taxonomy of real and virtual world display integration. *Mixed Reality - Merging Real and Virtual Worlds*.
- [64] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. In *IEICE Trans. Information Systems*, pages 1321–1329, 1994.
- [65] S. Owada, F. Nielsen, M. Okabe, and T. Igarashi. Volumetric illustration: Designing 3d models with internal textures. *Proceedings of ACM SIGGRAPH(SIGGRAPH2004)*, pages 322–328, 2004.
- [66] B. Pflesser, A. Petersik, U. Tiede, K. Hohne, and R. Leuwer. Volume cutting for virtual petrous bone surgery, 2002.

- [67] B.T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.
- [68] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume on standard pc graphics hardware using multi-textures and multi-stage rasterization. In *HWWS '00: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 109–118, New York, NY, USA, 2000. ACM Press.
- [69] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 231–238, Aire-la-Ville, Switzerland, 2003. Eurographics Association.
- [70] F. Sauer, A. Khamene, B. Bascle, S. Vogt, and G.J. Rubinob. Augmented reality visualization in imri operating room: System description and pre-clinical testing. In *Proceedings of SPIE, Medical Imaging*, volume 4681, pages 446–454, 2002.
- [71] F. Sauer, U.J. Schoepf, A. Khamene, S. Vogt, Marco Das, and Stuart G. Silverman. Augmented reality system for ct-guided interventions: System description and initial phantom trials. In *Medical Imaging: Visualization, Image-Guided Procedures, and Display*, 2003.
- [72] F. Sauer, F. Wenzel, S. Vogt, Y. Tao, Y. Genc, and A. Bani-Hashemi. Augmented workspace: designing an ar testbed. In *IEEE and ACM International Symposium on Augmented Reality*, pages 47–53, 2000.
- [73] M. Sheelagh, T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, page 46, Washington, DC, USA, 1996. IEEE Computer Society.
- [74] P. Shirley and A. Tuchman. A polygonal approximation to direct scalar volume rendering. In *VVS '90: Proceedings of the 1990 workshop on Volume visualization*, pages 63–70, New York, NY, USA, 1990. ACM Press.
- [75] T. Sielhorst, C. Bichlmeier, S. Heining, and N. Navab. Depth perception a major issue in medical ar: Evaluation study by twenty surgeons. In *Proceedings of MICCAI 2006, LNCS*, pages 364–372, Copenhagen, Denmark, Oct. 2006. MICCAI Society, Springer.
- [76] M.J. Sinai, T.L. Ooi, and Z.J. He. Terrain influences the accurate judgement of distance. *Nature*, 1998.
- [77] M. Straka, M. Cervenansky, A. La Cruz, A. Koechl, M. Sramek, M.E. Groeller, and D. Fleischmann. The vesselglyph: Focus & context visualization in ct-angiography. In *IEEE*, editor, *IEEE Visualization 2004*, pages 392–385. IEEE, oct 2004.

-
- [78] I.E. Sutherland. A head-mounted three dimensional display. In *AFIPS Fall Joint Computer Conference*, 1968.
- [79] B. Thomas, B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris, and W. Piekarski. Arquake: An outdoor/indoor augmented reality first person application. In *Fourth International Symposium on Wearable Computers*, 2000.
- [80] J.T. Todd, L. Thaler, and T.M.H. Dijkstra. The effects of field of view on the perception of 3d slant from texture. *Vision Research*, 45:1501–1517, 2005.
- [81] A. Tsiaras and B. Werth. *The Architecture and Design of Man and Woman: The Marvel of the Human Body, Revealed*. Doubleday, firsts edition, 2004.
- [82] J.K. Udupa and G.T. Herman, editors. *3D imaging in medicine*. CRC Press, Inc., Boca Raton, FL, USA, 1991.
- [83] Y. Uralsky. Efficient soft-edged shadows using pixel shader branching. In *GPU Gems 2*. Ed. NVIDIA Corporation, 2005.
- [84] J.J. van Wijk. Image based flow visualization for curved surfaces. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 17, Washington, DC, USA, 2003. IEEE Computer Society.
- [85] I. Viola, E. Gröller, M. Hadwiger, K. Böhler, B. Preim, M.C. Sousa, D.S. Ebert, and D. Stredney. Ieee visualization tutorial on illustrative visualization, 2005.
- [86] I. Viola, A. Kanitsar, and M.E. Gröller. Importance-driven volume rendering. In *Proceedings of IEEE Visualization'04*, pages 139–145, 2004.
- [87] B. Wallisch, W. Meyer, A. Kanitsar, and M.E. Gröller. Information highlighting by color dependent depth perception with chromo-stereoscopy. Technical report, 2002.
- [88] S.W. Wang and A.E. Kaufman. Volume sculpting. In *SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff., New York, NY, USA, 1995. ACM Press.
- [89] D. Weiskopf, K. Engel, and T. Ertl. Volume clipping via per-fragment operations in texture-based volume visualization. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 93–100, Washington, DC, USA, 2002. IEEE Computer Society.
- [90] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 169–177, New York, NY, USA, 1998. ACM Press.

- [91] L. Westover. Footprint evaluation for volume rendering. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 367–376, New York, NY, USA, 1990. ACM Press.
- [92] L. Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, New York, NY, USA, 1978. ACM Press.
- [93] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int. J. Comput. Vision*, 27(2):161–195, 1998.
- [94] X. Zhong, P. Liu, N. Geoganas, and P. Boulanger. Designing a vision-based collaborative augmented reality application for industrial training. *Information Technology*, 45(1):7–18, 2001.
- [95] J. Zhou, A. Döring, and K.D. Tönnies. Distance based enhancement for focal region based volume rendering. In *Bildverarbeitung für die Medizin*, pages 199–203, 2004.
- [96] P.G. Zimbardo and R.J. Gerrig. *Psychology and Life (17th Edition)*. Allyn and Bacon, 2004.