

Visual Computing for Medicine

Schriftliche Habilitationsleistung

zur Vorlage bei der
Fakultät für Informations- und
Kognitionswissenschaften
der Eberhard-Karls-Universität Tübingen

vorgelegt von
Dr. rer. nat. Dirk M. Bartz
aus Tübingen

Tübingen
2005

Contents

Contents	i
List of Figures	v
List of Tables	ix
1 Introduction	1
I Foundations of Medical Volume Data	3
2 Acquisition and Interpretation of Medical Volume Data	5
2.1 Medical Image Data	5
2.2 Medical Image Data Modalities	9
2.2.1 X-Ray Imaging	10
2.2.2 CT Data	13
2.2.3 MRI Data	18
2.2.4 Rotational X-Ray	22
2.2.5 Other Modalities	23
2.3 Data Artifacts	25
2.3.1 Sampling Theorem	25
2.3.2 Undersampling and Aliasing	27
2.3.3 Partial Volume Effect	29
2.3.4 Interpolation Artifacts	31
2.3.5 Signal Artifacts	33
Bibliography	37
II Data Enhancement and Filtering	41
3 Volumetric High Dynamic Range Windowing	43
3.1 Introduction	44
3.2 Related Work	45
3.3 Volumetric High Dynamic Range Windowing	47
3.3.1 Zone Intensity Mapping	48
3.3.2 Volumetric Dodging-and-Burning	49

3.4	Results	52
3.4.1	Computed Tomography	52
3.4.2	Magnetic Resonance Imaging	53
3.4.3	Discussion	55
3.4.4	Error Metrics	58
3.5	Conclusion and Future Work	60
	Bibliography	63
III	Analysis and Classification	65
4	SignatureSpace: A Multidimensional, Exploratory Approach for the Analysis of Volume Data	67
4.1	Introduction	68
4.2	Parallel Coordinates for Volumetric Datasets	70
4.3	Analysis of Volume Data Using the SignatureSpace	75
4.3.1	Strategies for Using the <i>SignatureSpace</i>	81
4.3.2	Memory Usage	84
4.4	Results	84
4.5	Conclusion and Future Work	85
	Bibliography	87
IV	Segmentation, Volumetry, and Visualization	91
5	Segmenting The Mastoid: Allocating Space in The Head for a Hearing Aid Implantation	93
5.1	Introduction and Related Work	93
5.2	Methods	94
5.3	Results	96
5.4	Discussion	97
	Bibliography	99
6	Live-Wire Revisited	101
6.1	Introduction and Related Work	101
6.2	Method	102
6.2.1	Live-Wire Overview	102
6.2.2	Adaptive Propagation	103
6.3	Results	104
6.4	Discussion	104
	Bibliography	107

7 Accurate Volumetric Measurements of Anatomical Cavities	109
7.1 Introduction	110
7.2 Related Work	111
7.3 Objectives	111
7.4 Methods	111
7.4.1 Segmentation	112
7.4.2 Computation of the Volume	113
7.5 Results and Discussion	115
7.6 Conclusion	116
Bibliography	119
8 Hybrid Segmentation and Exploration of the Human Lungs	121
8.1 Introduction	122
8.2 Related Work	124
8.3 Methods for Virtual Bronchoscopy	125
8.3.1 Segmentation	126
8.3.2 Masked 3D Region Growing	126
8.3.3 2D Wave Propagation	127
8.3.4 2D Template Matching	130
8.3.5 Segmenting Blood Vessels	131
8.3.6 Virtual Endoscopy	132
8.4 Results	134
8.5 Conclusions and Future Work	137
Bibliography	139
9 Scaffolding-based Segmentation of Coronary Vascular Structures	141
9.1 Introduction	142
9.2 Structure and Function of the Heart	143
9.3 Related Work	144
9.4 Reconstruction of Coronary Blood Vessels	146
9.4.1 Virtual Endoscopy	147
9.4.2 Virtual Coronary Angioscopy	149
9.4.3 Results	152
9.5 Conclusions and Future Work	153
Bibliography	155
V Navigation	159
10 A Navigated Virtual Endoscopy System for Oral-, Maxillo-Facial Surgery and Neurosurgery	161

10.1 Introduction	161
10.2 Material and Methods	163
10.3 Discussion	164

Bibliography 167

11 Medical Augmented Reality based on Commercial Image Guided Surgery 169

11.1 Introduction	170
11.1.1 Medical Augmented Reality	170
11.1.2 Drawbacks of Existing Systems	171
11.2 Components of a Medical AR System	172
11.2.1 Camera / Instrument Tracking	172
11.2.2 Patient Registration	172
11.2.3 Creation of Preoperative Planning Data	172
11.3 The VectorVision IGS system	173
11.3.1 VectorVision Link	174
11.4 The New Augmented Reality System	175
11.4.1 Camera Tracking	175
11.4.2 Additional Support by the IGS system	180
11.5 Results and Discussion	181
11.6 Conclusion and Future Work	182

Bibliography 185

12 Occlusion Handling for Medical Augmented Reality Using a Volumetric Phantom Model 187

12.1 Introduction	188
12.2 Related Work	189
12.3 Augmented Reality based on Image Guided Surgery	190
12.3.1 Camera Tracking	190
12.3.2 Additional Support by the IGS System	193
12.4 Static Occlusion Handling	194
12.4.1 Volume Preprocessing	194
12.4.2 Integration into Rendering Process	198
12.5 Results	199
12.6 Conclusion	203

Bibliography 207

List of Figures

2.1	Voxels on a grid	6
2.2	Volume cell	7
2.3	Trilinear Interpolation	7
2.4	Bilinear contours in a volume cell	8
2.5	X-Ray Imaging	11
2.6	Digital subtraction angiography to analyze cerebral blood vessels	12
2.7	Principle of CT	14
2.8	Comparison of T1- and T2-weighted cerebral MRI data.	20
2.9	Abdominal MRI data.	21
2.10	Rotational angiography of cerebral arteries	23
2.11	Sine-function undersampling	26
2.12	Sine-function sampling	27
2.13	Undersampling artifacts	28
2.14	Box filter for low pass filtering	29
2.15	Partial volume effect	30
2.16	Binary segmentation block artifacts	31
2.17	Staircasing artifacts	32
2.18	Corrected staircasing artifacts	34
2.19	CT artifacts	35
3.1	Metal artifacts in CT Head dataset	52
3.2	Histograms of 3D CISS MRI scan of a head	54
3.3	Example image slices from MRI CISS dataset	55
3.4	Visual differences between different windowing operators	57
3.5	Other windowing operator applied to the MRI CISS dataset	59
3.6	Dataset overview	62
4.1	Parallel coordinates	71
4.2	Unfolding clusters with opacity modulation	72
4.3	Background noise removal in engine dataset	74
4.4	Background noise removal in lobster dataset	75
4.5	Partial volume effect in the tooth dataset	77
4.6	SignatureSpace of the cones dataset	78
4.7	SignatureSpace of the fuel dataset	79
4.8	SignatureSpace of the hydrogen dataset	79

4.9	SignatureSpace of the plexiglas phantom dataset	80
4.10	Colormapped tooth dataset	80
4.11	SignatureSpace of the MRI head dataset	83
4.12	Isosurface rendering of signatures volumes	85
5.1	System block diagram	95
5.2	Segmentation process	95
5.3	Segmented mastoid	97
6.1	Propagation of control points.	103
6.2	Adjusting the position of a propagated control point. . .	104
6.3	Propagation example	105
7.1	6mm phantom dataset	110
7.2	Image slice from phantom segmentation	113
7.3	Volume computation of a voxel sphere	113
7.4	Cases of volumetry classification	114
7.5	3D model of a cerebral ventricular system	115
8.1	Results of three segmentation stages	122
8.2	Segmentation pipeline	126
8.3	Logarithmic histogram of CT dataset	127
8.4	Completion of bronchi walls	128
8.5	Template matching	130
8.6	Reconstruction of iso-surface of dataset 2	133
8.7	Virtual bronchoscopy of dataset 1	135
8.8	Virtual bronchoscopy of dataset 2	138
9.1	Reconstructed blood-filled cavities of the heart	143
9.2	Slice views of heart dataset	147
9.3	Virtual endoscopic view through the circumflex branch .	148
9.4	Virtual endoscopic view through the LAD	149
9.5	Path voxelization and segmentation	150
9.6	Axial view of heart dataset	151
9.7	Cardio vascular imaging - Reconstruction of the coronary artery tree	154
9.8	Cardio vascular imaging - RIVA stenosis	154
10.1	Navigated endoscopy in maxillo-facial surgery	162
10.2	Multi-modal representation of the cerebral ventricular system	163
10.3	Architecture of integrated virtual and navigated endoscopy	164
11.1	VectorVision IGS system and tracked webcam	173
11.2	Infrared tracking of surgical instrument	174
11.3	Calibration of a surgical tool in the IGS system.	176
11.4	6-DOF instrument data delivered by image guided surgery.	177
11.5	Marker corners are indicated by the pointer tool	178

11.6	Marker corners are defined in a clockwise order	179
11.7	The ARToolKit marker is used as the common point of reference.	180
11.8	Overview of transformation matrices	180
11.9	Multiple AR views	181
12.1	VectorVision image guided surgery system and tracked webcam	191
12.2	Marker corners are defined in a clockwise order from P_1 to P_4	192
12.3	Virtual tumor model in augmented reality system	193
12.4	Comparison of slice from input and visual hull volumes	195
12.5	Slice of a volume dataset of the plastic skull	197
12.6	After the raycasting process	197
12.7	Overview of the volume processing pipeline	198
12.8	Visual hull volumes computed for four volume datasets	200
12.9	Comparison of the iso-surfaces	201
12.10	Occlusion handling in our augmented reality	202

List of Tables

2.1	Hounsfield values for selected tissue types	14
3.1	Logarithmus computation time	49
3.2	Windowing computation times	56
3.3	Entropy of windowing methods	60
4.1	Datasets examined with SignatureSpace	81
7.1	Volumetric measurements	116
8.1	Breakdown of typical scene complexity	136
12.1	Dimensions and bit depth per voxel of test datasets . . .	199
12.2	Triangle counts in iso-surfaces generated from unprocessed input and visual hull volumes	201
12.3	Average time required for rendering iso-surfaces into the depth buffer	203

Chapter 1

Introduction

Visual computing addresses various aspects of the processing of image data. These aspects include the full visual computing pipeline starting from data acquisition, via image processing, the graphical representation of the data, to the interaction with them.

These aspects have been in the focus of my work of the past 13 years, and became the focus of the research group “Visual Computing for Medicine” (VCM). Although we concentrate on the medical domain as major application fields, I would like to stress that the research contributions are addressing general problems of image processing and computer graphics. Furthermore, they can be applied to many other application fields.

The following five parts structure the content into major stages of the visual computing pipeline, while the chapters focus on the specific contributions to the pipeline stages, mostly in the context of medical application. Part I starts discussing the fundamentals of medical imaging in Chapter 2. Specifically, it gives an overview on the structure of volumetric image datasets in Section 2.1, and describes typical data acquisition modalities, such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and several more (Section 2.2). Since volumetric data is of a discrete nature, the fundamentals of signal theory – the sampling theorem (Section 2.3.1) – and the source of image artifacts will be discussed in Section 2.3. The consequences of incorrect sampling for discrete volumetric image data are widespread and lead to typical artifacts like aliasing (Section 2.3.2), the partial volume effect (Section 2.3.3), interpolation artifacts (Section 2.3.4), and signal artifacts (Section 2.3.5) itself.

The second part discusses approaches on the enhancement and filtering of volumetric image data in Chapter 3. Namely, it addresses the necessary windowing operation, where a higher dynamic range – or high precision voxel value range – is mapped into a smaller one. In order to maintain a good contrast, an operator for high dynamic range windowing is introduced and its effectiveness for volumetric image data is demonstrated.

Part III focuses on the analysis and classification of volumetric data. Chapter 4 introduces the *SignatureSpace*, which combines methods from abstract information visualization with methods from volume visualization to analyze volumetric image data from various sources. Specifically, a parallel coordinates display has been extended to be able to process a large number of samples. Each of the axis of the parallel coordinates display represents different signatures, which represent derived, auxiliary voxel volumes from the original volume dataset. These signature volumes can be used for a multidimensional, exploratory analysis of the original volume dataset.

Part IV focuses on a variety of approaches for the segmentation of different organs, the estimation of the respective occupied volumes, and their visual representation. In Chapter 5, a volume processing pipeline based on the region-growing paradigm is presented for the segmentation of the mastoid, a spongy bone structure behind the human ear. This segmentation is required for the allocation of space for an hearing aid. Next to the region-growing approach, the LiveWire-approach is used for the interactive correction of some parts of the segmentation. Unfortunately, LiveWire does not extend well into three dimensions. Hence, an adaptive propagation technique for LiveWire is introduced in Chapter 6.

The estimation of the volume of specific organs is important for the assessment of the performance of organs (e.g., for the heart), for treatment plans (e.g., an aneurysm), and for many other specific tasks in interventional medicine. In Chapter 7, a method for the estimation of the volume of segmented organ structures is presented and evaluated based on scanned phantom models.

The final segmentation approach is discussed in Chapter 8, addressing the complex and difficult segmentation of human airways and blood vessels. Furthermore, this chapter focuses also on the visual representation of multi-organ structures from endoscopic viewpoints.

Lastly, Part V describes the use of augmented reality techniques that combine real-time video-streams with interactive computer graphics representation of virtual information in the operating room (OR). Chapter 10 outlines how this can be achieved by combining an image-based navigation system with the virtual endoscopy system. The technical questions of this combination are discussed in Chapters 11 and 12. Specifically in the last chapter, the difficult occlusion problem in augmented reality displays is addressed.

Part I

Foundations of Medical Volume Data

Chapter 2

Acquisition and Interpretation of Medical Volume Data

In this chapter, we introduce important concepts of medical volume data in the scope of data acquisition and related data artifacts. In particular, we will discuss the structure of volumetric data and introduce the respective terminology (Section 2.1). In Section 2.2, we briefly discuss the most important image data modalities, such as CT and MRI data and focus on their characteristics which are essential for image analysis and visualization. This section is complemented by a discussion of frequent image data artifacts and their sources (Section 2.3).

2.1 Medical Image Data

Medical image data is usually represented as a stack of individual images. Each image represents a thin slice of the scanned body part and is composed of individual pixels (picture elements). These pixels are arranged on a two-dimensional grid, where the distance between two pixels is typically constant in each direction. For most medical image modalities¹, the horizontal (x) and vertical (y) directions have identical distances, which are called the *Pixel Distance*. A constant pixel distance allows the calculation of the actual position by multiplying the respective distance value with the respective pixel index i . If we assume that i is indexing the horizontal x position and j is indexing the vertical y position, the position of pixel $P_{i,j}$ is determined. Figure 2.1 left describes the 2D image grid arrangement.

Volumetric data combines individual images into a three-dimensional representation on a three-dimensional grid (see Fig. 2.1 right). The data elements are now called *voxels* (volume elements) and they are located on the grid points. In addition to the horizontal (x) and vertical (y)

¹A modality in our context is identical with a specific image acquisition technique, such as CT or MRI.

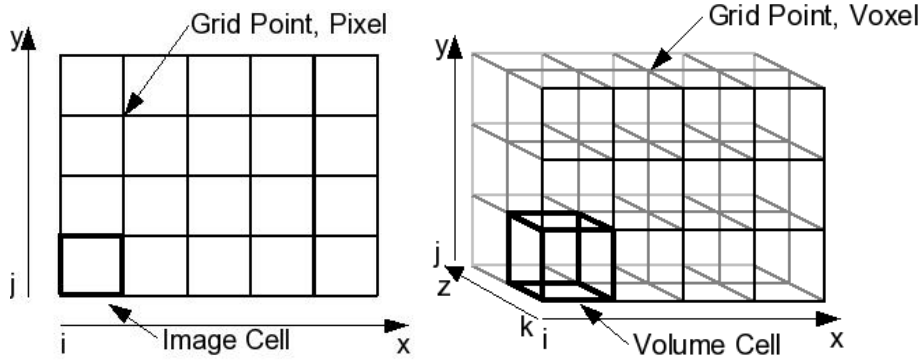


Figure 2.1: Left: 2D grid, where all pixels of an image are arranged on the grid points of the grid. Right: In volume datasets, the voxels are arranged on a 3D grid.

dimensions, we now also have a dimension representing the depth (z). The distance between two neighboring images (slices) is called *slice distance*. The three distances in every direction are also called *voxel spacing*. Similar to the pixels, the position of a voxel $V_{i,j,k}$ is determined by the distance values and the voxel index (i, j, k) (Fig. 2.2).

If the pixel distance is identical to the slice distance, we speak of an isotropic grid or dataset. If this is not the case, we speak of an anisotropic grid. Most datasets in medical imaging are anisotropic and in many cases the slice distance is several times larger than the pixel distance (Fig. 2.2). Eight neighboring voxels which form a cuboid (see Fig. 2.1 right) are called a *volume cell* or short a *cell*.

Since the grid points can be seen as being aligned on a cartesian coordinate system, this grid type is also called cartesian or uniform grid. Its special features are constant or regular spacing in each dimension, regular geometry that can be computed by the grid index and the spacing, regular topology (it has the same connectivity for all grid points), and finally it is only composed of cuboid (volume) cells. While most datasets in medical imaging are defined on such grid types, it is important to be aware that there are other grid types with varying spacing, varying geometry, varying topology, and lastly, with varying cell types. These grids can be of a non-uniform, curvilinear, or irregular type. More information on these grid types can be found in the literature on scientific visualization [SML98].

A volume dataset is defined only at the discrete grid positions. In many cases, however, we need to calculate sample points within a volume cell. In the early days of volumetric data processing, this was achieved by nearest-neighbor interpolation, where a sample point within a volume cell was assigned the voxel value of the voxel with the smallest distance from that sample point². While nearest-neighbor interpolation

²The concept of nearest-neighbor interpolation was also the basis of an outdated voxel concept, where a voxel was considered the whole space that had a constant voxel value, similar to the volume cell in the modern voxel concept.

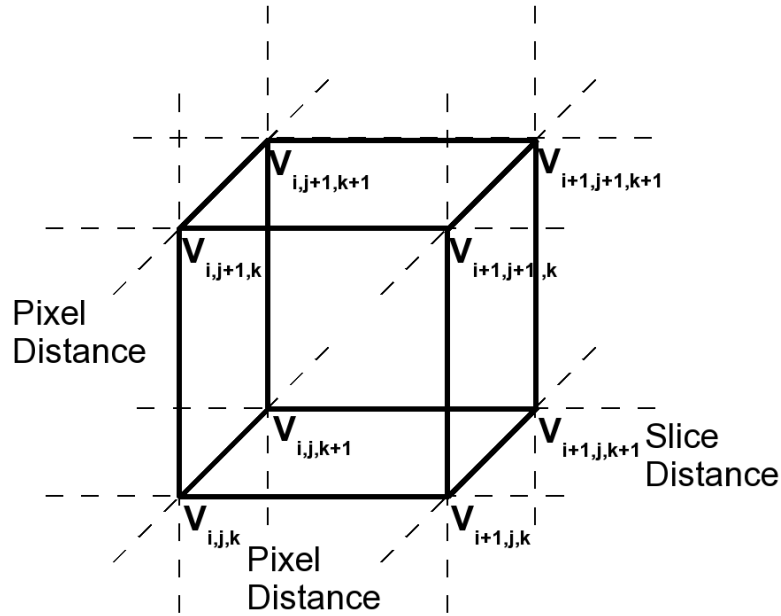


Figure 2.2: Volume Cell: The indices i, j, k indicate the voxels that span the volume cell. Pixel distances are the spaces between the voxels that are indexed via i and j (within one image slice), and slice distance is the space between the voxels indexed via k (between image slices).

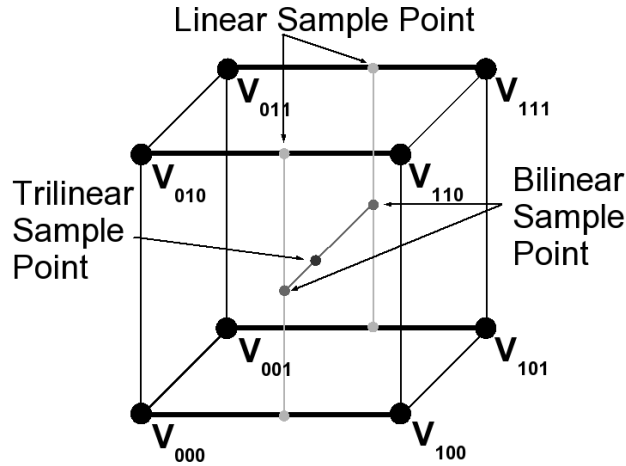


Figure 2.3: Trilinear Interpolation: Linear sample points are computed on the frontal and backward edges of the volume cell. Based on these points, bilinear interpolation is computed by linear interpolation. Finally, the trilinear sample point is computed by linear interpolation between the bilinear sample points.

introduced only very modest computational costs, its visual quality became quickly unacceptable. The low visual quality was due to the discontinuous interpolation results in the middle between neighboring voxels, resulting in a very blocky appearance of reconstructed surfaces.

Today, more advanced interpolation schemes are used. The most popular scheme is trilinear interpolation, which is composed of seven linear interpolations (Fig. 2.3). In the first composition step, four linear interpolations compute the weighted sample points between two neighboring voxels on an edge of a volume cell along one direction (either x , y , or z). In the second composition step, the result of two such linear interpolations of voxels that are located on the same face of the volume cell are combined by linear interpolation to compute the result of a bilinear interpolation. Finally, the result of the two bilinear interpolations on opposite faces of the volume cell are combined by a linear interpolation into the final trilinear sample point within the volume cell.

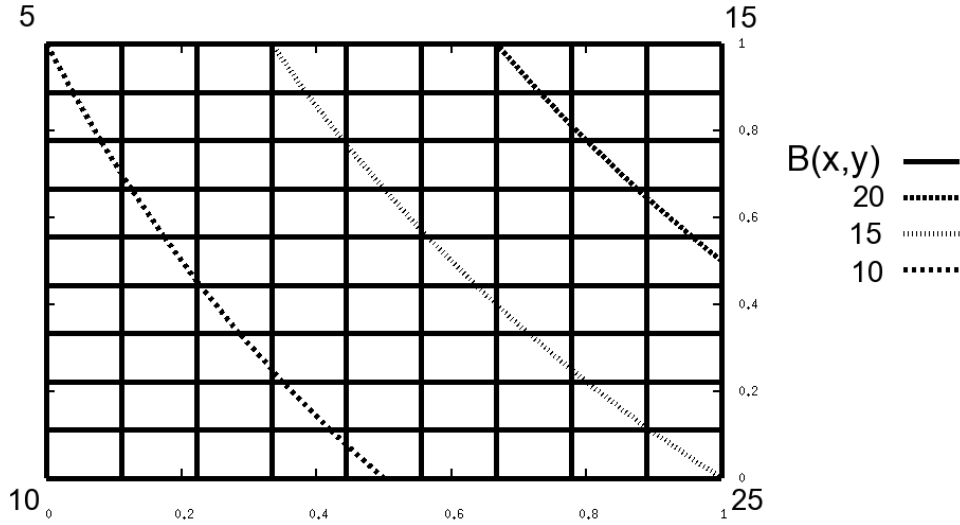


Figure 2.4: Bilinear contours in a volume cell. The bilinearly interpolated 2D cell $B(x,y)$ is seen from above. The voxel values on the corners are 5, 15, 25, 10 (clockwise, starting from the upper left corner). Three isocontours with the isovalues 10, 15, and 20 are shown.

As described above, three levels of linear interpolations are chained into one function. Since each interpolation uses its own variable, it is technically a trivariate linear function. However, if an isosurface with a specific isovalue c is extracted, a root-finding operation is performed. One variable is successively substituted by the remaining ones, and we get a cubic function for the isosurface, or a quadratic function for an isocontour in 2D (see Fig. 2.4).

For a more simple representation of the interpolations in Equations 2.1- 2.3, we will abbreviate terms like $V_{i,j,k}$ with V_{000} and terms like $V_{i+1,j+1,k+1}$ with V_{111} , where the index represents the increment of the position indices i, j, k . $V_{i+1,j,k}$ for example is now abbreviated with V_{100} . Equation 2.1 shows a linear interpolation between two voxel values, and Equation 2.2 shows the respective bilinear interpolation

between four voxel values. α , α_1 , and α_2 represent the respective interpolation weights for linear and bilinear interpolation, and x, y, z for the trilinear interpolation³.

$$L(\alpha) = V_0 * (1 - \alpha) + V_1 * \alpha, \quad |0 \leq \alpha \leq 1 \quad (2.1)$$

$$\begin{aligned} B(\alpha_1, \alpha_2) &= L_0(\alpha_1) * (1 - \alpha_2) + L_1(\alpha_1) * \alpha_2 \\ &= (V_{00} * (1 - \alpha) + V_{10} * \alpha) * (1 - \alpha_2) \\ &\quad + (V_{01} * (1 - \alpha) + V_{11} * \alpha) * \alpha_2 \end{aligned} \quad (2.2)$$

$$\begin{aligned} T(x, y, z) &= B_0(x, y) * (1 - z) + B_1(x, y) * z \\ &= ((L_0(x) * (1 - y) + L_1(x) * y)) * (1 - z) \\ &\quad + ((L_2(x) * (1 - y) + L_3(x) * y)) * z \\ &= (((V_{000} * (1 - x) + V_{100} * x)) * (1 - y) + \\ &\quad ((V_{010} * (1 - x) + V_{110} * x)) * y) * (1 - z) \\ &\quad + (((V_{001} * (1 - x) + V_{101} * x)) * (1 - y) + \\ &\quad ((V_{011} * (1 - x) + V_{111} * x)) * y) * z \end{aligned} \quad (2.3)$$

Trilinear interpolation involves only the immediate voxel neighborhood of a sample point, which are the voxels of the enclosing volume cell (Fig. 2.3). Other interpolation schemes involve higher-order interpolation functions, such as triquadratic or tricubic spline functions, which take a larger voxel neighborhood into account. Alternatively, the sample value can be reconstructed by convolving the voxel values with a Gaussian filter kernel.

2.2 Medical Image Data Modalities

The sources of volumetric data are manifold. Typically, they can be generated by simulations or by measurements. In our context, we focus on measured volumetric data, which has been acquired by 2D or 3D scanning technology. In this section, we will discuss different modalities that are available for the acquisition of interior data of the object to be scanned. We can classify the available methods into projection techniques, which combines available information into a less-detailed representation (usually 2D), and into tomographic techniques, which can generate a representation that maintains the information at their 3D position. More specifically, a tomographic representation is composed of several sections⁴ of the scanned object. The word “tomogra-

³Since these interpolation can be computed in all three directions, we use only one voxel index for linear, and two voxel indices for bilinear interpolation, and α , α_1 , and α_2 as parameters respectively.

⁴More precisely, these sections are computed from several input measurements convolved with filter kernels. Hence, they could be viewed as a type of projection as well, but at a much smaller scale.

phy” itself is composed of the Greek works “tomos” for section (or slice through that object) and “graphi” for writing.

Typical examples for projection modalities are X-Ray and ultrasound, which will be explained in Sections 2.2.1 and 2.2.5. Computer Tomography (Section 2.2.2) and Magnetic Resonance Imaging (Section 2.2.3) are examples for tomographic modalities.

Alternatively, we can classify image acquisition methods after their underlying physical principles. X-ray (Section 2.2.1), Computer Tomography (Section 2.2.2), rotational angiography (Section 2.2.4), and ultrasound (Section 2.2.5) are based on absorption and reflection of an emitted signal (x-ray or ultrasound waves). In contrast, Magnetic Resonance Imaging (Section 2.2.3) measures the magnetic spin (actually the relaxation times) of the nuclei of specific atoms.

The question which modality is chosen depends largely on the circumstances of scanning. In particular medical data might be acquired in emergency situations, where a fast scanning method might be preferred to one with a better image quality that is significantly slower (time constraint). In most cases, however, it depends on the focus organ and how it can be scanned (diagnostic constraint). In the past years, one circumstance became more important; data is acquired intra-operatively (immediacy constraint). This option becomes important in cases where the organ anatomy (location, orientation, and shape) has changed after a pre-operative acquired scan. However, intra-operative imaging modalities have disadvantages, such as complex setup, low image quality, or they are difficult to use.

After acquisition, the data is usually provided in the “DICOM” format. DICOM is an acronym for “Digital Imaging and COmmunications in Medicine” and defines a PDU-based (protocol data unit) format for a (largely) vendor-independent data exchange. DICOM accommodates the requirements of many different modalities and is a complex format. Next to commercial systems, many OpenSource libraries [OFF05] are available that can manage DICOM data.

In the remaining Sections, the various image data modalities are explained, namely X-Rays (Section 2.2.1), Computer Tomography (Section 2.2.2), Magnetic Resonance Imaging (Section 2.2.3), Rotational or C-arm (bi-plane) X-Ray (Section 2.2.4), and several other image acquisition techniques (Section 2.2.5).

2.2.1 X-Ray Imaging

The discovery of X-rays by Wilhelm Conrad Röntgen in 1895 was the birth of medical imaging. For the first time, interior parts of a body could be visualized without actually cutting into it. Figure 2.5 shows examples of X-ray imaging. The left image shows one of the first images made by Conrad Röntgen. It is said to be the hand of his wife. The right image shows a more recent X-ray image of a left hand. The

name X-rays was coined in Röntgen's original paper [Rön95], where it denotes a new, unknown ("X") kind of radiation. They are also known as Röntgen rays. Note Figure 2.5 shows two X-ray images that have inverted intensities, due to different recording techniques. In the left image, low densities (e.g., air) are mapped to white, and high densities (e.g., bone) are mapped to dark. In contrast, the right image shows high densities in white, and low densities in black.

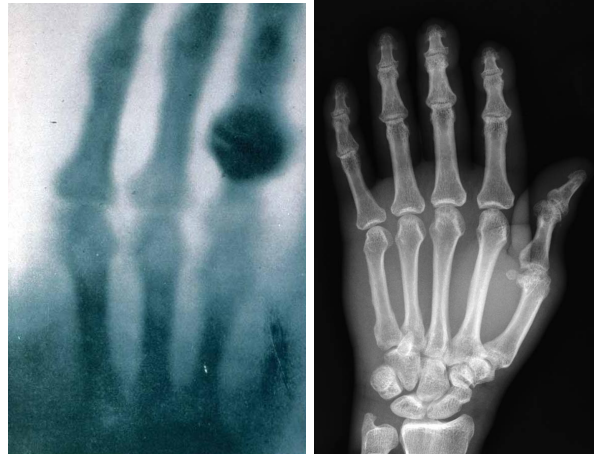


Figure 2.5: X-Ray Imaging: Left - X-ray image of the right hand of the wife of Röntgen, right - recent X-ray scan of a left hand (Image courtesy of Harvard Medical School).

X-ray images are based on the absorption and scattering of X-ray quanta traveling through the scanned object. The denser the current part of that object is (e.g., bone or metal), the more the X-ray quanta are absorbed.

The X-ray quanta are generated by electrons that are shot at a target material. When these electrons hit the target material (usually Tungsten), the kinetic energy is transformed into heat (approximately 99% [LOPR97]) and X-ray quanta. The X-ray quanta in fact represent a spectrum of different energies, which are based to the larger extent on the *bremsstrahlung*, when the electrons are slowed down near the nuclei of the target material, and the characteristic radiation, when the electrons force electrons of the target material into a higher energy level (photo effect). If the electrons cease to exist, we speak of *absorption*. If they are deflected (with or without loss of energy), we speak of *scattering*.

The speed (and hence the energy) of the electrons depends on the strength of the electric field that accelerates the electrons. If the generating voltage is larger than 100kV, we speak of hard X-rays, otherwise of soft X-rays. If hard X-rays are used to generate X-ray quanta, the scattering effect dominates the interaction with the object. Consequently, the quanta are only absorbed in very dense material. In

medical imaging, this is exploited for the representation of bone structures. In contrast, soft X-rays result predominately in absorption, and hence this can be used for the representation of soft-tissue. This, however, results in more energy intake of the tissue itself, which in turn leads to more tissue damage. Therefore, soft X-rays are more harmful than hard X-rays.

Today, images generated by X-rays are typically provided as digital image data. The necessary digitalization can basically be achieved in two ways. Either they are recorded on film material, which needs to be scanned afterwards (“computed radiographs”), or it is directly recorded by a detector array (“direct radiograph”).

Fluoroscopy: Fluoroscopy is a special version of direct radiographs, where the result can be immediately represented on a screen. Since the permanent exposure to X-radiation requires a significantly lower intensity, the overall image quality does not equal the quality of regular still X-ray images. However, the focus here is on dynamic processes that are visible over time. Typical examples are angiographies, where an injected contrast agent is tracked while it travels through the respective blood vessels.

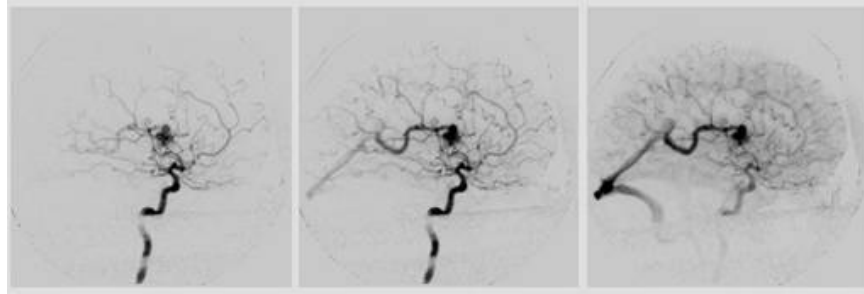


Figure 2.6: Digital subtraction angiography to analyze cerebral blood vessels. Images are courtesy of B. Tomandl, University Hospital Erlangen-Nürnberg

Digital Subtraction Angiography (DSA): DSA tracks the blood flow through contrast agent enhanced imaging. In order to maximize the representation of the blood vessels, an “mask image” of the object before injecting the contrast agent is taken. During the contrast-enhanced phase, this mask image is subtracted from acquired images, thus leaving only the changing parts of the images, which in case of a DSA is typically a blood vessel tree. Other potential obstructing structures are removed with the mask image.

Mammography: This variation of X-ray imaging is used to acquire images of the soft tissue of a female breast, usually for cancer provisions.

However, soft X-rays are required, which in turn increase the potential tissue damage of the patient.

Discussion

Compared to other (especially 3D) image modalities, 2D X-ray imaging provides only a projection image of the scanned object. Hence, the representation of positional relationships is difficult. In most cases, two X-ray images are acquired from two different viewing directions (bi-planar) to provide a better spatial understanding of the different anatomical structures.

Despite the availability of many different 3D modalities and their advantages, 2D X-ray imaging is still the standard image acquisition method in clinical practice.

2.2.2 CT Data

The introduction of X-ray Computed Tomography (CT) in 1972 by Godfrey Hounsfield [Hou72] provided for the first time a volumetric representation of objects, and not only a 2D projection of a volumetric object. Generally, it is seen as one of the major milestones in medical imaging [CJS93].

In essence, CT data is composed of a series of individual X-ray images that are composed into one volume dataset. These X-ray images are acquired by an emitter/detector system that is rotating around the object to be scanned (Fig. 2.7). Each X-ray image represents an intensity profile measured by the detectors. From the intensity profiles of a full rotation, the slice image of the scanned object is computed based on the Radon transform [CJS93], which essentially is based on the Fourier theorem. Afterwards, a table on which the scanned object is positioned will be moved forward and the next image slice of the object will be acquired.

The computed intensity value represent the densities of the scanned object. In medical imaging, these intensity values are normalized into Hounsfield Units (HU), named after Godfrey Hounsfield. This normalization maps the data range into a 12bit range, where the intensity of water is mapped to zero, and air is mapped to -1000. This normalization map is formalized in Equation 2.4, where I_{H_2O} represents the intensity value of water. Table 2.1 shows the Hounsfield units for different human organ tissues.

$$HU = \frac{I - I_{H_2O}}{I_{H_2O} - I_{Air}} * 1000 \quad (2.4)$$

The major differences between the development stages of CT [Kal00] are different projection reconstruction algorithms and the emitter and detector architecture (data acquisition). The first generation of CT

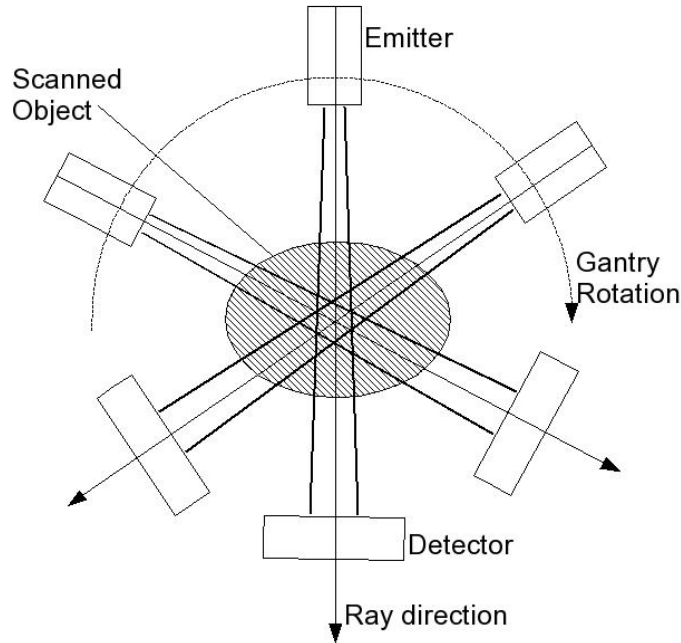


Figure 2.7: Principle of CT image acquisition: The emitter/detector system (gantry) rotates around the scanned object and acquires a series of projection profiles.

scanners was basically the experimental setting of Hounsfield's CT scanner. It used a single pencil-like X-ray beam emitter and a single detector on the opposite side of the object. To acquire a data slice, the pencil beam was translated along the object and rotated afterwards for the next series of beams. All together, the costly mechanical movement of emitter and detector caused long scanning times, ranging from several minutes to several hours at a resolution of 80×80 pixels per scan/slice. Furthermore, the single emitter/detector architecture enabled only a poor utilization of the emitted radiation.

The next generation and first commercial generation of CT scanner

Table 2.1: Hounsfield values for selected tissue types. From [LOPR97]

Tissue type	Hounsfield Value Interval
Air	-1000
Lung tissue	-900...-170
Fat tissue	-220...-30
Water (H_2O)	0
Pancreas	10...40
Liver	20...60
Heart	20...50
Kidney	30...50
Bones	45...3000

used small angle fan beams (which were computationally transformed into parallel beams) and multiple detectors to scan two neighboring rotational projections at the same time. This technique reduced the number of necessary rotations, and hence the required scanning time (10 - 60 seconds, up to several minutes per slice) needed for a sufficient reconstruction. It also provided a better utilization of the emitted radiation. Both first and second generation techniques are parallel beam devices, deploying different reconstruction algorithms than the next fan beam devices.

The next improvement increased the fan beam angle and the number of detectors to cover the whole object, thus the translating movement became unnecessary and reduced the scanning time to five seconds per slice. Similar to the previous techniques, the radiation was emitted at fixed time intervals and was measured by the detectors.

In the fourth generation, the rotating detector was replaced by a fixed circular ring detector, which reduced the technical effort of moving the larger mass of emitter and detector. Here, the radiation was permanently emitted and only the detectors were enabled at certain intervals. However, several problems of ring detectors led to further developments in favor of third generation scanners. Besides the higher costs for the detector ring, specific X-ray scattering problems reduced the image quality of these systems, while collimator technology could reduce the scattering problems with a rotating emitter/detector system of the third generation.

Currently, the state-of-the-art are spiral or helical CT, where the emitter/detector system is rotating permanently around the object, while the object is moving continuously in the perpendicular direction to acquire a full data volume. This technique allows faster scanner, due to the continuous rotating movement of the emitter/detector system. This saves the time for the previously required time to accelerate and slow down these heavy parts of the scanner. Since 1999, multiple layers of emitters and detectors are combined to create multi-slice CT scanners (twin or quad slices), realizing fast and isotropic scanning of large object areas. Today (mid 2005), the state-of-the-art in multi-slice CT scanners acquire up to 64 slices (actually spirals) at a time

Besides the architectural development, different volume/slice reconstruction algorithms are used in the various systems and generations. The basic approach is the back transformation of the slice projections into the volume slices by the Radon transform [CJS93]. In the beginning of computed tomography, algebraic reconstruction techniques (ART) were used to solve this back projection problem. However, the high computational costs of iteratively solving the large matrices⁵ rendered this approach as not feasible for standard applications [Kal00]. The standard method today is filtered back-projection [FDK84], where each projection is composed according to the measured direction. While

⁵The size of a reconstruction matrix is equal to the resolution of the slice.

there are parallel and fan beam methods to perform this back-projection, the current fan beam methods for CT are more complex and less efficient than state-of-the-art parallel beam reconstruction algorithms. Hence, the projections of today's fan-beam scanners are re-sorted in parallel beams before the actual reconstruction. Another modification is required to address the continuously moving object tray of modern spiral CT scanners, where a z-interpolation corrects the measured projections according to the tray movement [Kal00]. In the future, cone beam reconstruction algorithms will probably replace the current methods [Kal00], which are successfully used in rotational X-ray (see Section 2.2.4) already.

Parameters of CT Scanning

Datasets acquired by a CT scanner can be described by their resolution, composed of the *number of slices*, the *number of pixels per slice*, and the *voxel distances*⁶ (Section 2.1). The number of pixels in one slice is also referred to as *image matrix*; an image matrix of 512×512 simply describes an image slice of that resolution. If we have a CT dataset with an image matrix of 512×512 and 300 slices, we have a volume dataset with the resolution $512 \times 512 \times 300$. The distances between the voxels are differentiated into the slice (out-of-plane) and pixel distance (in-plane). As already mentioned in Section 2.1, a dataset with a slice distance different to the pixel distance is described as an anisotropic dataset, while equal distances constitute an isotropic dataset.

A slice in a volumetric dataset from a scanned resource does not represent an infinitesimal thin data slice; it represents a specific area of the image matrix times the *slice thickness*. This is essentially based on the fact, that a reconstructed image pixel value is a weighted average of the local spatial neighborhood. This neighborhood in z-direction (between slices) can be seen as the slice thickness.

The rotating emitter/detector system of a CT scanner – through which the table on which the scanned object is positioned is moved – is called the *gantry* (Fig. 2.7). The tilt of the gantry defines also the tilt of the image slices. Since most CT datasets are of anisotropic spacing, it can be useful to adapt the tilt gantry to the target organ of an examination, e.g. to ensure optimal sampling of the respective organ structures. Therefore, the *gantry tilt* is an additional parameter of CT scanning.

When X-rays are casted from the emitter, they usually diverge like a fan from the emitter. Since these diverging X-rays will influence neighboring signal acquisition, it is useful to focus the rays along the main direction of the emitter. This is achieved by an aperture that

⁶Note that the typical definition of resolution only describes how much detail can be represented in an image (or volume). It does not describe how accurately the original objects are represented. Our concept of resolution also includes the voxel distance and therefore also describes (to some extent) the acquisition accuracy.

limits the fan-out of the X-rays and is called *collimation*. The related collimation parameter specifies how open the aperture (the collimator) is.

Two more important parameters for CT scanning are *pitch* and *table increment*, also known as *table feed*. The latter parameter (table feed or table increment) specify how much the table (where the object to be scanned is positioned) is advanced through the scanner during one rotation of the gantry. The pitch p defines the ratio between the table increment d and the *total slice collimation*, which is the factor of number of slices per rotation M (one for a regular spiral CT scanner, or four for a four-slice multi-slice CT scanner) and the slice thickness S . Therefore, we can compute the pitch as $p = d/(M \cdot S)$. If, for example, we have a four-slice CT scanner ($M=4$), a slice thickness of $S=1\text{mm}$, and a table feed of $d=6\text{mm}$, we can compute the pitch as $p=1.5$ [CT05].

Many CT examinations concern the blood vessel system. However, the blood vessels are difficult to read in standard CT scans. Therefore, a contrast agent (usually iodine) is injected into the artery system to enhance the representation of the arteries (and later during the venous cycle, the respective veins). Important parameters to consider for a contrast-enhanced CT scan are quantity of contrast agent (typically between 300-400mgI/ml) and how the contrast agent is administered (as bolus, continuously, or in a varying distribution). Depending how long the scan is performed after the injection, the visual representation of the arterial or venous system will be enhanced (with an increased voxel intensity). Note that the image intensity of adjacent organs can also be slightly increased, due to the contrast agent.

Discussion

There are several issues that need to be considered in CT data acquisition. First of all, the radiation dose that is associated with each CT scan (since it is based on X-ray technology) should be reasonable considering the diagnostic value of the enhanced image quality. Furthermore, a higher resolution will cause also a higher noise level, which in turn can be compensated by a higher radiation dose. The lower noise level of a lower resolution image is mainly due to the stronger inherent smoothing.

The data quantity (number of voxels, number of image slices) of a CT scan depends largely on the chosen scanner technology (single-slice, multi-slice) and on the specific exam. Typically, a head exam will require approximately 200 slices, while an thorax exam requires 400-500 slices already. A periphery vessel exam might even require up to 1700 slices (although usually much less). Since every slice can contain an image matrix of 512×512 voxels, the overall data size of such an exam is enormous and needs to be addressed by the data processing and visualization system, as well as by the archiving system.

Finally, data artifacts can pose a considerable challenge for the representation of data. These artifacts can be due to technical problems (although they became quite seldom), or due to signal artifacts, such as metal artifacts, e.g., due to tooth fillings. Section 2.3 contains a discussion of the various causes and effects of data artifacts.

2.2.3 MRI Data

Magnetic resonance imaging is based on different properties of human tissue in a magnetic field. In particular, the different densities of Hydrogen nuclei are exploited for image generation. They can be considered as small dipole magnets aligning themselves either parallel or anti-parallel along the magnetic field. While aligned in that field, the Hydrogen protons (a Hydrogen nucleus consists only of a proton) spin arbitrarily around the axis of the field. This spinning is called the *precession* of the nuclei. If energy as a radio-frequency pulse signal (RF) is applied to the magnetic field at the *Larmor*-frequency, the nuclear resonance forces the protons to receive some of the energy from the RF. This pulse also forces all the protons to spin synchronously, or *in phase* and increasingly to flip into the anti-parallel orientation of higher energy, until the number of parallel protons is equal to the number of anti-parallel oriented protons. The duration of the RF pulse determines the amount of precession; e.g., a *90 degree pulse* will force the protons into a 90 degree precession, where the precession vector of the protons is completely perpendicular to the magnetic field, resulting in a zero *z*-component (along the magnetic field). After the stimulation of the protons, they slowly release the received energy, *de-phase* and re-align with the magnetic field. This *relaxation* is described as *free induction decay (FID)* and is divided into the transverse and longitudinal relaxation. The first relaxation of the transverse magnetization – also called spin-spin relaxation – describes the de-phasing of the *x/y*-component of the precession. The time required for this relaxation is called T2 and is in the order of a few milliseconds. The longitudinal, or spin-lattice (or spin-grid) relaxation describes the re-alignment of the precession with the magnetic field, thus the restoring of the *z*-component. This relaxation time is called T1 and is in the order of a second. T1 largely depends on the material (tissue), its structure, and its surrounding tissue. Since water has a long T1 and T2 relaxation time, tissue that contains a large ratio of water will also have a long T1 and T2 time. The actual measured volumetric information is the proton density σ , which needs to be reconstructed at the specific voxels.

To reconstruct the spatial information of the measured signal, two additional gradient magnetic fields are applied [LOPR97]. The first field selects the slice in *z*-direction of the volume, since only one layer of protons suffices the Larmor-frequency for the main and gradient magnetic fields. An additional gradient field in *x*-direction selects a *y*-slab. By

rotating the gradient fields, other projection directions are measured. In particular this feature is very useful, since the slice direction can be adapted to the medical exam. This is not possible with a CT scan (always axial).

Alternatively, a different approach became available with the 2D Fourier reconstruction, which requires only one gradient field. Here, the x/y -coordinate is encoded into that signal [LOPR97] by the increasing de-phasing signal of the transverse relaxation along the x and y directions, which generates specific frequencies into the signal. Thus, the RF intensity (proton/spin intensity) is encoded into the intensity of the signal, while the position is encoded into the frequencies [CJS93, LOPR97]. Note that the quality of the image data strongly depends on the homogeneity and strength of the magnetic field [Hou80, Hah05].

Depending on the strength of the magnetic field, MRI scanners are characterized as “low field” scanners, with a magnetic field strength of up to 0.8T (Tesla), as “full field” scanners, with a magnetic field strength of 1.5T, or recently as “high field” scanners with a typical magnetic field strength of 3T or more.

Parameters of MRI Scanning

MR acquisition itself is characterized by a huge parameter space. Different protocols describe sequences of various RF pulses and the actual measurement of the signal, where the time between the initial stimulation and the measurement is called *echo time* T_E and the time between two (initial) stimulation cycles is called *repetition time* T_R . By varying T_E and T_R , different weight data can be achieved; i.e., with short echo and repetition times, the proton density is mostly dominated by the T1 relaxation time (T1 weighted), with long T_E and T_R , the T2 relaxation time is dominating the signal (T2 weighted), or with a long T_R and a short T_E , the resulting data is neither T1 nor T2 and can be seen as “the pure (proton) density function” (proton-weighted) [CJS93].

Figure 2.8 compares T1- and T2-weighted images acquired for neuroradiological diagnosis. While T1-weighted acquisition sequences emphasize tissue signals, it de-emphasizes fluid signals with a very low image intensity. In contrast, T2-weighted acquisition sequences emphasize fluid signals with a high image intensity.

Orthogonal to the T1/T2-weighting, a variety of acquisition sequences are available. Each sequence puts emphasis on different optimization targets. An MRI FLASH (Fast Low Angle Shot) sequence for example enables a very rapid image acquisition. Combined with a T1-weighted image acquisition, it is often used for the acquisition of dynamic phenomena, e.g., the blood flow. Other pulse sequences, such as 3D CISS (Constructive Interference Steady State) combined with a T2-weighting, emphasize on high spatial resolution. Further sequences focus on different aspects of the metabolism. MRI Angiography (MRA)

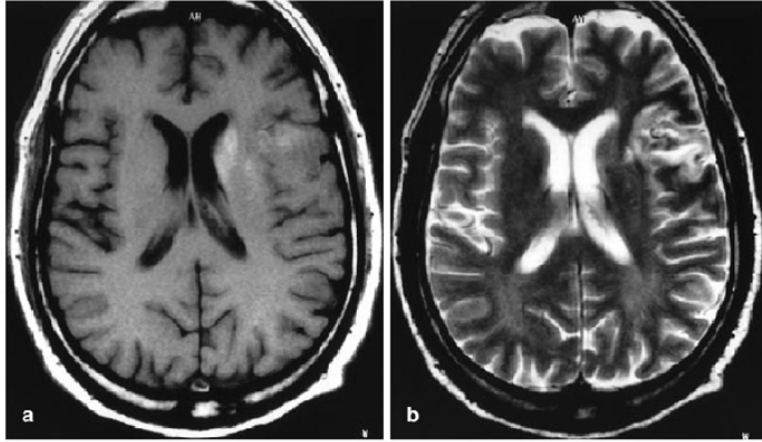


Figure 2.8: Comparison of T1- and T2-weighted cerebral MRI data. In the left T1-weighted image, the cerebrospinal fluid appears dark whereas in the right image the same region appears white. Also grey and white matter exhibit different intensity values in these images.

is focusing on vascular imaging with or without a contrast enhancing agent. Typical examples of these (non-contrast agent) imaging protocols are Time-of-Flight (TOF) and Phase-Contrast-Angiography (PCA). Note that flow artifacts can reduce the quality and in some cases of turbulent blood flows lead to pseudo stenosis.

The large potential parameter space of MRI leads to a high number of imaging protocols that put an emphasis on different functional or physiological properties of the tissue. Therefore, an MRI examination does often contain a variety of five or more different MRI scans. New pulse sequences are also continuously explored and developed by MR physicists to optimize for special diagnostic procedures.

A typical MRI scans take between 2 and 25 minutes. This relatively long scanning time is dominated by the relaxation of the spins, not by the time required for the measurements. MRI is also characterized by a certain spatial resolution. In general, MRI data – acquired with a full field MRI scanner – have a lower resolution compared to CT image data (the in-plane resolution/pixel distance is often between 1 and 2 mm and the slice distance 2 to 5mm). This however, can be different with modern high field MRI scanners, which acquire significantly higher (spatial and/or temporal) resolution image data.

MRI Image Artifacts. While we will discuss image artifacts in general in Section 2.3, we will briefly summarize typical MRI artifacts in this paragraph. MRI image artifacts can be sorted into artifacts by physiological causes (e.g., artifacts due to (body) motion or (blood) flow), hardware problems such as inhomogeneities of the magnetic field, and problems with the inherent physics (chemical shift, presence of metal, etc.). A discussion of MRI artifacts can be found at [MT05]



Figure 2.9: Abdominal MRI data. The liver appear light in the peripheral regions and darker in the central regions. Data is courtesy of J. Debatin, University of Essen.

with images of the respective artifacts at [Gre05].

Intra-operative MRI data. MRI data can also be acquired in an intra-operative setting to provide up-to-date data. Special “open” MRI scanners are used which allow interventions, while acquiring intra-operative images. This, however, also requires the use of MR-suitable instruments that can be used in or near a strong magnetic field. Due to the significantly less powerful magnetic field of Open-MRI of 0.5-0.8T (“low field”), the quality of the images is also significantly lower, due to limited resolution and high noise level. Nevertheless, it provides intra-operative imaging, while the surgeons still can access the patient in the MRI scanner.

Recently, full magnetic field MRI scanners (1.5T, “full field”) have been introduced into the operating room (OR) [NGvKF03]. Since these scanners are not open, the patient has to be moved while lying on the operating table into the scanner for the image acquisition and moved back to the surgeons, outside the area that is immediately influenced by the magnetic field. Next to the space required for the full-sized MRI, additional space in the OR is needed for the safety distance to the scanner, and for the transportation of the anaesthetized patient. This renders full field intra-operative MRI very difficult to install, since regular ORs will not be able to accommodate the required space.

One of the phenomena that is targeted by intra-operative MRI is the brain-shift effect in neurosurgery. This effect describes the local changes of location and shape of the brain, after opening of the skull and dura [NBG⁺01]. These changes render the pre-operatively acquired image data (at least partially) useless, since it does not represent the current situation sufficiently accurate.

Functional MRI. Functional MRI or short fMRI provides func-

tional information with respect to different areas in the human brain. These areas are activated by means of sensory stimulation and motor or memory tasks. fMRI takes advantage of measurable changes of local blood flow and oxygenation. [RIT⁺01] illustrate how intra-operative and functional MR images may be combined to enhance neurosurgical interventions (see also [TOW⁺03]).

Diffusion Tensor Imaging. As another important MRI flavor, MRI Diffusion Tensor Imaging (MRI DTI) has recently become an image modality of high interest. Essentially, several diffusion weighted datasets are measured and combined into a volume dataset that provides a second order tensor at every voxel position. This data represents diffusion information how Hydrogen is transported. The basic idea of DTI is that if this diffusion has a clear direction (one major eigenvalue in the respective tensors), it indicates the direction of a brain fiber. Note that these fibers are significantly smaller than the current resolution of MRI scanners. Therefore, it represents only a statistical probability of the direction of a whole set of fibers in that location.

Discussion of CT and MRI Modalities

CT and MRI data have different advantages and disadvantages, resulting in situation specific applications. Typically, CTs are used in any kind of emergency situations, since it is usually a significantly faster, and may be even more accurate scanning modality. Furthermore, it allows the representation of most life-threatening conditions. However, CT data acquisition does involve a certain amount of exposure to radiation and allows only for a limited soft tissue discrimination.

In contrast, MRI provides a large variety of image acquisition protocols, which allow a flexible and good soft-tissue differentiation. Another advantage is that MRI does not involve any ionizing radiation and has no known harmful effects on the body tissue. However, skeletal structures cannot easily be represented with MRI. Furthermore, the strong magnetic field around the MRI scanner generates constraints for patients with a pace-maker or other (medical) devices.

In terms of image data processing, the standardized image intensity (in Hounsfield units) of CT data allows a significantly easier processing than MR data, where the image intensities vary from protocol to protocol and from scan to scan (even with the same patient). These introduces difficulties in the detection of organ boundaries, and even for the interpretation of the respective image data.

2.2.4 Rotational X-Ray

Rotational X-ray is a recent scanning technology which started from digital subtraction angiography (DSA, see Section 2.2.1). Hence, most of the associated applications are angiography applications and this technique is frequently referred to as rotational angiography [Fah99,

GSR⁺00] (Fig. 2.10). Since the rotation of the emitter/detector system is performed on a C-shape mounting, it is also called C-arm X-ray. Many systems consist of a pair of emitter/detector systems to acquire two projections at a time. These systems are named Rotational bi-planar X-Ray.

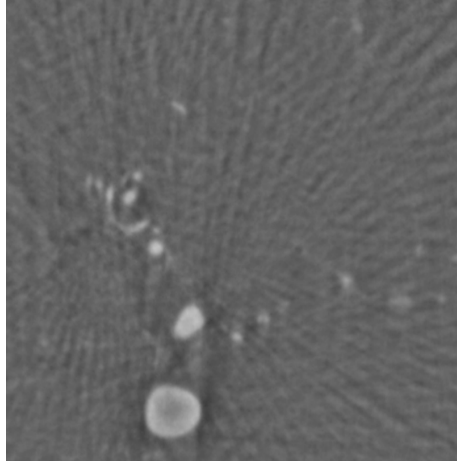


Figure 2.10: Rotational angiography of cerebral arteries. Due to the used reconstruction algorithm, only bones (not visible) and blood vessels that are enhanced with a contrast agent can be represented well. The image shows a large aneurysm of the anterior cerebral artery in the lower left.

To generate volumetric datasets, a series of up to 132 X-ray projections are taken from a rotation range of 200 degrees around the scanning object. In contrast to CT, rotational X-ray is using a full array of up to 1024^2 detectors which allow the measurement of a full cone of rays. To account for reconstruction errors [Kal00], a modified back-projection algorithm is used. Additionally, special filter kernels are used to further reduce potential artifacts. Current rotational angiography systems provide very high resolution, isotropic datasets, good reconstruction quality (for selected organs, such as bones and contrast agent enhanced blood vessels), and a high data acquisition speed of up to 13 seconds for a full scan.

2.2.5 Other Modalities

Next to the already mentioned medical imaging modalities, other scanning technologies have been introduced into medical imaging. In this context, only a very brief overview on some of these scanning technologies is provided.

Ultrasound: With ultrasound scanning, sound waves are emitted at very high frequencies (more than 20K Hz) from the ultrasound probe (“transducer”) through the respective body part. Similar to a sonar, the

reflections of these sound waves are received by the transducer and used to generate the respective images. In the *B-mode* or *brightness mode*, the resulting pixels are representing the echos received by the transducer and show the respective body parts. Images generated by this mode typically show anatomical details. In contrast, the *A-mode* can be used to measure distances within the body, the *M-mode* to measure motion (e.g., of the heart), and finally, the *Doppler-mode* is used to measure velocities. Combined with the B-mode, it typically is used to generate images of blood vessels and their respective blood flow.

Special ultrasound scanning devices allow also three-dimensional image acquisition by acquiring and accumulating multiple scans.

One of the advantages of ultrasound is that it typically is a real-time imaging method that can be widely used – in particular with intra-operative imaging – and does not employ ionizing radiation.

Position Emission Tomography - PET: PET scans are used to trace metabolistic activity. A radiopharmaceutical substance (“tracer”) – labeled with a radioactive isotope – is injected and traced throughout the body to its target area. At the target area, the substance is processed by the metabolism and the radioactive isotope decays, emitting gamma rays in antiparallel direction, which are detected by the PET scanner. This way, the location of current metabolistic activity – e.g., brain activity while performing a specific task – is revealed.

Often, PET scan image data is combined with CT or MRI data to combine the functional data with morphological data of the respective body parts. This requires the alignment (“registration”) of the generated volumetric datasets.

PET has two major drawbacks. First, it generates rather low resolution images (increasing the difficulties of registration with another data modality), and second, the tracer substances have to be produced on-site with a cyclotron, a rather short time before they are injected. This renders PET scanning a rather expensive image modality.

Recently, combinations of PET and CT have been introduced to provide an inherent registration of PET and CT image data.

Single Photon Emission Computed Tomography - SPECT: SPECT scanning works on the same basic principle as PET. A radiopharmaceutical substance is traced by a gamma ray camera to reveal the location of the body part where the tracer is processed. In contrast to PET, SPECT is more limited in terms of spatial and temporal resolution, and they are more limited what they can monitor. However, SPECT tracers decay more slowly than PET tracers, thus allowing the examination of longer lasting metabolistic functions.

2.3 Data Artifacts

Data artifacts are a phenomenon that is virtually omni-present in medical imaging. Nevertheless, the source and appearance of these artifacts is often not clear to people in charge of the acquisition and representation of medical image data. Fortunately, medical imaging data have their theoretical roots in digital signal processing, and therefore, artifacts can be explained by and traced back to the foundations of signal processing. In this section, we provide a brief introduction into the sampling theorem and its influence on medical imaging. Specifically, we describe and explain the major data artifacts in medical volumetric data, based on medical 3D scanners.

Important concepts for the discussion of many data artifacts are the spatial domain and the frequency domain. So far, we basically only examined datasets represented in the spatial domain, where the data is arranged as voxels or pixels on a 3D or 2D grid. Since this grid spans into space, depending on the number of voxels in each dimension and the spacing between the voxels, they are a representation in the spatial domain.

The frequency domain is a different representation of spatial data, which have been Fourier transformed. Now the data (or signals) are represented by a band of different frequencies, hence the name frequency domain. To transform the signals from the frequency domain to the spatial domain, we need to apply the Inverse Fourier transformation.

2.3.1 Sampling Theorem

The main basis in signal theory is the sampling theorem that is credited to Nyquist [Nyq24] – who first formulated the theorem in 1928 – and to Shannon [Sha49] – who formally proved it in 1949. The sampling theorem is also known as the Whittaker-Nyquist-Kotelnikov-Shannon sampling theorem.

The sampling theorem essentially says that a signal has to be sampled at least twice as fast as the original signal. Since signals and their respective speed can be easier expressed by frequencies, most explanations of artifacts are based on their representation in the frequency domain. The sampling frequency demanded by the sampling theorem is also called the *Nyquist frequency*, or the reciprocal term *Nyquist rate*.

The transformation of signals into the frequency domain (Fig. 2.11) is done by the Fourier transformation, which essentially reformulates the signal into a cosine function space. If, for example, we transform the sine function into the frequency domain, this results in a peak (or Dirac impulse/function) at the frequency of the sine function (Fig. 2.11 right). Due to the symmetry of the Fourier transform for real values, there are two peaks on both sides of the ordinate (y) axis.

However, we present here only a very compact explanation of sam-

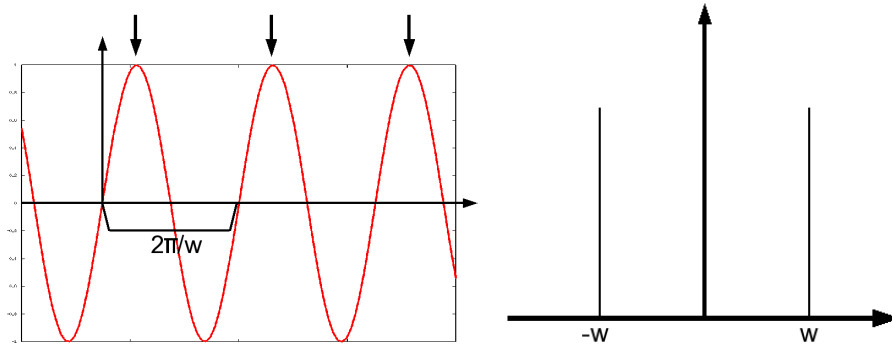


Figure 2.11: Sine-function is sampled (arrows) at same speed as sine periodicity of $T = 2\pi/w$. Left: spatial domain representation, right: frequency domain representation, where the frequency is represented by two symmetric peaks at $-w$ and w .

pling, which we based – with a bit of hand-waving – in the spatial domain. More details on that transformation and the frequency-domain-based interpretation can be found in the books by Lehmann et al. [LOPR97] and Glassner [Gla95].

As all image data can be interpreted as a spatial or Fourier-transformed signal, we base our discussion on a simple example of an one-dimensional signal, the sine function. This function has a straightforward representation in the spatial and the frequency domain. Figure 2.11 left shows a section of the sine function around the origin. If we transform this continuous representation into a discrete representation, we need to take samples of the continuous sine function to measure its characteristics. Figure 2.11 left demonstrates what happens if we take the samples at the same speed (or frequency) as our original function. Since the sine function has the periodicity $T = 2\pi/w$ (or the frequency of $w/2\pi$), this sampling speed would be also T . As Figure 2.11 left shows, sampling the sine function at the same speed would recover always the same sine value in different periods, thus pretending that we are recovering a constant function.

If we now increase the sampling speed to half of the periodicity of the continuous function, the minimum demand of the sampling theorem, we can now recover the correct characteristic of the sine function, as it can be seen in Figure 2.12 left. However, depending on what exact position in the period T of the original function we take the sample, we recover different amplitudes of the original signal. In an unfortunate case, we always sample the zero crossing of the sine function, as shown in Figure 2.12 right. In this case, the characteristics could be correctly recovered, but the amplitude of the signal was recovered in an unfortunate way, so we are back with a constant signal. Overall, sampling at a rate satisfying the sampling theorem does not guarantee that the full signal strength is reconstructed, although higher sampling rates usually

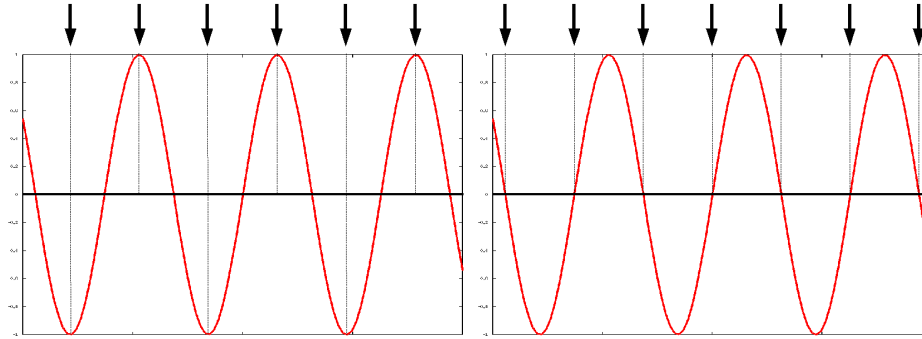


Figure 2.12: Sine-function is sampled (arrows) at Nyquist rate. Left: Sampling at optimal position (detects peaks and valleys), right: Sampling phase shift - only a constant (zero) amplitude is sampled.

approach the original strength.

In the frequency domain, sampling of the original signal is described as the convolution of the original signal with a comb function (with peaks repeating at the sampling frequency). Due to the periodicity of the comb function (we take samples at regular positions), the convolved signal also exposes a replicating pattern. Since exactly one of these signal patterns is used, we need to select one copy of it with a low pass filter, a technique that is explained in the next section. If the sampling rate is not high enough, it will result in an overlap of the replicating patterns, and hence in the inability to select one copy of the pattern.

Finally, it is important to keep in mind that medical image data does consist of a full spectrum of frequencies, which are revealed once the Fourier transform of the data is calculated. Therefore, the respective limiting cut-off frequency of this spectrum should be taken into account, when estimating the correct sampling frequency.

2.3.2 Undersampling and Aliasing

Aliasing is a phenomenon that is directly related to sampling. Essentially, it is caused by an incorrectly reconstructed signal, due to insufficient sampling. Therefore, the signal is mistaken for another signal, its *alias*. A typical visual result of aliasing is the Moiré artifact pattern, which can also be observed in our daily environment, e.g., when looking through two fences while we are moving. The top row of Figure 2.13 shows examples of the Moiré artifact, where the surface of the scanned phantom object is sampled at a speed below the Nyquist rate. This insufficient sampling is also called undersampling. Another undersampling artifact is shown in the middle row of Figure 2.13, where an insufficient slice distance is not able to recover the full geometry of the scanned object.

There are basically two possible solutions to overcome undersam-

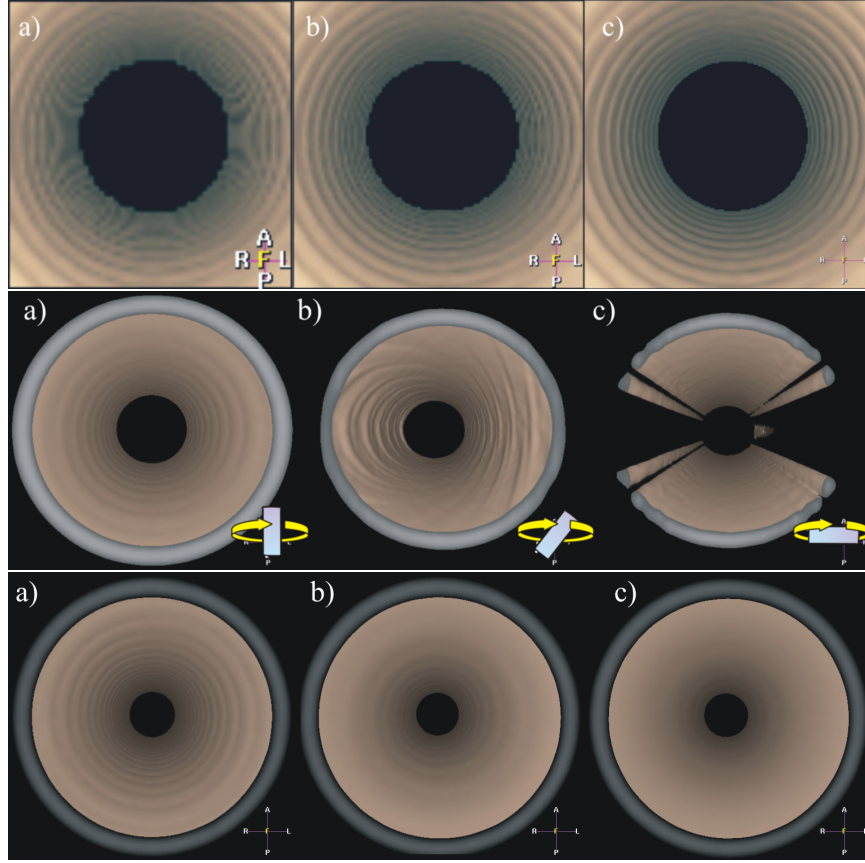


Figure 2.13: Tube phantom with perspective projection from an endoscopic viewpoint. Top row: The sampling rate is increased from the left to right and the Moiré artifacts are reduced. Middle row: Different slice and pixel distance in anisotropic datasets can lead to insufficient sampling, if the object orientation is changed. Bottom row: If too many frequencies are removed by a low pass filter, details will disappear (the number of removed frequencies is increasing from left to right). Images are courtesy of F. Dammann and A. Bode of the University Hospital Tübingen.

pling; first of all, the sampling rate can be increased until we satisfy the Nyquist rate. Figure 2.13 shows in the top row how the Moiré artifacts are reduced by increasing the sampling rate.

The other solution is to band-limit the original signal by performing a low pass filtering step first. As mentioned above, a signal will be represented by a spectrum of frequencies in the frequency domain. If this spectrum extends into a higher frequency range that we cannot correctly reconstruct, we remove those frequencies from the spectrum by convolving the original signal with a low pass filter (removing higher frequencies, keeping lower frequencies, hence it is called a low pass filter). This means for the convolved signal that it does not contain the offending higher frequencies; the band of frequencies has been limited.

In the spatial domain, this results in a smoothed or blurred signal (or image). It depends on the size and quality of the low pass filter how

much the new signal is blurred; the more high frequencies are removed, the more blurred is the new signal. The bottom row of Figure 2.13 demonstrates what happens if too many high frequencies are removed.

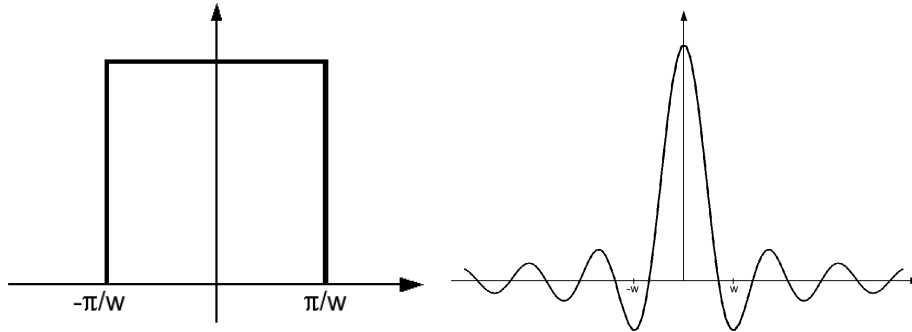


Figure 2.14: The (1d) box filter has the shape of a rectangle in the frequency domain (left) and is a sinc function (right) with infinite support in the spatial domain.

Extensively band-limited signals often lead to excessively blurred data, where more details than intended are removed. Therefore, the low pass filter has to be carefully designed in terms of its size (or extent). The choice which low pass filter to use is also important. Typically, three different low pass filter are considered; a box filter, a triangle filter, and a Gaussian filter. The box filter is called that way, because of its frequency domain representation as a rectangle (Fig. 2.14 left). If it is convolved with the spectrum of the original signal, only the frequencies covered by the box are kept in the convolved signal. The triangle filter works similar, except that it introduces different weights for frequencies off-center of the triangle filter. Both filters have one common drawback; while they have a simple representation in the frequency domain, their representation in the spatial domain is the sinc function whose oscillating infinite extent must be truncated (Fig. 2.14 right). This truncation however, leads to an imperfect box filter representation in the frequency domain.

This situation is less difficult with the third low pass filter candidate, the Gaussian filter, which essentially is a Gaussian function in the frequency and the spatial domain. Furthermore, it has a better weighting cut-off than the triangle filter. While it still has an infinite extent that needs to be truncated, it lacks the oscillating behavior. Furthermore, it always has positive weighting values. More information on filtering can be found in Glassner's book [Gla95].

2.3.3 Partial Volume Effect

The partial volume effect is an issue that specifically concerns medical image data. Due to limited resolution at volume reconstruction – when the volume dataset is reconstructed from the measured signals

– large intensity differences cannot properly be reconstructed. In order to avoid undersampling artifacts, the signal is band limited and results in a smoothed intensity in the transitional region between the low and high intensity signals. Therefore, this sample represents a local neighborhood of the original signal values.

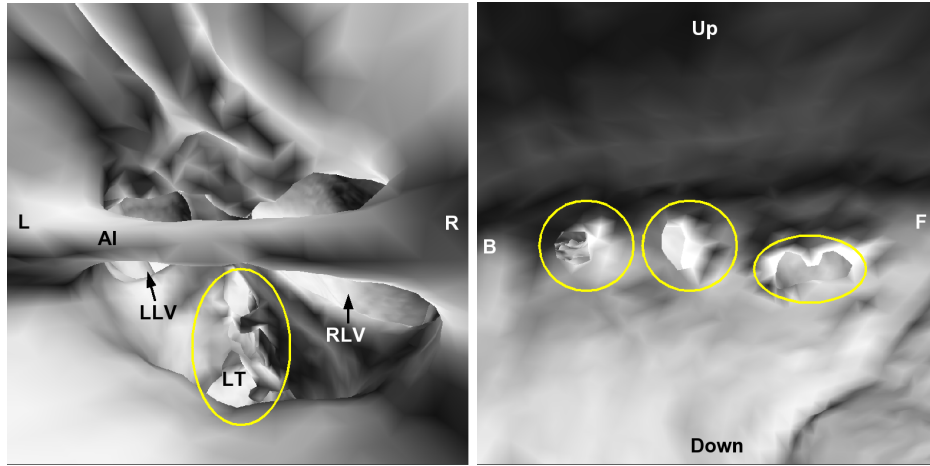


Figure 2.15: Partial volume effect in an MRI scan of the cerebral ventricular system: The thin membrane Lamina Terminalis (LT, yellow/bright marking) at the floor of the third cerebral ventricle could not be fully reconstructed (left). False connections (yellow/bright markings) are due to the incompletely reconstructed septum between the upper lateral cerebral ventricles (right). L and R point to the left and right directions, B points to the back direction (or posterior), and F points to the frontal direction (or anterior). Up and down point to the respective directions. AI depicts the Adhesio Interthalamica, and LLV and RLV to the entrance of the left and right cerebral ventricles (Foramen of Monro).

The practical consequences for medical image data are broad. Large differences in intensities can be caused, if neighboring tissues create significantly varying signal responses. If for example a bone is measured with a CT scanner, the neighboring tissue with a significantly lower density will create a steep intensity difference that cannot properly be reconstructed. Hence, the voxels on the boundary between the two tissue materials will be low pass filtered. While this low pass filtering is a good approach to avoid aliasing artifacts, the smoothed intensities create at the same time false connections or holes in the resulting image data. Figure 2.15 left demonstrates this effect. A thin membrane located in the area indicated by the yellow/bright oval is affected by the partial volume effect. Since the neighboring cavities, which are separated by this membrane, are represented with high intensities, the low intensity thin membrane could not be fully reconstructed. Figure 2.15 right shows a similar situation. The measuring resolution of the MRI scanner is not sufficient to represent the (low intensity) septum between the (high intensity) upper cerebral ventricles, resulting in false connections between the upper ventricles.

Partial volume effects are omnipresent in medical image data, and must therefore be taken into account for segmentation, classification, and quantitative image analysis. If a small anatomical structure with a significantly different intensity than the surrounding voxels needs to be scanned, the resolution must be chosen accordingly. Hahn discussed the impact of the partial volume effect to quantitative image analysis. For more information, please refer to [Hah05].

2.3.4 Interpolation Artifacts

As mentioned earlier in Section 2.1, many visual representations use interpolation schemes to compute data at positions between defined grid points. One of the most popular schemes is trilinear interpolation (Eq. 2.3), which is used for position and normal computation. Since the normals in volume datasets are usually approximated by gradients, central differences are typically used to estimate the gradients [HB86]. Unfortunately, standard central differences will generate artifacts, if the intensity differences are large, or the grid spacing is anisotropic.

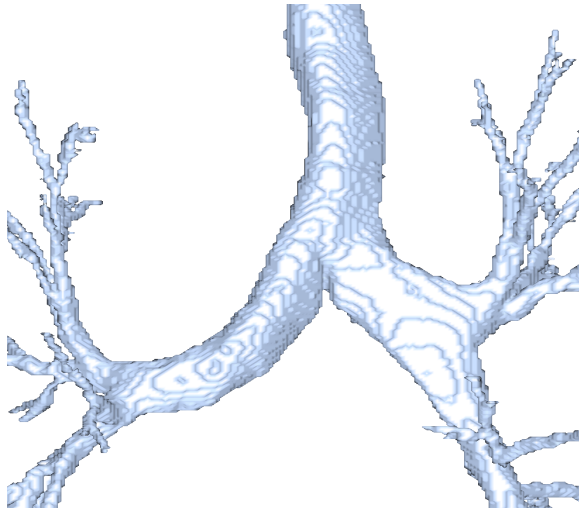


Figure 2.16: Binary segmentation block artifacts. The image shows the block artifacts of an isosurface reconstruction of the label volume of a bronchi dataset. The dataset is courtesy of D. Mayer of the University Hospital Mainz.

In the case of binary segmentation, individual voxels are labeled as part of the segment (on) or not part of the segment (off). If the respective isosurface representing the material interface between the segment and its neighborhood is computed, large intensity differences on the material interface (all off-voxels are set to zero, all on-voxels contain the original voxel value) occur. In these situations, the positions of the isosurface and in particular their normals will experience the above mentioned artifacts and result in a blocky appearance (Fig. 2.16). This blockiness aggravates, if the isosurface is computed on the label volume

itself, where only one (for voxels of the segment) and zero intensities are available. In these cases, all smoothing effects of the partial volume effects are gone.

Fortunately, there are several remedies for these interpolation artifacts, if the original data is used for isosurface computation and the label volume is only used as a map to indicate the voxels of the segments. The first remedy adds a layer of off-voxels around the segment voxels. This additional off-layer allows smoother local gradients on the material interface of the segment, since the intensity differences are significantly smaller than to zero intensities, due to the partial volume effect [HB86]. Another possible remedy changes the original data values and it might hence be less appropriate for certain situations, in particular in diagnostics. Here, the material interface in the voxel values is smoothed by applying a low-pass volume filter, which generates a smooth transition between the segment and the off-voxels. However, this also changes the resulting isosurface and must hence be applied with great care.

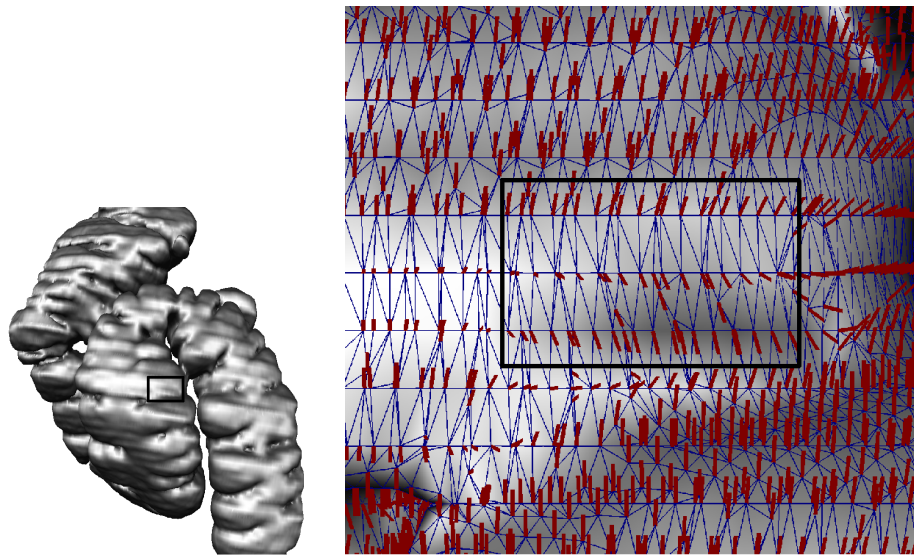


Figure 2.17: Staircasing artifacts in the polygonal reconstruction of the left flexure of a colon dataset. The right image shows a magnification of the marked area in the left overview image. Here, the flipping (red) normals at the staircase artifact (in the black rectangle) demonstrate the origin of the artifact.

A related interpolation issue arises also with the normal estimation based on central differences on anisotropic grids. Typically, the normal at a computed sample point within a volume cell is based on the trilinear interpolation of normals approximated by central differences at the surrounding voxels of this volume cell. This gradient estimation scheme, however, assumes an isotropic grid, since all three vector components are computed the same way. Unfortunately, anisotropic datasets have different voxel distances in the three different spatial ori-

entations; usually the slice distance (z) is significantly larger than the pixel distance (x,y). Since these differences are not properly addressed in most rendering approaches, the normals at the computed sample points are distorted in the different spacing direction (typically in slice direction). Figure 2.17 left demonstrates this **staircasing artifact**, while the right figure shows the normals in the marked area of the left image in a polygonal isosurface representation⁷. Note that the flipped normal direction causes the staircasing artifact; the vertex positions are already correct.

Fortunately, there are also possible remedies, which address the anisotropic grid spacing. Probably the easiest approach is to correct the sample point normals according to the spacing. Figure 2.18 top row shows a possible solution, where the central differences of the normal estimation scheme are scaled by the voxel spacing, a solution which is similar to curvilinear or structured grid datasets, where the normals also need to be corrected to address different cell sizes. As a variation of this approach for a polygonal representation, we can also compute the normals of the polygons (e.g., the crossproduct for triangles) and distribute a weighted average of these normals to the participating vertices. However, this normal computation scheme produces less smooth normals than the (corrected) gradient based method.

Alternatively, the data volume can be re-sampled into an isotropic grid dataset using an appropriate reconstruction filter (e.g., a Gauss-filter). The results of this approach can be seen in the lower row of Figure 2.18. While the quality of the resampling solution is better than for the re-scaled version, a drawback of this solution is the significantly increased volume size, which requires more memory and rendering resources.

There are several other possible solutions that reduce the appearance of staircasing artifacts. An overview of normal estimation schemes was provided by Möller et al. [MMMY97]. More recently, Neumann et al. suggested to use linear regression to estimate better gradients [NCKG00].

2.3.5 Signal Artifacts

One class of artifacts in medical volume data is caused by the data acquisition techniques themselves. While artifacts due to malfunctions of the measurement units (e.g., the detector array of a CT scanner) or other parts of the mechanical and electronic components became rather uncommon with the improving technology, other artifacts still persist.

Among the most notorious signal artifacts are *metal artifacts*, which appear if metal (e.g., steel or gold) is present in the body part to be scanned. Unfortunately, metal is used for many implants, such as tooth

⁷Note that staircasing artifacts can appear as well in direct as in indirect volume rendered images.

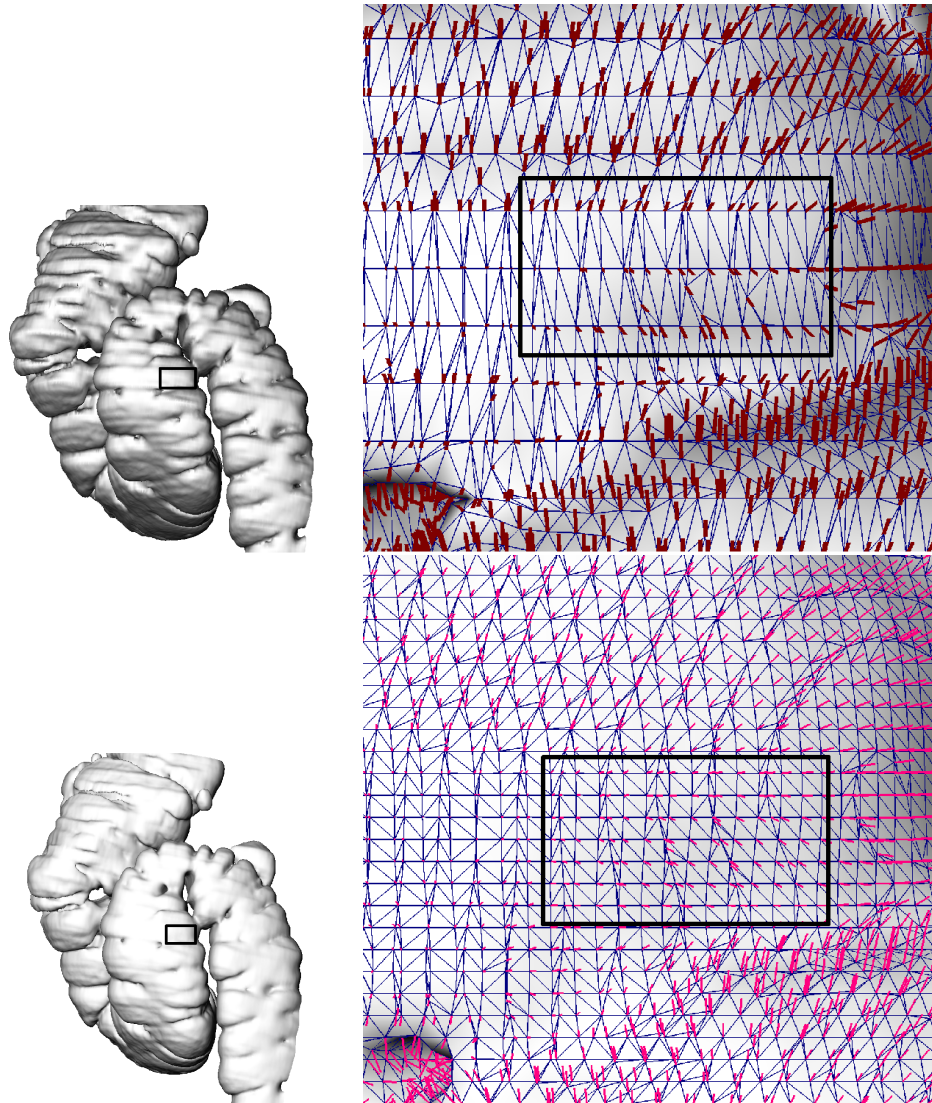


Figure 2.18: Corrected normals to significantly reduce the staircasing artifact in the polygonal reconstruction of the left flexure of a colon dataset. The top row shows the correction based on the scaled gradient components, and the bottom row shows a corrected representation based on a Gaussian resampling of the volume. The left column show the overall result, the right column shows a magnification of the marked staircase region in the left image with the now corrected normals, compared to Figure 2.17.

fillings, which are present in virtually every scanned mandible or maxilla bone, or more rarely, a metallic projectile. Similar to the partial volume effect, metal artifacts are partially due to an insufficient sampling of the very high metal/tissue intensity difference [Kal00]. Furthermore, the X-rays are almost blocked by the metal, what actually contributes most to the signal artifacts and can affect large areas of several image slices in a CT dataset (Fig. 2.19 left).

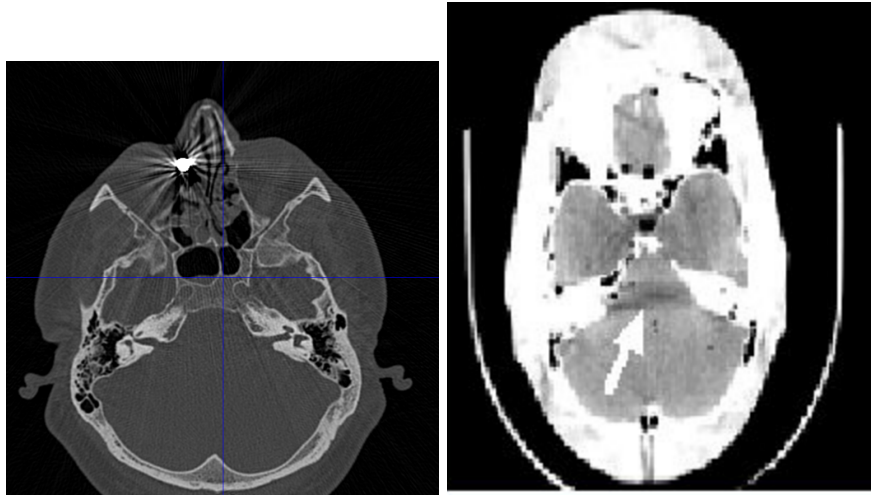


Figure 2.19: CT artifacts: Left, a metallic projectile in the right nose (mirrored CT image) degrade CT image quality in the maxillary region (Image is courtesy of J. Hoffmann of the University Hospital Tübingen); right, beam hardening artifact near the base of the skull (arrow) (Image is courtesy of J. Hsieh, General Electric).

MRI image data is also degraded in the presence of metal, since it causes inhomogeneities in the magnetic field. These inhomogeneities can lead to distortions of the image data (suggesting a somewhat different geometry) and to local changes of signal intensity. This is discussed in more detail in [Hol01].

Another known CT data artifact is beam hardening. This type of artifact usually appears as a dark area behind thick bone structures (Fig. 2.19 right). The reasons for this darkening lies in the uneven attenuation of the X-ray spectrum while passing through thick objects, e.g. bones of the base of the skull.

Other artifacts occur due to patient movement (e.g., breathing, voluntary movements, moving body organs, etc.), where even small movements can lead to distortions to the whole image stack. In most cases, this kind of artifact can be significantly reduced with an increased scanning speed.

Bibliography

- [CJS93] Z. Cho, J. Jones, and M. Singh. *Foundations of Medical Imaging*. John Wiley, New York, NY, 1993.
- [CT05] Multislice CT. Multislice-CT - From Seeing to Understanding. <http://www.multislice-ct.com>, accessed 2005.
- [Fah99] R. Fahrig. *Computed Rotational Angiography*. PhD thesis, University of Western Ontario, 1999.
- [FDK84] L. Feldkamp, L. Davis, and J. Kress. Practical Cone-Beam Algorithm. *Journal of the Optical Society of America*, 1(6):612–619, 1984.
- [Gla95] A. Glassner. *Principles of Digital Image Synthesis - Volume One*. Morgan Kaufmann Publishers, Inc., San Francisco, USA, 1995.
- [Gre05] C. Gregory. Chickscope: MRI Artifact Gallery. <http://chickscope.beckman.uiuc.edu/roosts/carl>, accessed 2005.
- [GSR⁺00] Ö. Gürvit, M. Skalej, R. Riekmann, U. Ernemann, and K. Voigt. Rotational Angiography and 3D Reconstruction in Neuroradiology. *electro medica*, 68(1):31–37, 2000.
- [Hah05] H. Hahn. *Morphological Voumetry: Theory, Concepts, and Application to Quantitative Medical Imaging*. PhD thesis, Department of Mathematics and Computer Science, University of Bremen, 2005.
- [HB86] H. Höhne and R. Bernstein. Shading 3D-Images from CT using Gray-level Gradients. *IEEE Transactions on Medical Imaging*, MI-5(1):45–47, 1986.
- [Hol01] M. Holden. *Registration of Serial MR Brain Images*. PhD thesis, King’s College London, University of London, 2001.
- [Hou72] G. Hounsfield. A Method of and Apperatus for Examination of a Body by Radiation such as X-ray or Gamma Radiation. British Patent No. 1283915, 1972.

- [Hou80] G. Hounsfield. Computed Medical Imaging: Nobel Lecture. *Journal of Computer Assisted Tomography*, 4(5):665–674, 1980.
- [Kal00] W. Kalender. *Computer Tomography*. PUBLICIS MCD Verlag, München, Germany, 2000.
- [LOPR97] T. Lehmann, W. Oberschelp, E. Pelikan, and R. Repges. *Bildverarbeitung für die Medizin: Grundlagen, Modelle, Methoden, Anwendungen*. Springer Verlag, Heidelberg, Germany, 1997.
- [MMMY97] T. Möller, R. Machiraju, K. Mueller, and R. Yagel. A Comparison of Normal Estimation Schemes. In *Proc. of IEEE Visualization*, pages 19–26, 1997.
- [MT05] MR-TIP. Magnetic Resonance Technology Information Portal. <http://www.mr-tip.com>, accessed 2005.
- [NBG⁺01] A. Nabavi, P. Black, D. Gering, C. Westin, V. Mehta, R. Pergolizzi Jr., M. Ferrant, S. Warfield, N. Hata, R. Schwartz, W. Wells, R. Kikinis, and F. Jolesz. Serial Intraoperative Magnetic Resonance Imaging of Brain Shift. *Neurosurgery*, 48(4):787–797, 2001.
- [NCKG00] L. Neumann, B. Csabfalvi, A. König, and E. Gröller. Gradient Estimation in Volume Data using 4D Linear Regression. In *Proc. of Eurographics*, pages 351–357, 2000.
- [NGvKF03] C. Nimsky, O. Ganslandt, B. von Keller, and R. Fahlbusch. Preliminary Experience in Glioma Surgery With Intraoperative High-Field MRI. *Acta Neurochirurgica*, 88:21–29, 2003.
- [Nyq24] H. Nyquist. Certain Factors Affecting Telegraph Speed. *Bell System Technical Journal*, 3(3):324–346, 1924.
- [OFF05] OFFIS e.V. DICOM @ OFFIS. <http://dicom.offis.de>, accessed 2005.
- [RIT⁺01] F. Roux, D. Ibarrola, M. Tremoulet, Y. Lazorthes, P. Henry, J. Sol, and I Berry. Methodological and Technical Issues for Integrating Functional Magnetic Resonance Imaging Data in a Neuronavigational System. *Neurosurgery*, 49(5):1145–56, 2001.
- [Rön95] W. Röntgen. Über eine neue Art von Strahlen (Vorläufige Mittheilung). *Sitzungsberichte der physikalisch-medizinischen Gesellschaft zu Würzburg*, pages 132–141, 1895.

-
- [Sha49] C. Shannon. Communication in the Presence of Noise. *Proc. of Institute of Radio Engineers*, 37(1):10–21, 1949.
- [SML98] W. Schroeder, K. Martin, and W. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 2nd edition, 1998.
- [TOW⁺03] I. Talos, L. O'Donnell, C. Westin, S. Warfield, W. Wells III, and S. Yoo et al. Diffusion Tensor and Functional MRI Fusion with Anatomical MRI for Image-Guided Neurosurgery. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 2879 of *LNCS*, pages 407–415. Springer, 2003.

Part II

Data Enhancement and Filtering

Chapter 3

Volumetric High Dynamic Range Windowing

Dirk Bartz, Benjamin Schnaidt, Jirko Cernik,
Ludwig Gauckler, Jan Fischer, and Angel del Río
Visual Computing for Medicine Group, University of Tübingen
Appeared as WSI Technical Report 2005-03, 2005.

Volume data is usually generated by measuring devices (eg. CT scanners, MRI scanners), mathematical functions (eg., Marschner/Lobb function), or by simulations. While all these sources typically generate 12 bit integer or floating point representations, commonly used displays are only capable of handling 8 bit gray or color levels. In a typical medical scenario, a 3D scanner will generate a 12 bit dataset, from which a subrange of the active full accuracy data range of 0 up to 4096 voxel values will be downsampled to an 8 bit per-voxel accuracy. This downsampling is usually achieved by a linear mapping operation and by clipping of value ranges left and right of the chosen subrange.

In this paper, we propose a novel windowing operation that is based on methods from high dynamic range image mapping. With this method, the contrast of mapped 8 bit volume datasets is significantly enhanced, in particular if the imaging modality allows for a high tissue differentiation (eg., MRI). Henceforth, it also allows better and easier segmentation and classification. We demonstrate the improved contrast with different error metrics and a perception-driven image difference to indicate differences between three different high dynamic range operators.

Keywords: Volume Data, Non-Linear Data Mapping, High Dynamic Range Mapping, Windowing

CR Categories: I.4.3: Enhancement: Filtering, Gray-scale manipulation, Sharpening and deblurring; I.4.7 Feature Measurement: Feature representation; I.4.8 Scene Analysis: Intensity.

3.1 Introduction

There are numerous sources for volumetric data. Among the most frequent ones are measuring devices – such as CT scanners, MRI scanners, etc. –, simulations of natural phenomena, or voxelization and implicit function schemes. Almost all volumetric data sources have in common that they generate voxel data with a float or integer data type. Very rarely, the original data is only an one-byte integer, the common data format of many volume rendering systems. Furthermore, most displays (gray-level or color), as they are used with virtually every computer, provide only an 8 bit data fidelity per color channel. Therefore, the float and integer data types need to be mapped into the smaller data range, if only one color channel (eg., for gray-level representations) is used. This is in particular true in most medical applications, where a high dynamic range display is often prohibitively expensive to be used at every workplace. Up to now, the common remedy is a linear windowing operation, which maps an usually interactively selected data range into 8 bits. This, however, means that the data has been downsampled to fit into 8 bits, or lower and higher data values are clamped. Usually, the data is windowed by a combination of downsampling and clamped, based on standardized settings for specific imaging protocols.

In this paper, we present a new approach for a volume data windowing operator that employs techniques from high dynamic range imaging to map volume data from the original voxel size, usually 12 bits¹, into a target voxel size, usually 8 bit. 12 bit voxels cannot really be considered representing a truly high dynamic range. Nevertheless, we exploit the algorithms to enhance the contrast in the 8 bit representation, but we do not require techniques that specifically address very high intensities. Note that this approach can also be used to increase contrast in native 8 bit datasets (mapping an 8 bit data range into a different 8 bit data range), or in representations where the full data range is lost or not available for other reasons.

In contrast to the original image tone mapping operators, our operator examines the whole volume dataset and is therefore volumetric in nature. Although we have only explored datasets obtained by medical 3D scanners, we believe that this method is equally applicable to other

¹3D medical image scanners usually generate data that requires up to 12 bits per voxel. While the resulting data range of 0 to 4095 (translating CT Hounsfield units into the positive range) is almost fully exploited by CT data, MRI data require only 9-10 bits per voxel.

scalar data types, such as single and double precision floats (of possibly truly high dynamic range), since the algorithm is already based on float numbers.

The operator is based on the Photographic Tone Reproduction operator (PTR) introduced by Reinhard et al. [RSSF02], but required several significant functional modifications for medical imaging. Furthermore, the volumetric nature of medical data required different processing strategies in order to cope with the increased space and time complexity.

In the remainder of the paper, we will discuss related work and introduce Reinhard's PTR operator (Section 3.2). In Section 3.3, we describe our approach and its contributions to extend the PTR operator for volumetric data. Afterwards, we demonstrate the new windowing operator in Section 3.4. Finally, we draw conclusions and provide perspectives for future work.

3.2 Related Work

Manual or automatic selection of an intensity window is a common step in the processing pipeline of medical visualization systems. Windowing is described in [Bar02] and, in more detail, in [Pet03]. Classical intensity windowing consists of selecting maximum and minimum intensities of interest. The values in this intensity range, which is usually defined within the 10 or 12 bit pixel depth provided by typical medical scanners, are then mapped linearly to the gray-scale resolution of the display device. Thus the acquired medical images are normally converted to 8 bit gray-scale images. Only expensive specialized display systems are capable of displaying images with higher pixel depths [Bar03]. Consequently, intensity windowing is described as an integral part of a medical visualization system [KGBS98].

The linear mapping of pixel or voxel intensities from medical scanner measurements to 8 bit gray values often does not produce satisfactory contrast. The intensities of certain types of tissue can be clustered within a very narrow range of values, making them difficult to distinguish on the display. The standard approach to improve contrast in images is histogram equalization [GW92], where the intensities of an image are transformed such that their distance in the intensity range depends on the respective weight in the image histogram. Although this non-linear mapping is based on the image data, it is a global operation and does not take local features into account.

More general mapping strategies and further improvements have been proposed in recent years, in order to maximize the quality obtained when displaying high dynamic range images. In computer graphics, this windowing process is usually known as *tone reproduction*, *tone mapping*, or *high dynamic range image mapping*.

Next to the global tone mapping operators (also known as *spatially*

uniform or *single-scale*), like the previously referenced histogram equalization, one can also find local operators (also known as *spatially varying* or *multi-scale*). While global operators perform the same mapping operation over the whole image, local operators can act distinctly over different regions in the image providing further control over the final result. Since the algorithm that we propose works on a multi-scale basis, we concentrate here on the most recent and relevant methods presented on local tone reproduction operators. A general survey of tone mapping techniques is given in [Dev02].

Since the early work of Chiu et al. [CHS⁺93], the advantage of analyzing different parts of a high dynamic range image has been proven to establish a visually satisfactory tone mapping. Many different local operators for this purpose have been derived afterwards. They exploit the characteristics of the human visual system so that the necessary reduction of the dynamic range and its inherent detriment of accuracy lead to a minimum loss of visible quality. In [FLW02] for instance, an attenuation of large luminance gradients is performed to compress these large gradients, while preserving fine details.

A different approach is presented in [DD02], which is based on the previous work by Tumblin et al. [THG99, TT99]. In this case an edge-preserving filter (*bilateral filter*) is employed to decompose the images into two layers, a base layer and a detail layer. The contrast is reduced in the base layer, which contains the large-scale variations, while visibility is preserved in the detail layer.

Ashikhmin presented a multi-pass strategy for tone reproduction [Ash02]. In a first stage, an estimation of the local adaptation luminance is performed at each point in the image. Then a compression function is applied to the values obtained during the first step in order to adapt to the required dynamic range of the display. In the third and final stage, relevant details are restored by a post-processing pass over the image. Each of these stages can be performed for a voxel at a time, thus the whole algorithm can be computationally combined into one pass.

Another tone mapping operator (PTR) has been proposed by Reinhard et al, in [RSSF02]. Their work is based on a classical method widely employed in photography to correctly reproduce real world high dynamic range scenes, called the Zone System [Ada80]. In this technique, the scene is divided into print zones mapping the values from pure black to white. A subjective middle brightness region of the scene is taken and assigned to the middle gray print zone. Next, luminance readings are taken for light and dark regions to determine the dynamic range of the scene. It must be noted that the term dynamic range is used here in the photographic sense, that is, the ratio of the highest to the lowest luminance regions with visible details. This is not the typical computer graphics meaning as ratio of the highest to the lowest scene luminance. Using this method, it is assured that the maximum possible detail is retained, whenever an appropriate middle gray

is found. Reinhard et al. propose a double pass strategy composed of an initial luminance mapping and a posterior automatic *dodging-and-burning* process. The initial mapping is a spatially uniform transformation where the tonal range of the output image is set in relation to the log-average luminance of the scene and a user-specified value for middle gray. During this mapping, high luminances are compressed. Then, in a second stage a local tone reproduction algorithm that applies *dodging-and-burning* is employed. This technique mimics the photographic *dodging-and-burning* process, where the portions of the print might receive a different exposure time from the negative, controlling the contrast on every region. In a similar way, contrast can be controlled locally over an entire region bounded by large contrast changes. These regions are determined using a center-surround function derived from the brightness perception model by Blommaert et al. [BM90].

3.3 Volumetric High Dynamic Range Windowing

Until now, tone mapping has only been used for two-dimensional images. We present a straightforward, but novel variation of this method suitable to enhance a full volumetric dataset. Since we focus here on gray-level medical images, we will use the terms luminance and intensity interchangeably.

We modified the tone mapping method presented by Reinhard et al. [RSSF02] such that it provides a more comprehensible representation of radiological 2D images. In this course, we have implemented two variations of the original algorithm. The first method performs high dynamic range windowing on the individual slice images of the data volume. The second method is a significantly different approach compared to Reinhard’s original paper and considers the entire volume dataset. Specifically, the Gaussian reconstruction described in Section 3.3.2, Equation 3.5 is evaluated in a 3D voxel neighborhood, in contrast to the 2D voxel neighborhood within the individual slices. Since the volumetric variation performs computationally only somewhat more expensive than the slice-based version, we will only discuss the volumetric variation. The slice-based variation can also be seen as the volumetric variation with a constant z parameter. The visual differences of the volumetric and slice-based approach are shown in Figure 3.4e and f.

In Sections 3.3.1 and 3.3.2, we will give a summary of the modified (but largely similar) intensity mapping (which we call zone intensity mapping) and of the extended volumetric *dodging-and-burning* approach, based on the formulas presented by Reinhard et al. [RSSF02]. Note that while the theoretical basis of the algorithm largely follow Reinhard’s approach – we use a slightly different and easier notation –, our implementation varies significantly to achieve the necessary computational performance. Unless noted otherwise, most of the param-

eter settings are tuned the same way as suggested by Reinhard et al. in [RSSF02]. In particular a , s , α , and ϵ are specified identical, since these values produced good results for medical image data as well. Therefore, we will not specifically mention this fact in the following sections.

3.3.1 Zone Intensity Mapping

In a first step the intensity of an input voxel is scaled, so that a certain mean intensity is achieved. This scaled intensity is obtained by the computation of the log-average intensity \bar{I}_w as shown in Equation 3.1. Please note that not all volumetric datasets use the full 12 bit data range ([0; 4095]). In those cases, the mapping takes only the actively used bit range (eg., 11 bits for [0; 2047]).

$$\bar{I}_w = \exp\left(\frac{1}{N} \cdot \sum_{x,y,z} \log(\delta + I_w(x, y, z))\right) \quad (3.1)$$

In Equation 3.1, I_w denotes the intensity of the original voxel (world intensity). N is the total number of voxels, and δ is a small number used for ensuring that the logarithm in the sum can be computed for possible voxels with intensity 0. The log-average intensity \bar{I}_w is computed over all three-dimensions of the volume and represents the geometric mean of $\delta + I_w(x, y, z)$. In contrast to Reinhard [RSSF02], who focuses on images from photography, we focus on medical scanned data which consists of significantly larger numbers of black (zero) voxels that contain no or only little information. Using a δ close to zero like Reinhard would overemphasize these black voxels, hence we use $\delta = 1.0$ and compensate by subtracting $\delta = 1.0$ from \bar{I}_w , resulting in Equation 3.2:

$$\bar{I}_w = \exp\left(\frac{1}{N} \cdot \sum_{x,y,z} \log(1 + I_w(x, y, z))\right) - 1 \quad (3.2)$$

The voxel intensities are adapted using the calculated log-average intensity and a target key value a . The key value a determines the overall brightness of the resulting image. As suggested in the original article, we have found a key value of 0.18 to produce good results in most cases [RSSF02]. This key value is essentially motivated by the human visual system, which perceives 18% reflectance as *middle gray*. This concept of a “normal key” works also for medical data, since it does not usually exceed 12 bit data values (and hence can be considered neither high nor low key). Each of the voxels is scaled according to Equation 3.3.

$$I(x, y, z) = \frac{a}{\bar{I}_w} \cdot I_w(x, y, z) \quad (3.3)$$

Table 3.1: Computation time (s) of logarithms in Equation 3.2. An upper bound of 0 represents the log-computation for every voxel.

<i>Dataset, Resolution</i>	<i>Time (s) for Upper Bound</i>		
	0	10^{100}	10^{300}
CT Head, 512 x 512 x 324	18.15	4.38	4.12
CT Thorax, 512 x 512 x 168	9.49	2.28	2.14
CT Angiography, 512 x 512 x 194	10.81	2.61	2.44
MRI CISS, 512 x 512 x 54	3.01	0.72	0.69
CT Backpack, 512 x 512 x 373	18.93	4.84	4.31

Using this adaptation, a comparable mean intensity is achieved for any given volume dataset. The scaled voxel intensity $I(x, y, z)$ is used as input for the following steps of the algorithm.

If we use only zone-intensity mapping (without the subsequent *dodging-and-burning*), we use Equation 3.4 with I_{MAX} as the maximum intensity found in the original input volume. This operator provides a general increased intensity and a slightly improved contrast.

$$I_c(x, y, z) = \frac{I(x, y, z) \left(1 + \frac{I(x, y, z)}{I_{MAX}^2}\right)}{1 + I(x, y, z)} \quad (3.4)$$

In terms of computational costs, the computation of the log values for the individual voxels is the single most expensive operation. Fortunately, Equation 3.2 can be reformulated as logarithm of the product of the argument of the logarithms: $\log(a \cdot b \cdot c) = \log(a) + \log(b) + \log(c)$. Albeit, the product over all voxels would quickly pass the accuracy bounds of 64 bit floats, we can still combine a significant number of products, before we are getting close to this bound. We use 10^{100} as bound – which is still significantly below the upper bound of 64 bit floats – and gain a speedup of four to five times compared to the direct implementation of Equation 3.2, as it can be seen in Table 3.1. It also shows that an increase to 10^{300} as upper bound does not significantly reduce the computing time.

3.3.2 Volumetric Dodging-and-Burning

The term *dodging-and-burning* was originally coined in photography and relates to a specific exposure technique that lightens or darkens regions in the final print [Ada83]. While *dodging-and-burning* in photography is largely depending on the individual choice of the photographer, Reinhard’s *dodging-and-burning* is an automatic process. Its purpose is essentially to treat regions with similar contrast (a mostly homogeneous region) identical. In this process, the original algorithm tries to identify these regions by computing the convolution of the intensity with a Gaussian filter of a specific size, which is called the *scale* of the filter. The largest filter that still covers a region, which has a similar contrast, is selected for further processing.

We have devised a volumetric extension of this *dodging-and-burning* operator for the direct application in three-dimensions. Unlike the original algorithm, it does not work on a per-slice basis. Rather, both the filter kernels and the convolutions used for determining the correct scale are defined in 3D.

In contrast to global tone mapping, *dodging-and-burning* works on local regions. As mentioned above, in a first step, the size of this local region is determined for each voxel. This size depends on the area around the voxel in which no large contrast changes occur.

In order to measure the contrast around a given voxel, we follow Reinhard et al., who used a center-surround function based on Blommaert’s model for brightness perception [BM90], which is based on spherical symmetric Gaussian filter kernels:

$$R_i(x, y, z, s) = \exp\left(-\frac{x^2 + y^2 + z^2}{(\alpha s^i)^2}\right) \quad (3.5)$$

Note that $R_i(x, y, z, s)$ is not yet normalized, which is done implicitly during processing later on. Also, we are using a slightly different notation than Reinhard et al. uses in Equations 5-8, which are, however, equivalent, since we iterate s instead of α .

For each of these kernels, the scale at which it is supposed to operate is given as s^i . Here s is the so-called center-surround ratio, which is the factor between two consecutive scales (Note that $s^0 = 1$, $s^1 = 1.6$, $s^2 = 1.6 \cdot 1.6$, etc.). The currently relevant scale is determined by the index i . We have chosen a fixed parameter value of $\alpha = 1/2\sqrt{2} \approx 0.35$ in our experiments to ensure that the smallest scale essentially covers only the voxel itself. Although this value can be changed by the user, this fixed α generated satisfying results for medical datasets.

We can now obtain the three-dimensional convolution of the intensities of the volume data with these Gaussians, as shown in Equation 3.6, whereas we are using only $n \times n \times n$ filter kernel, where n is by default 5.

$$V_i(x, y, z, s) = I(x, y, z) \otimes R_i(x, y, z, s) \quad (3.6)$$

Since Gaussian kernels are separable, we can compute the convolution in three 1D passes, in contrast to one 3D pass [RSSF02, Mit04]. Thus, the computational complexity per-voxel is $O(n + n + n)$, in contrast to $O(n \times n \times n)$. This results in only a 50% increase of the computational costs – going from the slice-based dodging and burning to the volumetric version –, instead of n -times higher.

The computation of the 3D convolutions requires accessing voxel data from different slices. This significantly degrades memory access efficiency, resulting in considerably longer processing times. Unfortunately, not all participating slices fit into the L1-cache for fast processing. However, the separated convolution scheme as described above

allows a much more compact memory representation. This way, the actual convolution fits easily into the L1-cache.

Please note that the frequency-based convolution – suggested by Reinhard et al. as convolution approach – has various other drawbacks compared to our truncated Gaussian filtering, since it requires the full (Fast) Fourier Transformation (FFT) of the data volume into frequency space. Furthermore, it would require this transformed data volume at several scales i , which would be prohibitive expensive for volumetric data, not only in computational costs, but also in memory requirements. Nevertheless, we tested this frequency space-based convolution and compared the voxel differences to our $n \times n \times n$ Gaussian filter convolution, with $n = 5, 11, 21$. The differences for $n = 5$ were largely negligible and for $n = 21$ not detectable. A similar result can be expected, if we compare the Gaussian filter matrix with a full-volume (full-image in 2D) sized filter. Therefore, we consider this matrix based convolution as an acceptable and significantly cheaper solution.

Now the center-surround function can be evaluated, which gives a measure for the activity (dynamics) at a given scale around the voxel (see Equation 3.7). Note that the center-surround ratio s is here already considered a (user specifiable) constant.

$$act_i(x, y, z) = \frac{V_{i-1}(x, y, z, s) - V_i(x, y, z, s)}{2^\phi a / (s^{i-1})^2 + V_{i-1}(x, y, z, s)} \quad (3.7)$$

The algorithm stops at the first scale index i for which $|act_i(x, y, z)|$ is larger than $\epsilon = 0.05$, and picks the previous scale index $i - 1$. Choosing the respective convolution result V_{i-1} , a new intensity for the given voxel can be computed according to Equation 3.8. The operator specified in this equation overall provides a compression of high intensities. While this is not necessary for most medical image data, it can be used to compensate very high intensities caused by metal artifacts in CT data (eg., by metal tooth fillings in Fig. 3.1).

$$I_c(x, y, z) = \frac{I(x, y, z)}{1 + V_{i-1}(x, y, z, s)} \quad (3.8)$$

A further improvement of the intensity mapping in Equation 3.8 is achieved by adapting the computed output I_c to the maximum intensity (I_{MAX}) that is present in the volume (Equation 3.9), similar to Equation 3.4 of the zone intensity mapping.

$$I_c(x, y, z) = \frac{I(x, y, z) \left(1 + \frac{I(x, y, z)}{I_{MAX}^2}\right)}{1 + V_{i-1}(x, y, z, s)} \quad (3.9)$$

Since $I_c(x, y, z)$ generates intensities in the $[0; 1]$ interval, it must be scaled into the target 8 bit range by scaling (and truncating) it with $(2^b - 1)$ with $b = 8$.

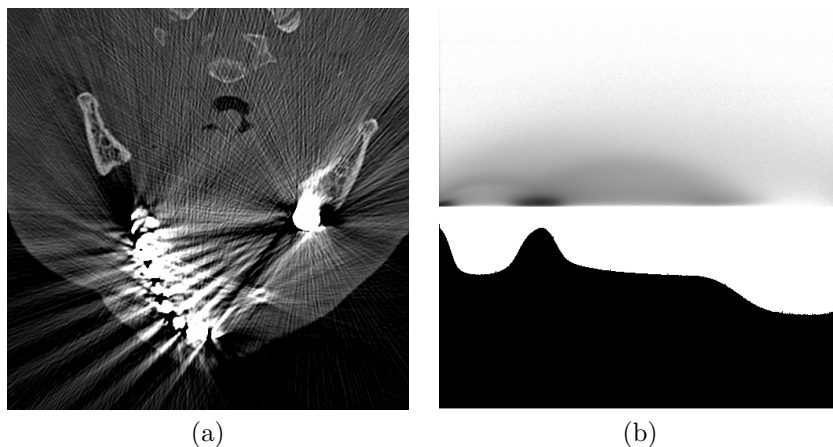


Figure 3.1: Metal artifacts in CT Head dataset. (a) Linearly mapped slice, (b) y-log-scale gradient magnitude (top) and voxel value histogram (bottom) of slice. The high intensities of the metal artifacts can be seen on the far right end of the histogram.

The described method is a local tone mapping operator, because the scale index i that is used for the actual mapping computation in Equation 3.9 is computed separately *for every voxel* and the respective filter has a limited local support. This has of course consequences for the data analysis, classification, and rendering, that we are going to discuss in Section 3.4.3.

3.4 Results

The major achievement of volumetric high dynamic range (VHDR) windowing is a contrast enhancement in comparison to a standard linear windowing operation. In this section, we will discuss the use of the new windowing operators. Although we implemented also a slice-based variation of this operator, we will focus only on the volumetric version. While the computational costs per-voxel of the slice-based version are about 30% lower than of the volumetric variation ($O(n + n + n)$ vs. $O(n + n)$), the latter takes into account the full spatial information of the voxels covered by the convolution filter in the same slice, and in the neighboring $(n - 1)/2$ slices in both slice directions.

In the course of this section, we will discuss Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) as two different imaging modalities. Unfortunately, no ultrasound or PET datasets were available at the time of the experiments. Hence, we cannot provide information on these modalities.

3.4.1 Computed Tomography

Figure 3.6a-c shows the same slice from a CT scan of a thorax, showing the lungs, the heart, the trachea, parts of the backbone, the ribs,

and the sternum. In Figure 3.6a, the full active bit range of 12 bits is mapped linearly to 8 bits. However, the full data range is not used by the CT dataset; all voxels are located in the data value range of $[0; 2700]$ (or $[-1023; 1677]$ Hounsfield units (HU)). Furthermore, an upper interval of 1000 HU does not contain useful information. If we add a clip operation to remove unused voxel value ranges at the windowing operation, the upper end (in this case) of the new value range will show a significant increase in the histogram, suggesting that there is a material interface or another feature.

In contrast, VHDR windowing provides an adaptive correction of the voxel values (see Fig. 3.6b), depending on the overall intensity of the image and a correction that takes into account the local neighborhood (*dodging-and-burning*, see Section 3.3.2). At first glance, we can see that the overall image is brighter than the one with linear mapping. Pixel by pixel comparison actually revealed that the contrast between darker tissue areas and brighter bone areas increased. Figure 3.6c exposes this difference between linear and the VHDR windowing operator. While there is a general shift of intensity, higher differences are in areas with a high contrast (eg., between bone and dark tissue). This is of course what could be expected, since these areas are exactly those which are supposed to be enhanced by the method.

The second examined CT dataset is a scan of a backpack filled with a variety of objects. The dataset occupies the full 12 bit range of $[0; 4095]$ (or $[-1023; 3071]$ Hounsfield units (HU)), whereas the metal artifacts caused by some of the objects ensure a rather dark intensity distribution for most objects (Fig. 3.6j). With VHDR windowing, the intensity contrast is significantly improved and considerably more structures – which were virtually invisible before – become visible (Fig. 3.6k). Figure 3.6l shows the voxels of one slice where the intensity changed most by our operator. Note that for this VHDR windowing, the key value a has been changed to 0.05 (default is $a = 0.18$).

3.4.2 Magnetic Resonance Imaging

Our second examined modality is MRI. In contrast to CT, MRI provides a whole zoo of different image acquisition protocols that generate image data focusing on different organs and tissue types. Furthermore, MRI generally provides more material (or tissue) differentiation than CT, thus making it an imaging modality with a good potential for contrast improvements. However, the different MRI image acquisition protocols generate data with limited 9-10 bits per voxel accuracy. Therefore, the data has a smaller dynamic range than datasets from CT.

In our first example, we examine a 3D CISS MRI imaging sequence, which is a T2-weighted protocol that emphasizes on the cerebro spinal fluid (CSF) filled cavities of the head. In contrast to our previous CT dataset, this MRI dataset has only 10 active bits and only the active

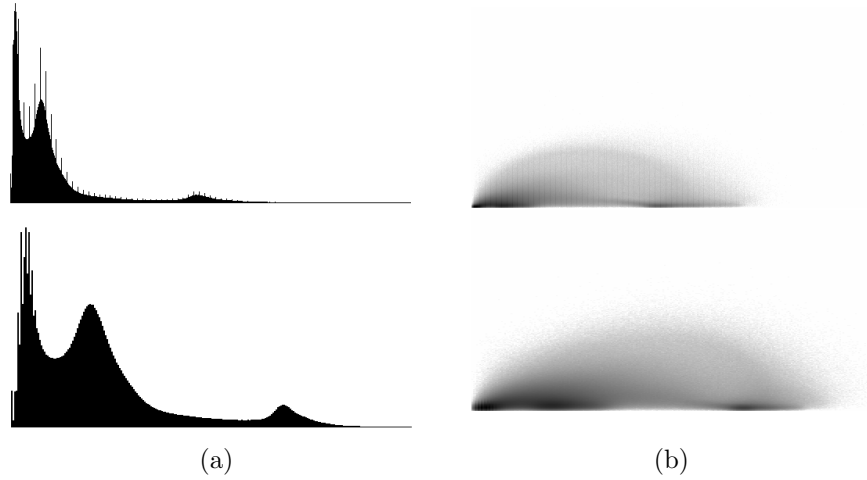


Figure 3.2: Histograms of 3D CISS MRI scan of a head. (a) shows the standard histogram of the (stretched) linear (top) and VHDR windowing (bottom) result and (b) shows the respective gradient magnitude histograms. The windowing results are shown in Figure 3.6d-e.

value interval of $[0; 850]$. Hence, we only map 10 bits to 8 bits for all windowing methods (Fig. 3.6d-e).

Figure 3.2 shows the standard (voxel value) and gradient magnitude histograms. We stretched the histograms of the linear windowing (top histograms) to the full range, thus allowing an easier comparison. Overall, we can observe that the histograms of the VHDR windowing (bottom) are less affected by a small number of high intensity noise values (upper/right ends of linear histograms) and relevant regions have a smoother development of histogram values. In particular peaks are more spread out, thus enabling a better data analysis, segmentation, or classification of the data.

Also with the 3D CISS MRI sequence, the contrast differences are quite obvious between VHDR windowing and the linear windowing (Fig. 3.6f). Beside the general brightness shift between the dark background and the head, edges are again emphasized, since they represent high contrast differences. We can also observe the enhanced intensity of the soft brain tissue (gray and white matter) after VHDR windowing.

In our last example, we examine another MRI dataset. This dataset was acquired by T1-weighted flash sequence that mapped CSF-filled cavities to black (Fig. 3.6g-i). Similar to the previous 3D CISS MRI dataset, only 9 bits are active in a voxel value interval of $[0; 350]$, hence only 9 bits are mapped into 8 bits by all windowing operators (Fig. 3.6g-i).

Again, we can observe a general brightness shift between the VHDR and linear windowing, and in addition a contrast enhancement on the edges and the enhanced soft brain tissue (Fig. 3.6h) intensity.

An interesting application of VHDR is contrast enhancement in low-

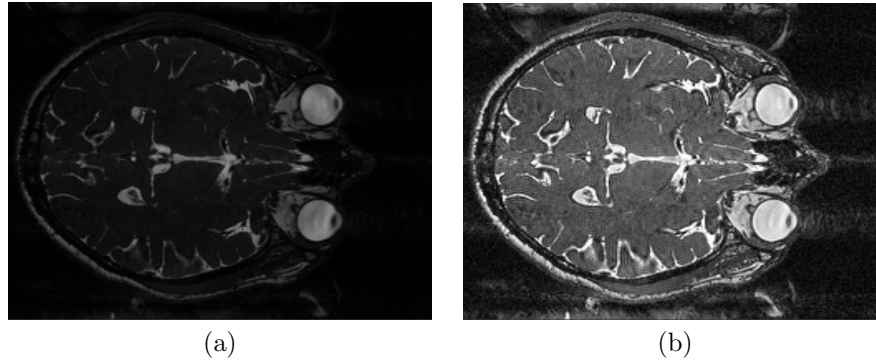


Figure 3.3: Example image slice from MRI CISS dataset. (a) Original 8 bit image slice, (b) VHDR enhanced 8 bit image slice.

fidelity datasets. Figure 3.3 demonstrates an example, where another 3D CISS MRI sequence dataset is only available in 8 bit fidelity. Due to the low voxel fidelity, the appearance of the MRI image slice is very dark (Fig. 3.3a). After VHDR windowing (from 8 bit to 8 bit), the contrast is significantly enhanced (Fig. 3.3b), while the noise level is kept at a much lower level than with histogram equalization (see Fig. 3.5 for the MRI CISS example).

3.4.3 Discussion

In the past two sections, we presented the application of volumetric high dynamic range (VHDR) windowing to three different volumetric datasets of different modalities. We could observe a general contrast enhancement with VHDR windowing methods, compared to linear windowing. At the same time however, VHDR windowing can also increase the noise level, if high-intensity noise is present, eg., with metal artifacts in CT data. For most practical aspects though, this increased artifact noise is negligible.

The increased contrast quality does not come for free; the full accuracy datasets need to be mapped using the linear or VHDR windowing operations in a pre-process. Not surprisingly, the required time depends on the size of the dataset and on the chosen windowing methods. In our experiments (see Table 3.2) with the various datasets discussed above, we measured timings between less than a second for linear windowing of the smallest datasets and up to 32 seconds for the full volumetric HDR windowing of the largest dataset. All measurements were performed on a PC, equipped with 2GB of main memory and an Intel P4 CPU running at 3GHz. We consider the increased computational costs for VHDR windowing as modest and easily affordable for the increased quality.

The VHDR windowing operators logically perform two processing passes to enhance the contrast in the datasets (in the actual implementation, both passes are integrated into one processing pass). While

Table 3.2: Timings for the various windowing methods on five different volume datasets.

<i>Dataset</i>	<i>Windowing Method</i>	<i>Time (s)</i>
CT Head 512 x 512 x 324	Linear	2.94
	Zone Intensity	4.27
	VHDR Windowing	30.49
CT Thorax 512 x 512 x 168	Linear	0.17
	Zone Intensity	0.21
	VHDR Windowing	16.31
CT Angiography 512 x 512 x 194	Linear	0.57
	Zone Intensity	0.98
	VHDR Windowing	12.49
MRI CISS 512 x 512 x 54	Linear	0.06
	Zone Intensity	0.09
	VHDR Windowing	7.09
CT Backpack 512 x 512 x 373	Linear	1.06
	Zone Intensity	2.07
	VHDR Windowing	31.95

the first pass uses a global, logarithmic scaling of the datasets – thus transforming the histogram of the data –, the second pass applies a filter of the local voxel region around every voxel. This however, takes into account the local spatial neighborhood of the dataset and not only their voxel values, resulting in a spatially varying operation. As a consequence, previously available mapping methods based on the voxel value (eg., transfer functions) will affect a different set of voxels. Fortunately, the number of these voxels is small and they almost always belong to an edge area with a now increased contrast. Most changes have been introduced by the global zone intensity mapping, thus the transfer functions – based on the original histogram – can be adapted to the new voxel values, according to the transformation of the histogram. The changes introduced by the local *dodging-and-burning* are limited to areas with high contrast (edges) and only emphasize the material interfaces (see Fig. 3.4a-d). Therefore, the consequences for a previously known classification are manageable.

Figure 3.4c-f shows the visual differences based on a visual difference predictor (VDP). Figure 3.4e-f also indicates the differences between the slice-based and the volumetric HDR windowing operator.

We also compared the performance of VHDR windowing with other high dynamic range operators. In particular, we also tested Ashikhmin’s tone reproduction operator [Ash02] and Durand et al.’s operator that is based on bilateral filtering [DD02]. As it turns out, both operators are less suited for windowing than Reinhard et al.’s PTR operator [RSSF02]. Both operators in particular focus on high dynamic range images with high image luminances, which are not really present in medical image data. Figure 3.5c and d shows absolute intensity differences mapped to a hue colormap (blue no difference, red large

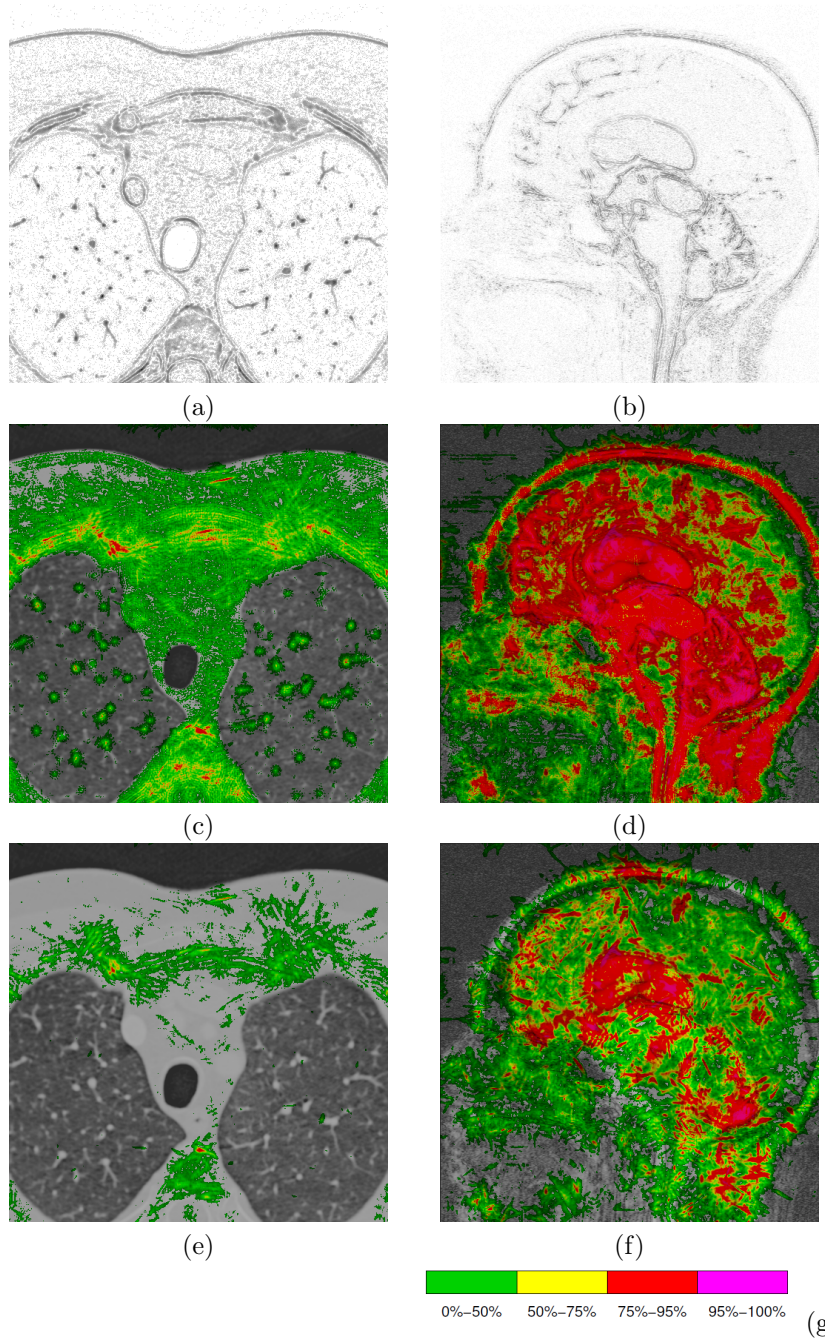


Figure 3.4: Visual differences between different windowing operators for CT Thorax (a,c,e) and 3D CISS MRI (b,d,f) datasets. The first row shows the relative intensity between zone intensity mapping and VHDR windowing (images are darkened for printing contrast). The other rows show a perceived intensity difference [MMS04] between zone intensity mapping and VHDR windowing (2nd row), and between slice-based HDR and VHDR windowing (3rd row) based on a Visual Difference Predictor [MMS04]. (g) shows its color-coded scale of the detection probability.

differences). Most differences are located on the contrast edges of the dataset, with some differences also in the higher intensity areas of the

CSF-filled ventricular system. This is not very surprising for Durand’s approach, since it is based on edge-preserving bilateral filtering. Our experience indicates that an operator which focuses more on the intermediate luminance values, like PTR, performs better for our application. Furthermore, Ashikhmin’s operator expects real luminance values for processing which cannot be substituted by the intensity values of the voxels, as it can be done with PTR. Finally, both operators are more costly to compute. Especially Durand’s bilateral filtering requires many costly *exp* operations, which considerably slow the algorithm down.

3.4.4 Error Metrics

An important question is how we can measure if our VHDR windowing method is actually improving the image quality compared to the traditional linear windowing. The question which metric to use is quite complex. While most high dynamic range mapping methods can be evaluated by esthetics (“Is the mapped image more pleasing? Does it convey the information?”) or psycho-physical metrics, this is not straight-forward for volumetric medical imaging datasets. Linear windowing translates the intensities of the original data directly into the lower range. A simple error metric based on the normalized intensities would not detect contrast improvements achieved by our algorithms. Therefore, we decided to use an entropy-based metric, assuming that VHDR windowing will maintain more information of the original data than linear windowing. The entropy itself is calculated by $H = -\sum_i (p_i \cdot \log_2(p_i))$, with p_i as the probability of intensity i in a voxel.

Table 3.3 shows the results of the measured entropy of five datasets with our three different windowing methods, compared to the original volume dataset. For all three datasets, the entropy is monotonically increasing with the quality of the windowing method. Generally, the entropy of an image would be maximized in a completely (histogram) equalized dataset, but this would in particular enhance the low intensity background noise, since a large number of background voxels occupies only a small number of low intensity histogram buckets. Figure 3.5a and b shows a histogram equalized slice from the MRI CISS dataset with such an increased noise level, demonstrating why it is not suitable for the enhancement of scanned volume data. This example also clearly shows that the entropy-based metric is not a perfectly suitable metric. VHDR does some kind of equalization of the voxel values, since it redistributes the intensity values to enhance the intensity range representation.

As an alternative, we also adopted a contrast measuring metric that is used in multimedia texture analysis [Ste00]. This contrast metric, described in Equation 3.10, represents the overall contrast in a data volume. $g[i, j]$ represents the gray-level co-occurrence matrix, which

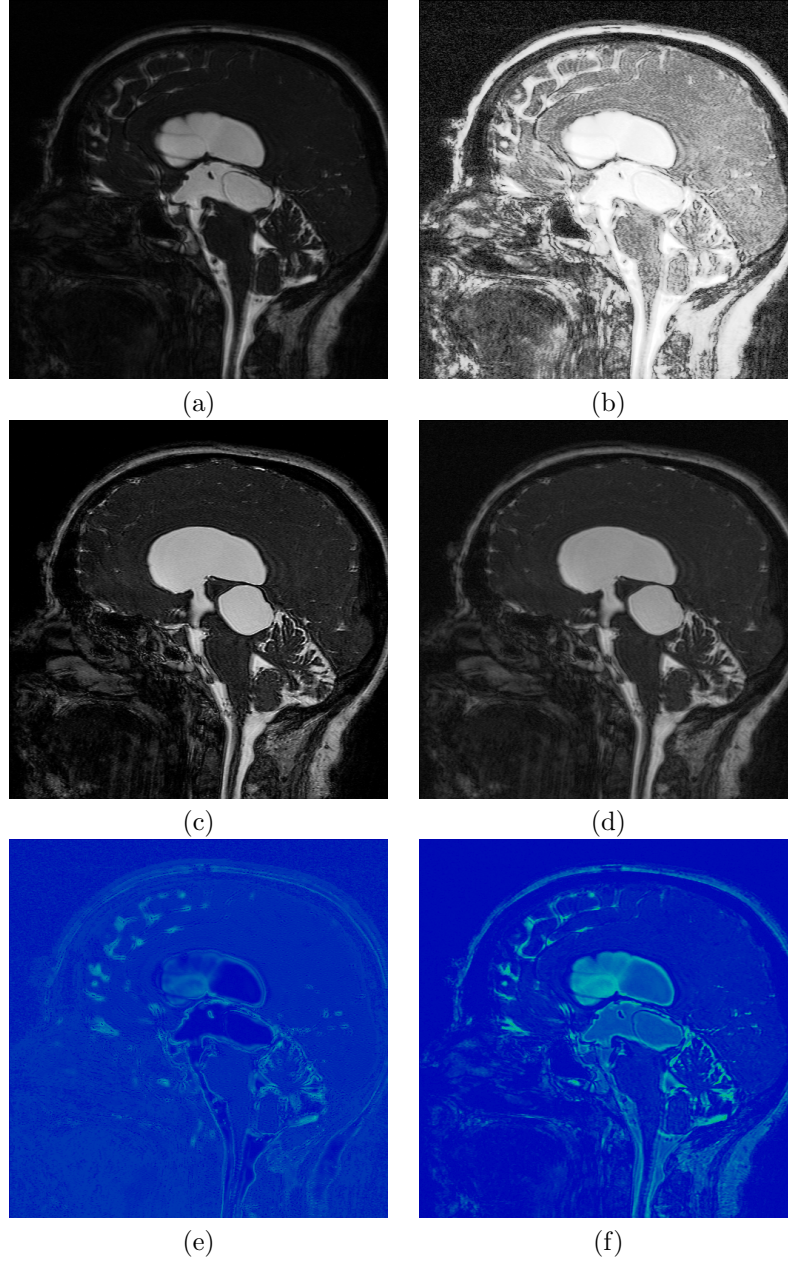


Figure 3.5: Other windowing operator applied to the MRI CISS dataset. (a) Original image slice, (b) histogram equalized image slice with increased background noise, (c) image slice after Ashikhmin's tone reproduction, and (d) after Durand's bilateral filtering. (e) and (f) show the absolute intensity difference between VHDR windowing (Fig. 3.6e) and after Ashikhmin's tone reproduction operator (e) and after Durand's bilateral filtering (f).

enumerates the voxels with value i that have a neighbor with voxel value j , with H as the maximum voxel value of the respective histogram (eg., 255 for 8 bit datasets), and N as the number of voxels. Note, that $g[i, j] \neq g[j, i]$, since different voxels have different neighborhoods. Table 3.3 provides also the results of the contrast metric for

Table 3.3: Computed entropy for all windowing methods.

<i>Dataset</i>	<i>Wind' Method</i>	<i>Entropy H</i>	<i>Contrast C</i>
CT Head 512 x 512 x 324	Original 12 bits	9.81	30331.20
	Linear	5.97	117.58
	Zone Intensity	6.33	148.94
	VHDR Wind'	6.52	266.77
CT Thorax 512 x 512 x 168	Original 12 bits	9.76	2772.43
	Linear	5.76	10.92
	Zone Intensity	6.33	21.28
	VHDR Wind'	6.35	25.34
CT Angiography 512 x 512 x 194	Original 12 bits	9.42	5126.89
	Linear	5.41	20.01
	Zone Intensity	6.29	30.98
	VHDR Wind'	6.35	43.69
MRI CISS 512 x 512 x 54	Original 10 bits	8.01	921.27
	Linear	5.02	14.50
	Zone Intensity	6.92	123.91
	VHDR Wind'	7.00	208.55
CT Backpack 512 x 512 x 373	Original 12 bits	4.55	8637.52
	Linear	2.43	33.44
	Zone Intensity	4.15	584.02
	VHDR Wind'	4.31	881.05

the different datasets. The significantly higher contrast values for the original dataset are due to the higher voxel depth of 12 bits (10 bits).

$$C = \sum_{i=0..H} \sum_{j=0..H} \frac{(i-j)^2 g[i, j]}{N} \quad (3.10)$$

Similar to the entropy metric, the calculated contrast C for the examined datasets is always significantly higher for the VHDR windowing methods, indicating an improved contrast.

3.5 Conclusion and Future Work

In this paper, we presented a new windowing operator that is based on a recently introduced tone mapping method. This operator can map any data item size (eg., 12 bit integer values) into any other data item size (eg., 8 bit integer values) and optimizes its range representation to enhance contrast. The operator can also be used to map 8 bit data into an 8 bit data range, in order to enhance the intensity contrast.

We evaluated the performance of our operator with an entropy metric and with a contrast-based metric from video-encoding. Both metrics clearly showed a significant improvement with our operator. Furthermore, we employed a perception-driven visual difference predictor to expose areas of perceptually largest differences between different HDR operators.

Still, more research needs to be done to consider psycho-physical

aspects of the human visual systems into an error metric. Another aspect of future work will look into the application of VHDR windowing for other data domains in medicine (eg., DTI datasets) or scientific visualization (eg., CFD). We believe that the methodology can also help here to map high dynamic range data from computational sciences into a range that is more feasible for many volume rendering systems with lower per-voxel accuracy.

Acknowledgments

This work has been supported by the Department for Neurosurgery of the University Hospital Tübingen and by the DFG project VIRTUE. Some datasets are courtesy of the Department of Radiology of the University Hospital Tübingen. The backpack dataset is courtesy of Kevin Kreeger and was downloaded from www.volvis.org. We thank Rafal Mantiuk of MPI Saarbrücken for providing us with his visual difference predicting software.

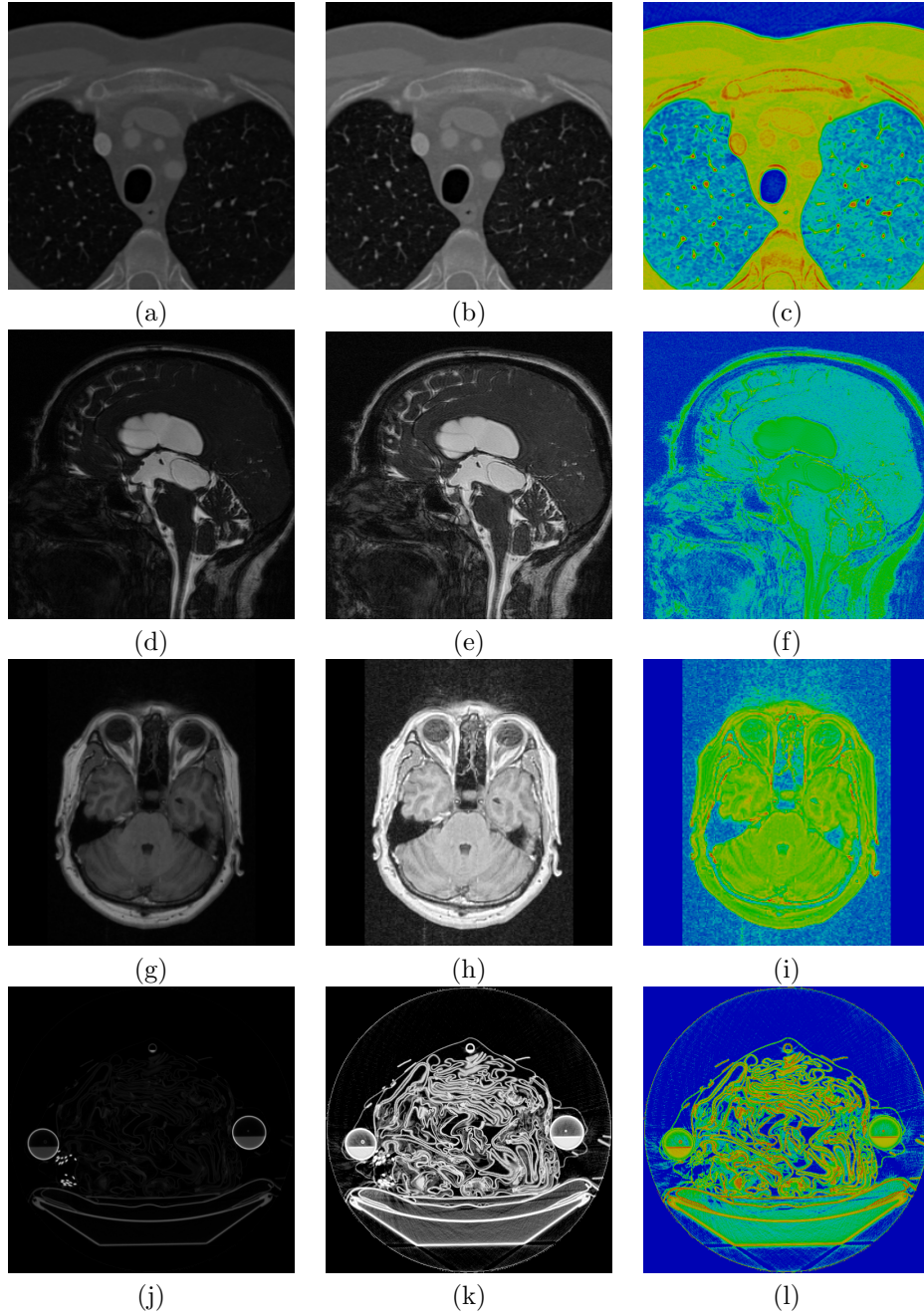


Figure 3.6: Dataset overview: The left column shows the result from the linear windowing (a,d,g,j), the middle columns shows the results from the volumetric high dynamic range (VHDR) windowing (b,e,h,k), and the right column shows the relative voxel intensity difference between linear and VHDR windowing (c,f,i,l), where the occurring difference is mapped to the full hue range (blue no difference, red largest difference). The top row shows the CT thorax, the second row shows the MRI CISS dataset, the third row shows a dataset from an MRI flash dataset, and the bottom row shows results for the CT backpack dataset. The actual slice size of the flash MRI dataset is smaller, than the image size in Fig. 3.6g-i. This can be seen on the left and right side of the images.

Bibliography

- [Ada80] A. Adams. The Camera. *The Ansel Adams Photography Series*, 1980. Little, Brown and Company.
- [Ada83] A. Adams. The Print. *The Ansel Adams Photography series*, 1983. Little, Brown and Company.
- [Ash02] M. Ashikhmin. A Tone Mapping Algorithm for High Contrast Images. In *Proc. of Eurographics Workshop on Rendering*, pages 134–156, 2002.
- [Bar02] D. Bartz. Advanced Virtual Medicine: Techniques and Applications for Virtual Endoscopy. In *ACM SIGGRAPH Course 52*, 2002.
- [Bar03] Barco. MeDis Grayscale CRT Display Systems Product Brochure. Barco, 2003.
- [BM90] F. Blommaert and J. Martens. An Object-Oriented Model for Brightness Perception. *Spatial Vision*, 5(1):15–41, 1990.
- [CHS⁺93] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman. Spatially Nonuniform Scaling Functions for High Contrast Images. In *Proc. of Graphics Interface*, pages 245–253, 1993.
- [DD02] F. Durand and J. Dorsey. Fast Bilateral Filtering for the Display of High Dynamic Range Image. In *Proc. of ACM SIGGRAPH*, pages 257–265, 2002.
- [Dev02] K. Devlin. A Review of Tone Reproduction Techniques. Technical Report 117, Department of Computer Science, University of Bristol, 2002.
- [FLW02] R. Fattal, D. Lischinski, and M. Werman. Gradient Domain High Dynamic Range Compression. In *Proc. of ACM SIGGRAPH*, pages 249–256, 2002.
- [GW92] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, Reading, 1992.

- [KGBS98] A. Kugler, T. Grunert, E. Becker, and W. Straßer. MED-Station: Bringing Together Medical Imaging and Diagnosis. In *Proc. of European Multimedia, Microprocessor Systems and Electronic Commerce*, 1998.
- [Mit04] J. Mitchell. Real-Time 3D Scene Post-Processing. Game Developer Conference 2003: http://www.ati.com/developer/gdc/GDC2003_ScenePost-processing.pdf, accessed 2004.
- [MMS04] R. Mantiuk, K. Myszkowski, and H. Seidel. Visible Difference Predictor for High Dynamic Range Images. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pages 2763–2769, 2004.
- [Pet03] H. Petterson. Window (The Encyclopaedia of Medical Imaging website). <http://www.amershamhealth.com/medencyclopaedia/medical>, accessed 2003.
- [RSSF02] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic Tone Reproduction for Digital Images. In *Proc. of ACM SIGGRAPH*, pages 267–276, 2002.
- [Ste00] R. Steinmetz. *Multimedia Technologie - Grundlagen, Komponenten und Systeme*. Springer-Verlag, Heidelberg, 3rd edition, 2000.
- [THG99] J. Tumblin, J. Hodgins, and B. Guenter. Two Methods for Display of High Contrast Images. *ACM Transactions on Graphics*, 1(18):56–94, 1999.
- [TT99] J. Tumblin and G. Turk. LCIS: A Boundary Hierarchy for Detail-Preserving Contrast Reduction. In *Proc. of ACM SIGGRAPH*, pages 83–90, 1999.

Part III

Analysis and Classification

Chapter 4

SignatureSpace: A Multidimensional, Exploratory Approach for the Analysis of Volume Data

Kith Pradhan², Dirk Bartz¹, and Klaus Mueller²

¹Visual Computing for Medicine Group, University of Tübingen

²Center for Visual Computing, Stony Brook University

Appeared as WSI Technical Report 2005-11, 2005.

The analysis of volumetric data is a crucial part in the visualization pipeline, since it determines the features in a volume dataset and henceforth, also its rendering parameters. Unfortunately, volume analysis can also be a very tedious and difficult challenge.

To cope with this challenge, this paper describes a novel information visualization driven, explorative approach that allows users to perform an analysis in a comprehensive fashion. From the original *data volume*, a variety of auxiliary data volumes, the *signature volumes*, are computed, which are based on intensity, gradients, and various other statistical metrics. Each of these signature volumes (or signatures in short) is then unified into a multi-dimensional *signature space* to create a comprehensive scope for the analysis. A mosaic of visualization techniques ranging from parallel coordinates, to colormaps and opacity modulation, is available to provide insight into the structure and feature distribution of the volume dataset, and thus enables a specification of complex multi-dimensional transfer functions and segmentations.

Keywords: Volume Data, Parallel Coordinates, Multi-Dimensional Visualization

CR Categories: I.3.3 Picture/Image Generation: Display algorithms; I.3.6 Methodology and Techniques: Graphics data structures and data types; Interaction techniques; I.3.7 Three-Dimensional Graphics and Realism: Color, shading, shadowing, and texture; I.4.6 Segmentation: Classification, Edge and feature detection; Partitioning; I.4.7 Feature Measurement: Feature representation;

4.1 Introduction

Volumetric data is a data concept for discrete data that is arranged on a grid structure and which can be from numerous sources. Most frequently, volumetric data is generated by 3D scanning devices (usually medical 3D scanners such as CT scanners, MRI scanners, etc.) or computed by simulations in physics, chemistry, biology, material sciences, or other engineering fields. If sufficient knowledge on the data is available, this information can be used to derive feature extraction parameters, segmentations, or the specification of transfer functions. Unfortunately, this process is not straightforward and often requires a time-consuming trial-and-error approach [PLB⁺01].

In most (semi-) automatic approaches to generate transfer functions, gradients of the respective voxels are computed and those with maximum gradient values (and the second derivatives of the voxel values) are considered as points of transition between two different materials [BPS97, KD98]. These approaches are basically motivated by a material boundary model in which the gradient magnitude value – usually implemented as the absolute value of the central difference at local voxels – is (almost) zero along the boundary and large across the boundary. Another effect that interferes with voxels along the boundaries is the partial volume effect in medical scanned data, which at the same time ensures a smooth gradient [HB86].

Bajaj et al. used next to gradients also the length and surface of a contour (isosurface), and the included volume confined by the isosurface into the contour spectrum. A super-imposed contour tree was also providing a more global view of changes in the contour spectrum [BPS97]. In contrast to Kindlmann and Durkin [KD98], they did not assume a specific material boundary model, since it is more of an exploratory tool. In that sense, their approach is somewhat similar to our approach (Sections 4.2 and 4.3). Fujishiro et al. [FAT99] use a Reeb-graph data-structure to model the topology of *critical isosurfaces* similar to the contour tree. Pekar et al. [PWH01] also used a similar approach, where they combined gradient, Laplace-filtered voxel values (essentially using the second derivative), volume and surface area confined by the potential isosurfaces into the total gradient curve, and – by

dividing the total gradient by the surface area – into the mean gradient curve.

Tenginakai et al. proposed local higher order moments and derived metrics such as skew and kurtosis to mark areas of material transition for the specification of isovalues [TLM01, TM02]. Again, the authors assumed a local material boundary model that no more than two materials interface with each other in a $w \times w \times w$ neighborhood.

He et al. took a stochastic/interaction-based approach to generate transfer functions. Piecewise function patterns were combined by a stochastic (genetic) algorithm and the results are evaluated by a human observer to guide the stochastic search process [HHKP96]. Another approach that combined interaction with a stochastic algorithm was proposed by Tzeng et al., [TLM03] where interactive painting operations are used to mark areas of interest, or areas to be omitted from the final specification. An artificial neural-network algorithm is used as machine-learning environment to derive rules which voxels should be included. In their paper, they also consider a multidimensional classification space, where they look into the voxel value, the gradient magnitude, the neighboring voxel values, and their position in the volume dataset through the neural-network.

Most methods aiming at the (semi-) automatic transfer function generation deal with some sort of gradient magnitude analysis and often also take into account a Laplacian or the second derivative of the voxel values. However, there are situations where gradient-based methods break down, because they do not depict the material interface alone. These situations are similar to settings where the intensity alone does not specify the materials precisely enough, eg., in the tooth dataset of the transfer-function bake-off [PLB⁺01], or when there are problems due to noise sensitivity.

Our observation is that most classification methods can be applied to a variety of datasets from different sources, but they rarely succeed with all datasets, since they usually focus on a few metrics. In contrast, our approach takes into account a manifold of metrics – which we call signatures –, such as intensity, multiple derived (eg., gradient, second derivatives, etc), and statistical metrics (higher order moments), and derived statistical quantities (skew and kurtosis) [TLM01]. We also inspect the directions along the gradients, providing some kind of history of the local voxels¹. Furthermore, any meaningful signature that has not (yet) been integrated into our variety of signatures, can easily be added, since none is specifically favored.

All signatures are presented in a modified parallel coordinate display, which provides a framework to identify meaningful signatures for a specific dataset. In that sense, we combine methods from information visualization and scientific (volume) visualization into the *SignatureSpace*, a notion that has been introduced with the SimVis system of Doleisch

¹A similar metric has also been explored in parallel by Lum and Ma [LM04].

et al. [DGH03]. The WEAVE system is another system, published even earlier, that combines a 4D visualization of a heart simulation with parallel coordinates and scatter plots [GRW⁺00]. Also a somewhat related approach has been presented by Tory et al. [TPM05], where parallel coordinates are used to represent visualization parameters, such as light parameters, viewing, and transfer functions.

The remaining paper is organized as follows. We start by presenting our modified parallel coordinate-based display, which is in particular suited for a large number of data samples – a common drawback of parallel coordinates – in Section 4.2. Section 4.3 focuses on the different signatures that we are using for the data analysis and we discuss results in Section 4.4. Finally, we draw conclusions and discuss future work in Section 4.5.

4.2 Parallel Coordinates for Volumetric Datasets

Parallel coordinates were introduced into data visualization by Inselberg et al. [Ins85, ID90]. In contrast to traditional coordinate systems like the cartesian coordinate system, the data axes are arranged parallel to each other, enabling a representation for many more dimensions than 3D and 4D cartesian systems. Data values are plotted as polylines over the various coordinate axes, depending on their respective values. However, parallel coordinates are afflicted by two major drawbacks. First, correlations between different dimensions can only be easily detected by an observer, if the respective data axes are located next to each other. Second, while the number of dimensions can be quite high, parallel coordinates are not very well suited for a large number of data samples, since they tend to obscure the view (Fig.4.1a).

A number of techniques have been proposed to address these problems, such as hierarchical parallel coordinates [FWR99] and brushing functionality [MW95]. Most important in our view is the application of colormaps to a specific data axis (Fig.4.1b), a technique which is quite well established in Information Visualization [GKW02] and respective systems, such as the XmdvTool [War94], but that we could not trace back to a specific research paper.

In particular Fua et al. [FWR99] combined the hierarchical display of data clusters with colormaps. The samples of the child dimensions in the nested cluster tree are rendered in a *variable-width opacity band*, where the opacity is decreasing the farther out of the cluster center (or the parent node in the tree) the data item is located. Miller and Wegman proposed density plots for modeling line opacity (alternatively line color saturation) to overcome display obstruction [MW91].

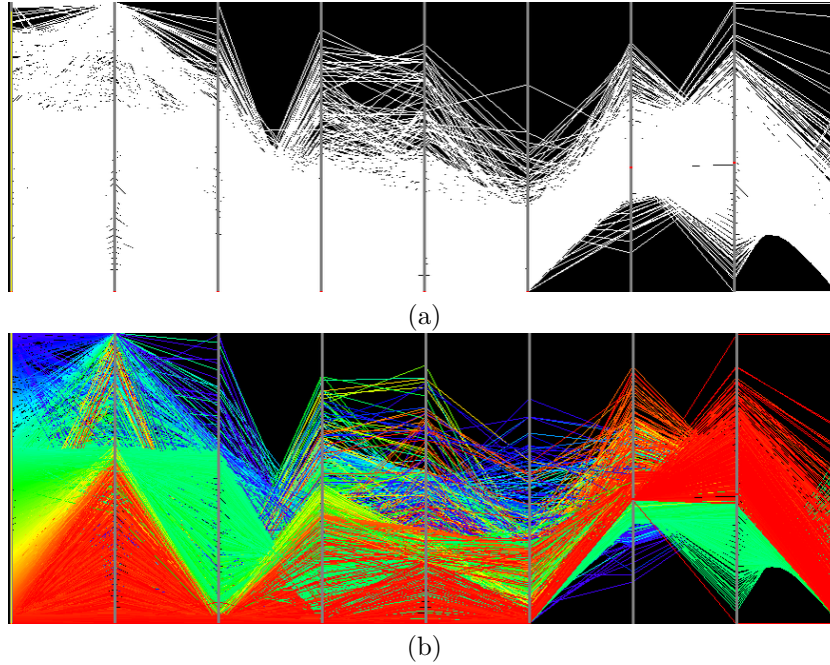


Figure 4.1: Parallel coordinates: (a) A parallel coordinate display is quickly obstructed if too many samples are visualized. (b) A colormap (on the left most intensity axis) can maintain the structure of the data.

Applying Colors

Colormaps are frequently used with parallel coordinates to highlight structures in dense datasets. In our *SignatureSpace*, they can be applied to every dimension to generate a visual structure for the voxel polylines in the parallel coordinate display. Furthermore, we can limit the application of the colormap to a specific subrange of a signature dimension, effectively skipping the non-selected ranges from the colored display; they remain in their previously acquired color, which is by default white (on a black background). This range limitation for the colormapping essentially implements a brushing operation.

The question what colormap should be used is not easy to answer. In many applications a so called rainbow colormap is used [FWR99], where the full hue range of the HSV color model is mapped to a selected color range. Unfortunately, hue is not a linearly perceived range, which can lead to interpretation problems of accordingly colored data [RTB96]. Better are isomorphic colormaps, such as saturation or luminance maps. However, in *SignatureSpace*, we only use the colormapping to reveal how neighboring data samples evolve over the other signatures and the clear drawbacks of a rainbow/hue colormap do not really have a significant impact here. Hence, we still use the rainbow/hue colormap, but technically, any other colormap could be used as well.

In order to take more than one signature dimension into account for

the coloring, we can apply a color to another, *secondary signature* dimension range. The colors of the colormap from the *primary signatures* are then blended with that color. Alternatively, other *secondary signatures* could modulate *Saturation* and *Value* in the HSV color model.

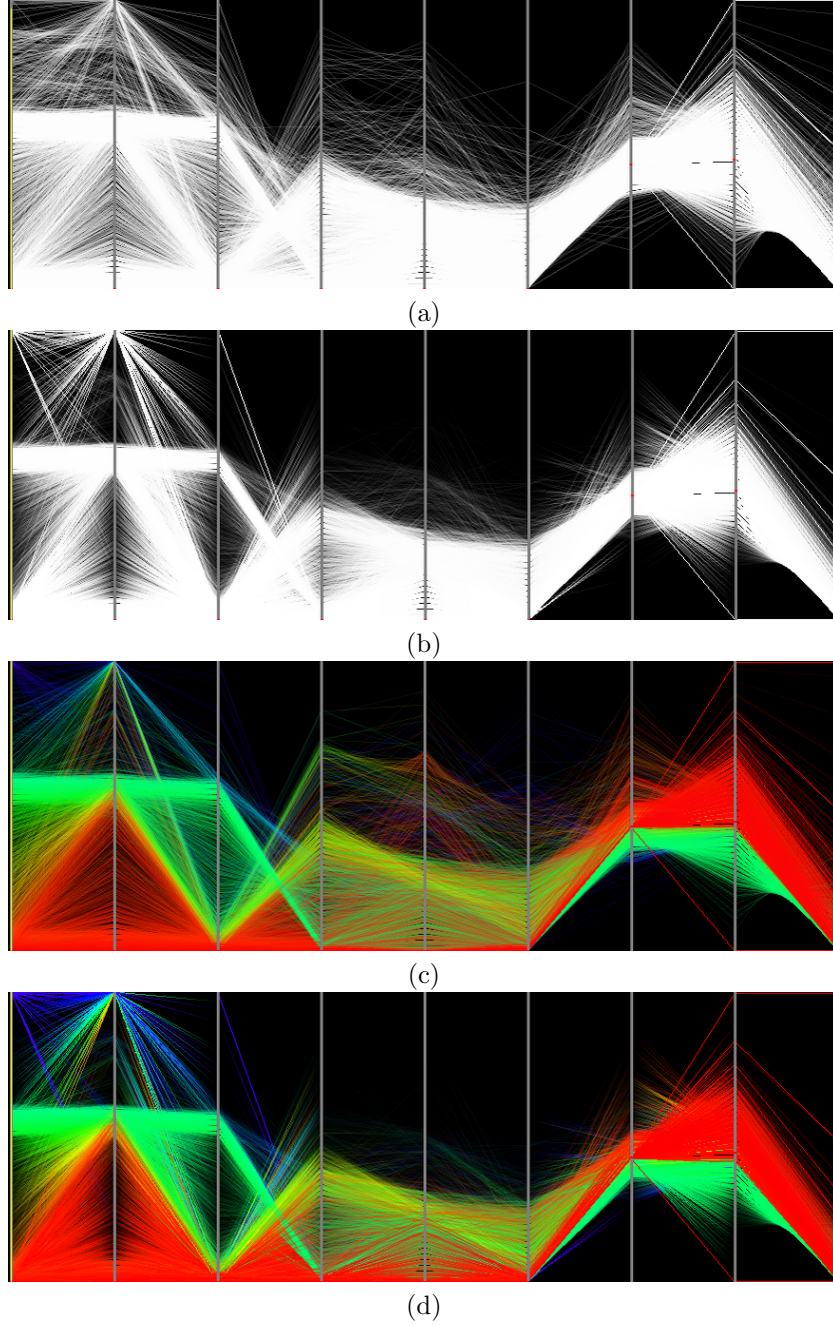


Figure 4.2: Unfolding clusters with opacity modulation: (a) shows an all-white coloring with Uniform Opacity Modulation (UOM), and (b) shows the same data with Weighted Opacity Modulation (WOM). UOM and WOM with an applied intensity colormap is shown in (c) and (d) respectively.

Modulating the Opacity

In order to modify our non-hierarchical version of parallel coordinates for a large number of data samples, we modulate the opacity of the voxel representation in two different ways. In the first option, *Uniform Opacity Modulation*, all data samples are assigned the same opacity value and are blended in our parallel coordinate display. Since the opacity value is controlled through a slider, the user can dynamically adjust the opacity value to the number of data samples in the volume dataset. This way, the large number of data samples (voxels of a volume dataset) – which can easily surpass millions of samples in even small datasets – does not obstruct the display (Fig.4.2ac).

The second option, *Weighted Opacity Modulation*, enables varying opacities of the voxels projection on all the different signature axes. If a large number of voxel polylines intersect an axis on the same value (actually on the same discrete value bucket), the opacity at that point should be higher than for a smaller number of voxel polylines. We achieve this goal by sorting all voxel values for a signature into n buckets and assign opacity values based on the number of voxels in each bucket. The opacity on the section of the polyline between two neighboring signature axes are linearly interpolated between the respective opacity values on the axes. This way, potential clusters on the signature axes are emphasized over more sparse regions (see Fig.4.2bd).

During the exploration of the *SignatureSpace*, we examine only one volume slice at the time, resulting in approximately 65536 interactively rendered and blended polylines at a time for our tested datasets. The more complex full volume sample exploration depends on the dataset size takes about 50 seconds. Therefore, all changes to of the overall opacity are performed interactively on slice level, and within seconds for the full volume.

Background Noise Removal

In contrast to the typical database query orientation of information visualization oriented data, typical volumetric data on regular grids have data samples for every grid position. Consequently, many of these data samples do not represent interesting information. Very often, they are only background noise or simply represent *empty* regions. Nevertheless, these irrelevant data samples can block the view on the actual data (Fig.4.3b). To compensate for this effect, we provided a removal functionality that removes data samples from the rendering process. After specifying a value range in a signature dimension, the respective voxels with a polyline intersecting that value range are removed (Fig.4.3c). Typically, this operation is performed in the voxel intensity signature, which represents the original voxel values (Fig.4.3d). However, it can also be applied to any other signature that exposes irrelevant voxels. The *lobster* dataset (Fig.4.4a) for example shows that the background

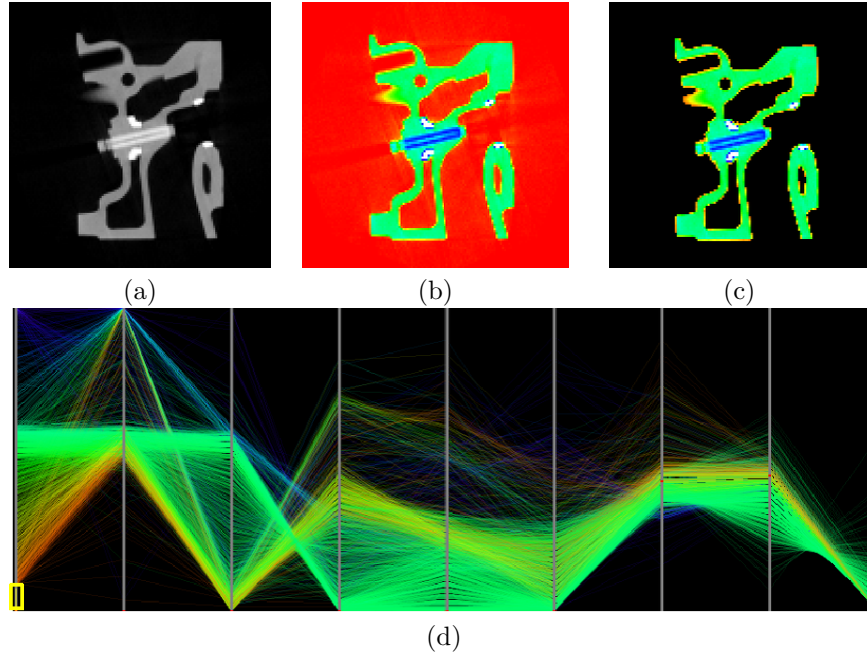


Figure 4.3: Background noise removal in a slice from the *engine* dataset: (a) Original (cropped) slice; (b) Background noise is also included in the voxel selection. (c) Background noise voxels are removed depending on how they belong to a certain value range in a specific signature (intensity in this example). The different colors mark the different materials according to the intensity colormap in (d). The remaining orange/yellow voxels are due to partial volume effects. (d) shows the *SignatureSpace* where the lower (red) intensity parts (yellow frame) are removed (compared to Fig. 4.2c)

noise removal operation can be easier achieved, without a potential loss of voxels that are subject to the partial volume effect or are right on the boundary of non-noise voxels. Figure 4.4 compares the background noise removal effects based on the intensity signature (Fig. 4.4ab) and noise removal based on the skew signature (Fig. 4.4cd). While the intensity based noise removal generates a much tighter result, it can lead to problems if the *lobster* needs to be segmented, since parts of the shell might be gone as well.

Scatter Plots

Parallel coordinates usually show a projection of the data distribution on a dimension axis. Although the density of polyline intersections (or their opacities) at certain values of that axis gives cues how the data is distributed, often one would like to see a histogram of the data, possibly together with another signature dimension. Similar to many other systems (e.g., [HLD02]), our *SignatureSpace* also provides this kind of scatter plots of the intensity (first signature dimension) and any other signature to allow a more detailed view on the data distribution.

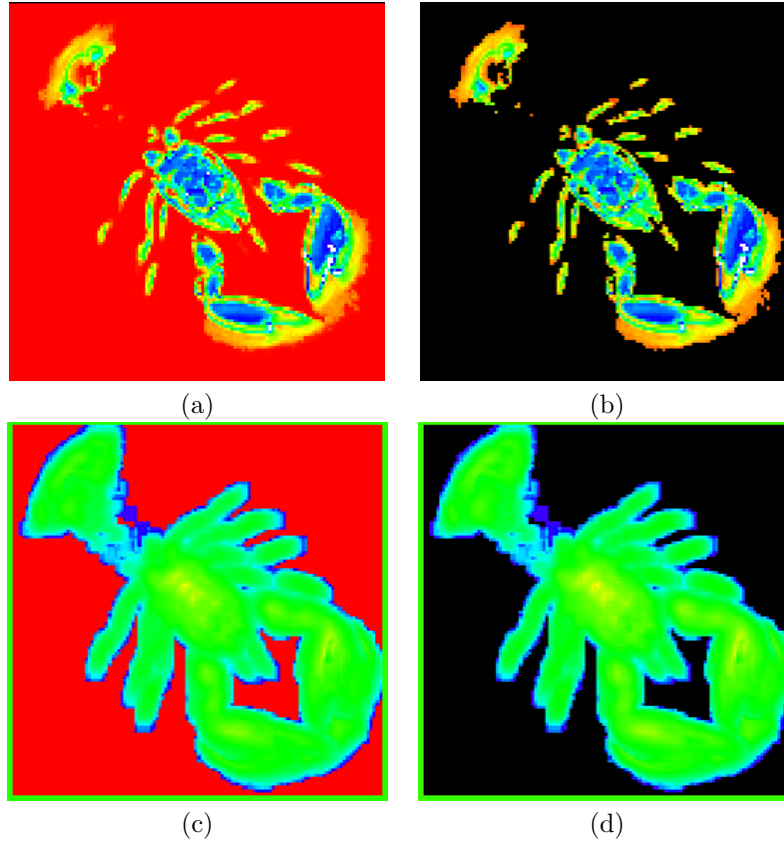


Figure 4.4: Background noise removal in a slice from the *lobster* dataset: (a) Dataset with colormapped intensity signature; (b) Background noise voxels are removed based on intensity. (c) Dataset with colormapped skew signature; (d) Background noise voxels are removed based on skew.

4.3 Analysis of Volume Data Using the SignatureSpace

Our modified parallel coordinates display consists of nine different signatures as described below. Almost all of the individual signatures have been used previously in the various approaches for the analysis of volumetric datasets or for the (semi-)automatic design of transfer functions. Only two signatures have been added by the authors of this paper, namely the up and down history, where we follow the local gradient one unit step forward and backward to examine the voxel intensity at the new position. This way, the signatures provide some notion of the materials on the other side of the boundary².

Currently, the *SignatureSpace* consists of the following signature dimensions:

- Intensity: the actual voxel value in the volume dataset (*data vol-*

²Lum and Ma also used the gradient direction for the specification of lighting transfer function [LM04].

ume)

- Up history of the voxel: we follow the gradient for one unit step and take the next voxel value as up history.
- Down history of the voxel: similar to the up history, but this time, we follow the negative gradient direction.
- Gradient magnitude: computed by the central difference at the voxels [BPS97, KD98].
- Second order moment³, which is computed by

$$m_2 = \frac{1}{N} \sum_{j=0}^{N-1} (x_j - M)^2 \quad (4.1)$$

with mean

$$M = \frac{1}{N} \sum_{j=0}^{N-1} x_j \quad (4.2)$$

where N is the number of voxels and x_j is the respective voxel value [TLM01].

- Second derivative: the central difference at the voxel position within the gradient field [KD98, PWH01].
- Third order moment³, which is computed by

$$m_3 = \frac{1}{N} \sum_{j=0}^{N-1} (x_j - M)^3 \quad (4.3)$$

with M, N, x_j as in m_2 [TLM01].

- Skew: a derived statistical value computed from higher order moments m_2 , mean M, and voxel value x_j [NIS04].

$$skew = \frac{1}{(N-1)m_2^{3/2}} \sum_{j=0}^{N-1} (x_j - M)^3 \quad (4.4)$$

- Kurtosis: also a derived statistical value from the higher order moments m_2 , mean M, and voxel value x_j [NIS04].

$$kurtosis = \frac{1}{(N-1)m_2^2} \sum_{j=0}^{N-1} (x_j - M)^4 - 3 \quad (4.5)$$

³Please note, that we do not use the local higher order moments described in [TLM01], but the “regular” higher order moments which consider the whole voxel context as neighborhood. Therefore, N represents all voxels in the current context, which can either be the current data slice, or the whole volume.

These signatures span the *SignatureSpace*, which we use for the analysis of volume data. However, additional signatures can be integrated, if one should decide that they convey useful information. Note that the different signatures are in different data ranges, thus there is only little range correlation between some of the signatures. Currently, we map the whole data range onto the respective coordinate axis. However, a sub range could be selected, which is either computed or selected manually.

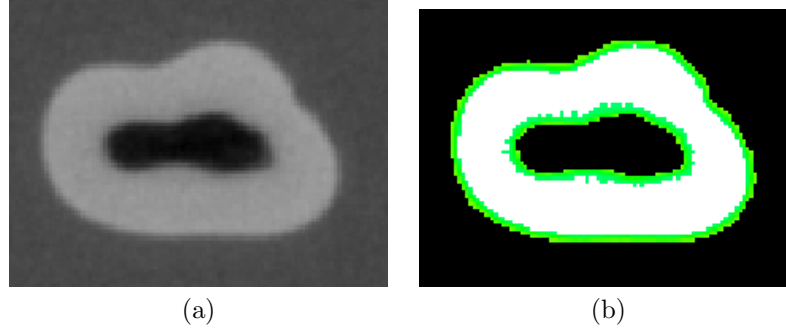


Figure 4.5: Slice from the *tooth* dataset, an industrial CT scan of a human tooth, which was part of the transfer-function bake-off. (a) shows an original cut slice from the dataset, and (b) shows voxels that are subject to the *partial volume effect* are rendered in green, while the actual feature voxel are rendered in white.

In the next paragraphs, we discuss the advantages of the *SignatureSpace* with several datasets, which are described in Table 4.1. These datasets represent various features with attenuating boundary features and typical datasets taken from daily life practice.

The traditional specification of transfer functions was only based on the intensity histogram of the voxels. Depending on the source and nature of a volume dataset, different materials could be identified by peaks and valleys in the histogram, if the intensity range of these materials were separable. The famous *engine* dataset – which is a CT scan of an engine block (see Table 4.1 and Fig. 4.3) – is an example for these cases. Also CT scans of a patient body provide easy differentiations between tissue, bone, and air. Even more interesting is the fact that a colormap applied to the intensity signature will also expose voxels that are subject to the notorious partial volume effect. With this effect, the sampling rate at volume reconstruction cannot correctly represent the material transition from a high intensity to a significantly lower one, and results in “voxel halo” of lower intensity around the high intensity materials (Fig. 4.5).

In case of an overlap, these materials cannot be differentiated by intensities alone. Furthermore, if a set of voxels of one material are adjacent to two different materials, eg., one with higher voxel values and one with lower voxel values, the traditional transfer functions based on a standard histogram cannot identify all material interfaces between

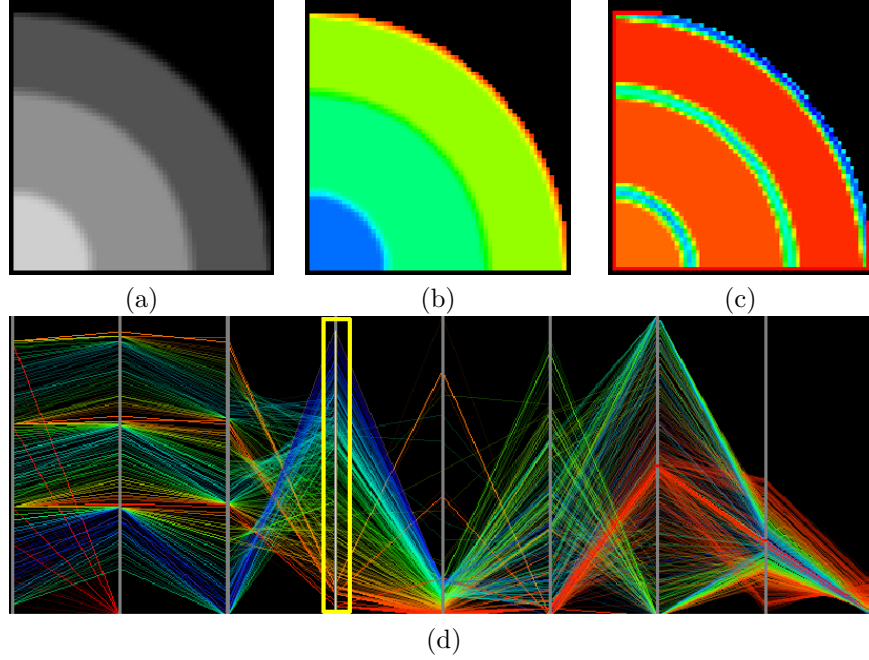


Figure 4.6: Slice from the *cones* dataset: (a) shows the original slice, (b) shows a colormapped slice based on the intensity signature, (c) shows a colormapped slice based on the gradient signature, and (d) shows the SignatureSpace with a colormap based on the (yellow framed axis) gradient signature.

those materials. This differentiation requires the gradient magnitude – since a local maximum in the gradient magnitude exposes one of the respective material interfaces – or a zero crossing in the second derivative [KD98]. An example of this situation can be seen in the *tooth* dataset (Fig. 4.5), and in the *cones* dataset (see Table 4.1 and Fig. 4.6). In the latter dataset, we see three different quarter cones of different intensity are put into each other, creating three different material boundaries (background/low, low/middle, middle/high; high/background is not present here), similar to the *cylinders* dataset in [KD98]. The *cones* dataset was generated by a volume modeling system.

Some datasets, however, do not expose a specific isosurface that could be specified. This frequently happens if a dataset represents a probability density or other discretized distribution functions. In these situations, a distinct gradient cannot be found and higher order functions need to be used. Figures 4.7 and 4.8 show examples of such an analysis, where the gradient colormap does not expose any specifically useful information. The more suitable third order moment colormap can be used to generate a transfer function.

Another situation is demonstrated with the *Plexiglas phantom* dataset (see Table 4.1 and Fig. 4.9). This dataset scanned by a rotational C-arm scanner and it is aimed at measuring reconstruction quality based on different scanning setups. It consists of three connected cavities

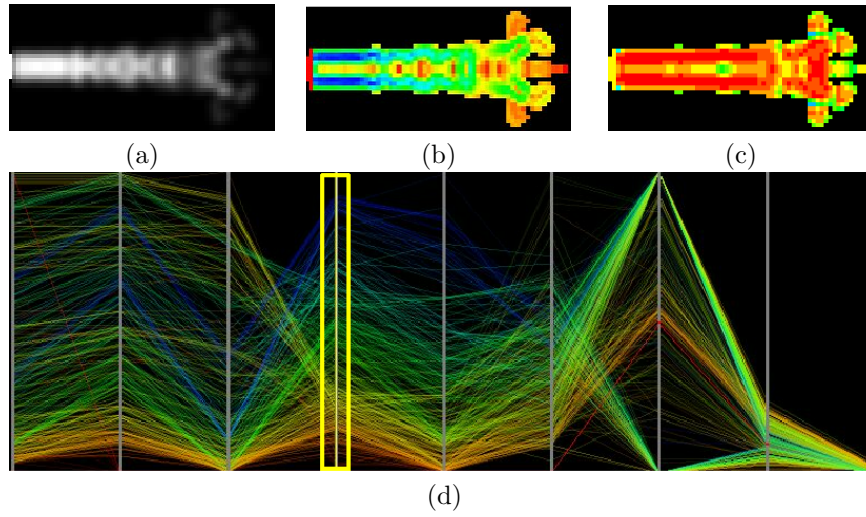


Figure 4.7: Colormapped slice from the *fuel* dataset, which was generated by a simulation of a fuel injection into a combustion chamber. (a) shows the original (cropped) slice, (b) shows a colormapped slice based on the gradient signature, which exposes the attenuation artifacts, and (c) shows a colormapped slice based on the skew. The SignatureSpace with a gradient colormap (yellow framed signature axis) is shown in (d).

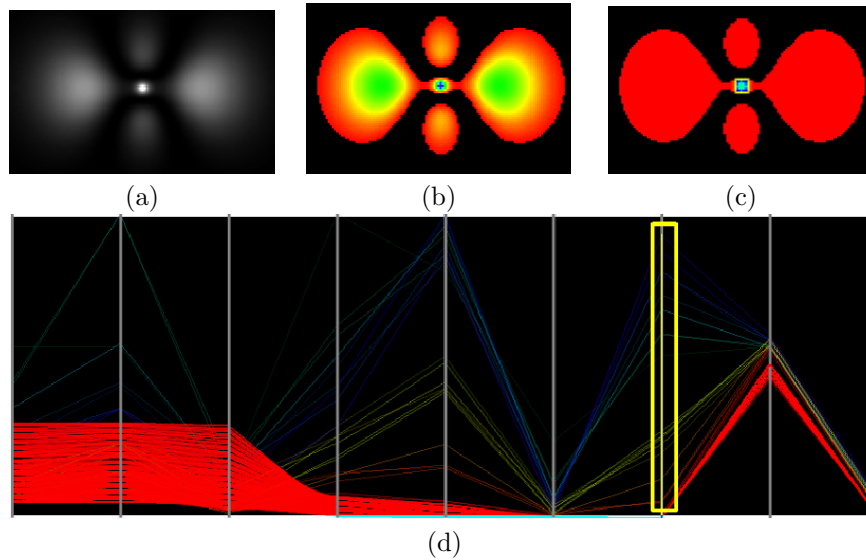


Figure 4.8: Colormapped slice from the *hydrogen* dataset, which was generated by a simulation of spatial probability of the electron in a hydrogen atom, residing in a strong magnetic field. (a) shows the original (cropped and brightened) slice, (b) shows a colormapped slice based on the intensity signature, and (c) shows a colormapped slice based on the 3rd order moment. The SignatureSpace with a 3rd order moment colormap (yellow framed signature axis) is shown in (d).

with a Plexiglas boundary, filled with a (exactly measured quantity of) contrast agent. Due to reconstruction artifacts of the scanning devices,

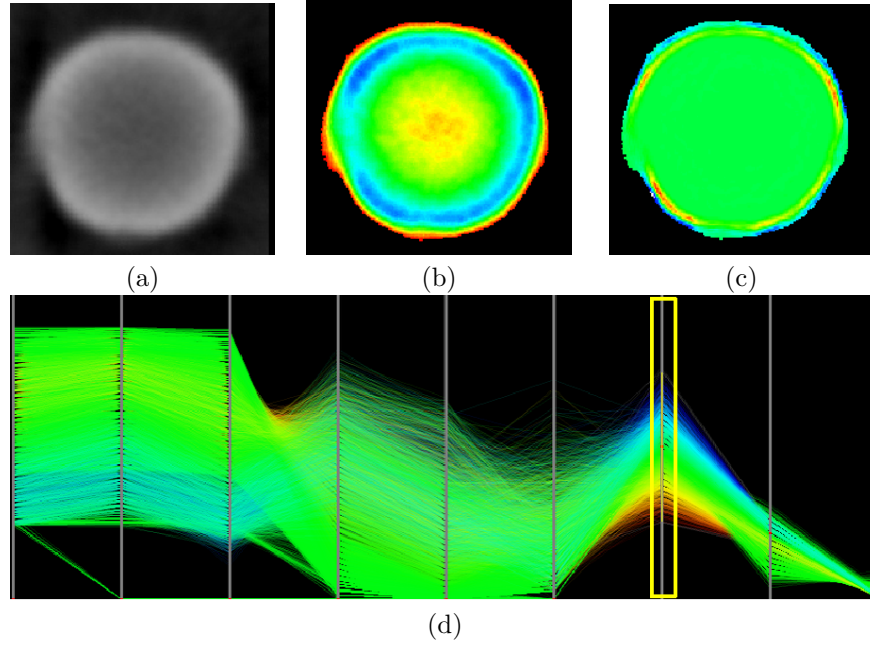


Figure 4.9: Colormapped slice from *Plexiglas phantom* dataset. (a) shows the original (cropped) slice, (b) shows a colormapped slice based on the intensity signature, which exposes the attenuation artifacts, and (c) shows a colormapped slice based on the higher 3rd moment that depicts the interior successfully. The SignatureSpace with a 3rd order moment colormap (yellow framed signature axis) is shown in (d).

the reconstructed volume dataset contains an attenuating region in the center of the cavities, in particular of the main cavity. Regular intensity or gradient-based analysis will interpret the attenuated region as a second boundary, which of course is only based on the data artifact and is not a real feature (Fig. 4.9b). Note that we used two different signatures to classify the dataset; we used the intensity signature for background noise removal and the third order moment for the colored classification.

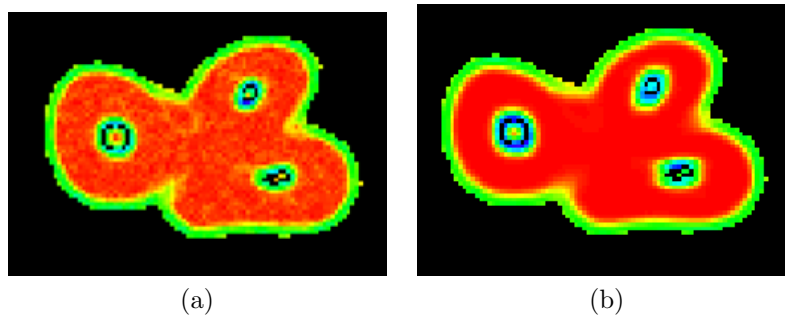


Figure 4.10: Slice from the *tooth* dataset, an industrial CT scan of a human tooth, which was part of the transfer-function bake-off. (a) shows slice with colormap of gradient signature, while (b) shows slice with colormap of 2nd order moment signature.

A quite similar effect is shown in Figure 4.10, where the colormapped second order moment signature generates significantly more homogeneous results than the colormap based on the gradient signature. Therefore, the second order moment signature enables an easier classification or segmentation task of the different materials. Here again, we used two different signatures for removal (intensity) and colored classification (second order moment).

With higher analysis methods such as second order moments and in particular the third order moment, shown in Figure 4.9c, this area is correctly interpreted as homogeneous region, although the artifacts in the volume dataset are generated by the attenuation.

4.3.1 Strategies for Using the *SignatureSpace*

Table 4.1: Dataset overview: The table shows the respective dataset resolutions, the total number of samples, the actually not-removed number of samples, and the used pre-processing time to compute the signatures on a Centrino Laptop running at 1700MHz, with 1GB of main memory. Some datasets have been cropped (^c) and some datasets have been downsampled (^d) to save memory space and (pre-)processing time.

Datasets Resolution	# Samples	# Relevant Samples	Pre-Processing Time (s)
Plexiglas Phantom ^c 162x172x214	5.963M	2,717K	565s
Tooth ^c 137x113x155	2.400M	426K	222
MRI Head ^d 127x127x 80	1.290M	502K	150
Lobster ^d 128x128x128	2.097M	311K	297
Engine ^d 128x128x128	2.097M	172K	258
Hydrogen Atom 128x128x128	2.097M	60K	257
Cones 64x64x64	262K	133K	26
Fuel 64x64x64	262K	8K	33

The *SignatureSpace* provides techniques for an exploratory data analysis approach for volumetric data. While some of the individual signatures do assume a data model, others do not. Consequently, we can use the *SignatureSpace* without having specific knowledge about the data to be analyzed.

Nevertheless, there are different characteristics of the *SignatureSpace* that can be used to examine specific signature dimensions for the various datasets.

- Clustering: As we know from information visualization, clusters expose similarities or correlation between data samples.
- Color distribution: The color distribution shows how the data samples move through the different signature dimensions, and thus expose possible correlations. The color distribution is also partially compensating the problems arising due to the different possible sequences of dimensions in the parallel coordinates.
- Sample distribution: If data samples are represented well in one signature dimension, they are not mapped into a very small data range of the respective signature axis.

In short, we are looking for signatures, where the data covers a significant part of the data range, and possibly has a reasonable sample cluster. We completely ignore signatures where the data is projected into a very small range, like in the kurtosis (last signature axis) in Figures 4.7 and 4.8.

The examples we have shown so far, use a variety of different signatures to extract a meaningful representation. Datasets that have a good material differentiation, with no intensity (voxel value) overlap, can be easily analyzed using intensity and gradient signatures, whereas the gradient signature is only necessary, if one material has boundaries to more than one other material (eg., datasets *cones*, *tooth*, and *engine* in Figures 4.6, 4.5, and 4.3).

However, if the boundary is less distinct, or might even suggest a misunderstanding of the data features (eg., *Plexiglas phantom*), intensity and gradient-based analysis will not lead to a successful solution. In this case, the third higher order moment was able to grasp the true data characteristics (Fig. 4.9).

Another example are the *fuel* and *hydrogen* datasets, generated by physical simulations (Figs. 4.7 and 4.8). Both datasets represent densities, where a specific boundary – except for a general outside boundary – cannot be found. Our *SignatureSpace* provides a framework that allows for an analysis that is not focusing on boundaries alone. In particular the third order moment signature for the *hydrogen* datasets allows for the identification of the different features in the dataset, as shown in Figure 4.8c, where lower density regions are marked in red, while the hot spot in the center is identified as a different feature. These marked regions can later be used as segmentation mask in typical segmentation environments.

In our last example, we demonstrate how the *SignatureSpace* can be used to classify different materials in *MRI head*, a downsampled T1-weighted MRI dataset. For the classification, we use again two signatures; the intensity signature to identify the background noise voxels (red in Figure 4.11b) and the second order moment signature for the identification of the skin (blue) and of the boundaries of the main brain

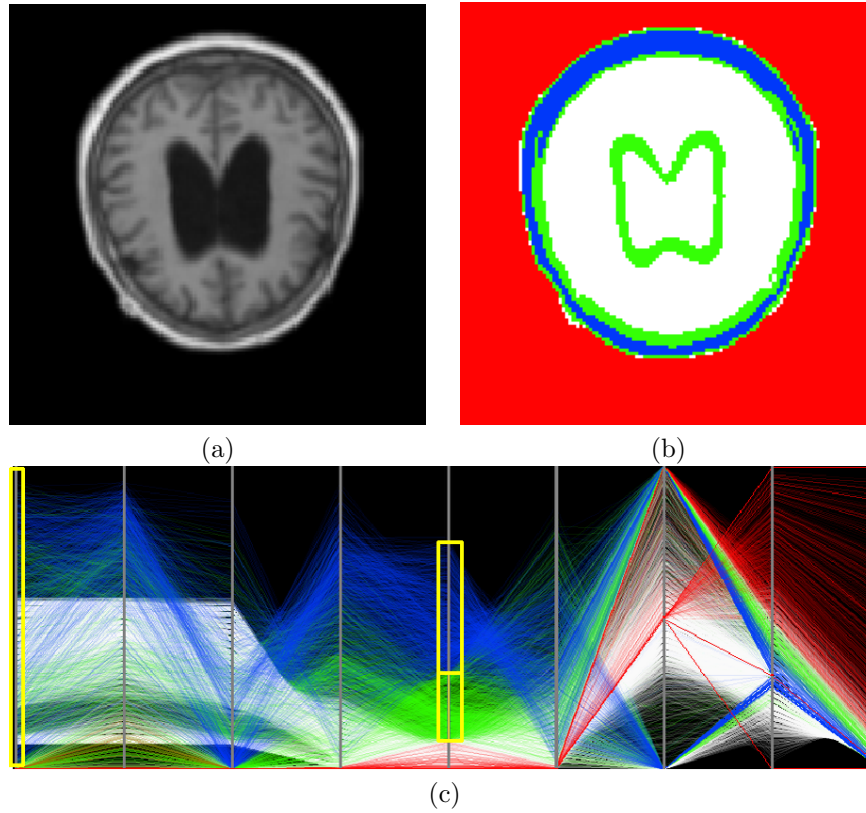


Figure 4.11: Colored slice from the *MRI head* dataset. (a) shows an original (cropped) slice, (b) shows a classification based on the intensity and 2nd order moment signatures. The *SignatureSpace* with a color band setting based on the intensity (red background; 1st framed signature axis) and 2nd order moment signatures (blue and green; 2nd yellow framed signature axis) is shown in (c).

fluid filled cavities (green). The respective *SignatureSpace* is shown in Figure 4.11c. Note the different colormap bands in the first (intensity) and fifth (second order moment) signatures from the left. Alternatively, the background noise voxels could have been identified by the down history signature (at the very bottom of the axis), or by the skew signature (also at the very bottom of the axis). The correlation between these signature dimensions becomes obvious in our modified parallel coordinate approach (Fig. 4.11c).

The process of removing background samples and finding an appropriate signature for the classification (or a combination of signatures) takes usually less than a minute. If one (or more) signatures are brushed to select a specific set of data samples, the overall time – including time for noise removal, brushing, exploration, and using different signature color mappings – increased in the presented examples to less than five minutes⁴.

⁴The longest classification time required the MRI head (Fig. 4.11). It involved an intensity colormap for background voxel removal, and a color band brushing of the 2nd order movement signature.

Please note that the objective of this paper is not to propose higher order moments or their derivatives as driving signatures for the data analysis, but to provide a holistic approach to data analysis, which keeps all available signatures in the focus of the analysis.

4.3.2 Memory Usage

The major problem of the *SignatureSpace* is its large memory consumption. Every signature requires a full volume sized data array to store the signature values for each voxel. To differentiate the various signatures from the original voxel volume, we call the additional volumes *signature volumes* and the original voxel volume *data volume*. Besides the large number of data and signature volumes, the memory consumption is also driven by the float values that are required by many of the *signature volumes*.

There are several possible solutions for the memory problem. Depending on the required accuracy and data range, the float data of some of the signatures could be downsampled into a single-byte integer representation. Another possibility of saving memory comes from the observation that the analysis is often based only on a subset of signatures. Hence, we can save the memory of the signatures that are not required. Finally, suitable compression schemes can be used to reduce the required memory, but at the same time, this would reduce the display and interaction time, since the respective signature volumes need to be decoded first.

Currently, we do not use any of the suggested solutions for the memory problem, but a downsampling of the dataset dimensions for large datasets (see Table 4.1). While there is a loss of accuracy involved, we believe that this can be neglected for the purpose of data analysis.

4.4 Results

In this section, we briefly present some rendering results, which have been generated based on some of the signature volumes. In Figure 4.12ab, we show two snapshots from the isosurface rendering of the *Engine* dataset. Please note how the gradient isosurface is bigger and more blurred (Fig. 4.12b), than the isosurface from the intensity signature (Fig. 4.12a). A similar observation can be done for the MRI head, which is rendered using the 2nd order moment (Fig. 4.12cd). These blurring effects are due to the inherent low-pass filtering of the gradient and 2nd order moment volumes. However, in a semi-transparent volume rendered image, this blurring would be less apparent.

Table 4.1 gives an overview of the datasets used in our experiments. It also contains the time needed by a Centrino laptop, equipped with 1GB of memory and one CPU running at 1.7GHz. Please note that

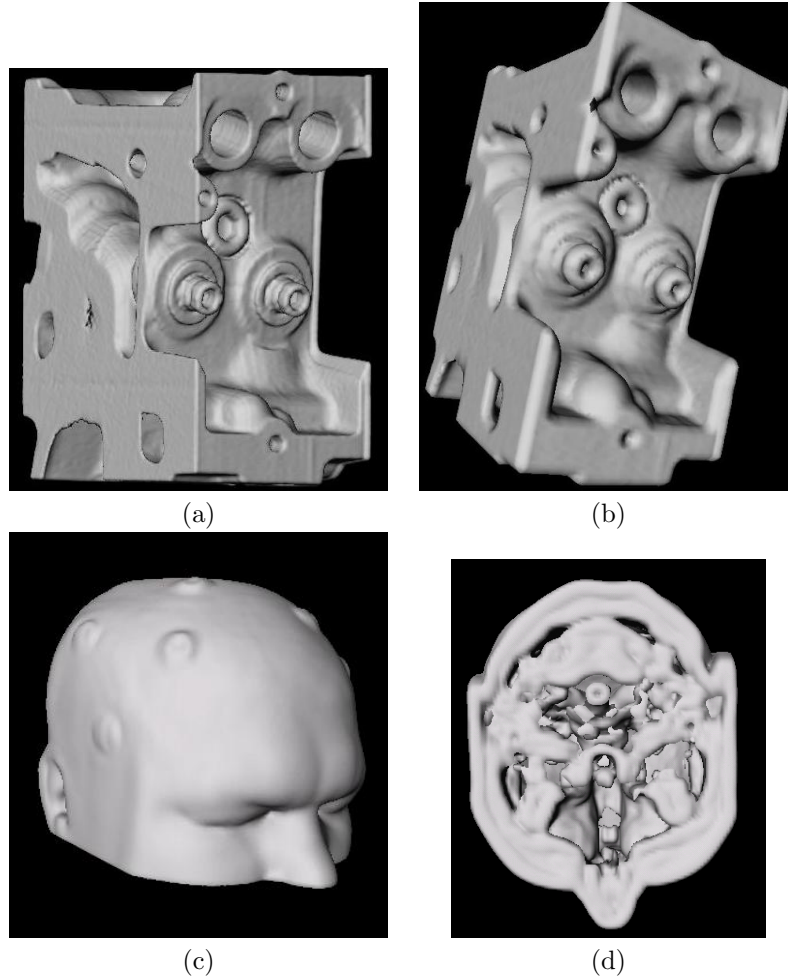


Figure 4.12: Isosurface rendering of *Engine* dataset. (a) shows the isosurface from the intensity signature, (b) shows the isosurface from the gradient signature. (c) and (d) show two snapshots from the skin isosurface of the 2nd order moment signature volume. Note that the bumps in the surface are fiducial markers attached to the head for an operation.

these timings are required only once to compute the *SignatureSpace*. Afterwards, it is stored to disk and can be used at any time.

SignatureSpace offers options to render the polylines of the parallel coordinate display based on the current volume slice, or based on the whole volume. Slice-based rendering is achieved at interactive framerates ($> 5\text{fps}$), while full volume rendering of the *SignatureSpace* took up to 30s for the largest volume (Plexiglas phantom).

4.5 Conclusion and Future Work

In this paper, we introduced the *SignatureSpace*, a novel approach for the analysis of volumetric datasets by adopting techniques from infor-

mation and scientific visualization. In this approach, several different metrics derived from the original volume datasets are presented in a parallel coordinate system, which has been improved to be able to handle volume datasets, since they contain orders of magnitude more data samples than typical information visualization datasets. In contrast to many other approaches for the analysis of volume datasets, we do not favor a specific data or boundary model; all signatures are presented equally to the user, who can quickly determine the most useful signature dimension for her/his analysis goal. The powerful combination of signatures enables a data-driven multi-metric classification strategy.

Since the *SignatureSpace* provides analysis only in the data or sample space – no spatial relationships are considered, it hence is a classification approach that provides support for the design of multi-dimensional transfer functions. However, the presented classification strategies can be used as a pre-process for any segmentation task.

In the future, we would like to explore further the exploration of multi-modal datasets with the *SignatureSpace*, where each modality (or data attribute values) is presented next to each other. This way the similarities throughout the different modalities or attributes can be observed in one display.

As we have already pointed out, the colormaps applied to the different signature dimensions can also be used to specify (multi-dimensional) transfer functions. We would like to explore that option further, in particular the possibility of automatically specifying suitable transfer functions.

Finally, the memory problem needs to be addressed in order to analyze truly large volumetric datasets. Right now, we only use dimension downsampling to handle medium and large datasets.

Acknowledgments

The *fuel* and *hydrogen* datasets are courtesy of the Priority Program SFB 382 of the German Research Foundation (DFG) and were downloaded from <http://www.volvis.org>. Also the original versions of the *lobster* and *engine* datasets can be found there. The *Plexiglas phantom* dataset is courtesy of Özlem Gürvit of the University Hospital Frankfurt, the MRI head of the University Hospital Tübingen, and the tooth dataset is courtesy of General Electric Corporate Research.

This research has been supported, in part, by NSF CAREER grant ACI-0093157, by the Department of Neurosurgery of the University Hospital Tübingen, and by DFG project VIRTUE of the focus program 1124. We would like to thank Urs Kanus for proof reading.

Bibliography

- [BPS97] C. Bajaj, V. Pascucci, and D. Schikore. The Contour Spectrum. In *Proc. of IEEE Visualization*, pages 167–174, 1997.
- [DGH03] H. Doleisch, M. Gasser, and H. Hauser. Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data. In *Proc. of Eurographics/IEEE VGTC Symposium on Visualization*, pages 239–248, 2003.
- [FAT99] I. Fujishiro, T. Azuma, and Y. Takeshima. Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis. In *Proc. of IEEE Visualization*, pages 467–470, 1999.
- [FWR99] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical Parallel Coordinates for Visualizing Large Multivariate Data Sets. In *Proc. of IEEE Visualization*, pages 43–50, 1999.
- [GKW02] G. Grinstein, D. Keim, and M. Ward. Information Visualization, Visual Data Mining, and Its Application to Drug Design. In *IEEE Visualization, tutorial 1*, 2002.
- [GRW⁺00] D. Gresh, B. Rogowitz, R. Winslow, D. Scollan, and C. Yung. WEAVE: A System for Visually Linking 3-D and Statistical Visualization, Applied to Cardiac Simulation and Measurement Data. In *Proc. of IEEE Visualization*, pages 489–492, 2000.
- [HB86] H. Höhne and R. Bernstein. Shading 3D-Images from CT using Gray-level Gradients. *IEEE Transactions on Medical Imaging*, MI-5(1):45–47, 1986.
- [HHKP96] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of Transfer Functions with Stochastic Search Techniques. In *Proc. of IEEE Visualization*, pages 227–234, 1996.
- [HLD02] H. Hauser, F. Ledermann, and H. Doleisch. Angular Brushing of Extended Parallel Coordinates. In *Proc. of IEEE Symposium on Information Visualization*, 2002.

- [ID90] A. Inselberg and B. Dimsdale. Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry. In *Proc. of IEEE Visualization*, pages 361–378, 1990.
- [Ins85] A. Inselberg. The Plane with Parallel Coordinates. *The Visual Computer*, 1:69–92, 1985.
- [KD98] G. Kindlmann and J. Durkin. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In *Proc. of IEEE/ACMSymposium on Volume Visualization*, pages 79–86, 1998.
- [LM04] E. Lum and K. Ma. Lighting Transfer Functions for Direct Volume Rendering. In *Proc. of IEEE Visualization*, pages –, 2004.
- [MW91] J. Miller and E. Wegman. Construction of Line Densities for Parallel Coordinate Plots. In *Computing and Graphics in Statistics*, pages 107–124. Springer-Verlag, Heidelberg/New York, 1991.
- [MW95] A. Martin and M. Ward. High Dimensional Brushing for Interactive Exploration of Multivariate Data. In *Proc. of IEEE Visualization*, pages 271–278, 1995.
- [NIS04] NIST. NIST/SEMATECH e-Handbook of Statistical Methods. <http://www.itl.nist.gov/div898/handbook/>, accessed April 2004, 2004.
- [PLB⁺01] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, R. Machiraju, and J. Lee. The Transfer Function Bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, 2001.
- [PWH01] V. Pekar, R. Wiemker, and D. Hempel. Fast Detection of Meaningful Isosurfaces for Volume Data Visualization. In *Proc. of IEEE Visualization*, pages 223–230, 2001.
- [RTB96] B. Rogowitz, L. Treinish, and S. Bryson. How Not To Lie With Visualization. *Computers in Physics*, 10(3):268–274, 1996.
- [TLM01] S. Tenginkai, J. Lee, and R. Machiraju. Salient Iso-Surface Detection with Model-Independent Statistical Signatures. In *Proc. of IEEE Visualization*, pages 231–238, 2001.
- [TLM03] F. Tzeng, E. Lum, and K. Ma. A Novel Interface for Higher-Dimensional Classification of Volume Data. In *Proc. of IEEE Visualization*, pages 505–512, 2003.

-
- [TM02] S. Tenginakai and R. Machiraju. Statistical Computation of Salient Iso-Values. In *Proc. of Eurographics/IEEE VGTC Symposium on Visualization*, pages 19–24, 2002.
 - [TPM05] M. Tory, S. Potts, and T. Möller. A Parallel Coordinates Style Interface for Exploratory Volume Visualization. In *IEEE Transactions on Visualization and Computer Graphics*, pages 71–80, 2005.
 - [War94] M. Ward. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proc. of IEEE Visualization*, pages 326–333, 1994.

Part IV

**Segmentation, Volumetry,
and Visualization**

Chapter 5

Segmenting The Mastoid: Allocating Space in The Head for a Hearing Aid Implantation

**Zein Salah¹, Dirk Bartz¹, Erwin Schwaderer,² Florian Dammann²,
Marcus Maassen³, and Wolfgang Straßer⁴**

¹Visual Computing for Medicine Group, University of Tübingen

²Department of Radiology, University Hospital Tübingen

³Department of ENT Surgery, University Hospital Tübingen

⁴WSI/GRIS, University of Tübingen

**Appeared in the Proceedings of the Workshop Bildverarbeitung
für die Medizin 2004.**

We present a procedure for the segmentation of the mastoid - a sponge-like bone structure behind the middle ear - from CT Images. This segmentation is a first pre-operative step for a robot-supported hearing aid implantation. Up to now, the mastoid is milled out by an ENT surgeon to allocate space for the hearing aid. This tedious milling process of the mastoid is intended to be performed by a robot to reduce the required time and improve the accuracy.

5.1 Introduction and Related Work

Segmentation of 3D medical images plays a vital role in many medical procedures, where it serves in most cases as a preparing step for further preoperative analysis and planning. In our research, we introduced a procedure for segmenting the mastoid. For this purpose, we investigated several segmentation methods provided by ITK (NLM In-

sight Registration and Segmentation Toolkit) and commonly used in segmentation.

Level-set-based segmentation filters worked well for the segmentation of various organs. For example, a level-set based approach was used in [vBSVN02] to segment the carotid artery in MR angiograms in order to determine the degree of stenosis. However, the inhomogeneous sponge-like nature of the mastoid blocked a successful application. Furthermore, level-set methods [MHE97, PD00] are significantly slower than the finally adopted region-growing method and require parameter tuning which contradicts our project goals. Other mostly automatic segmentation methods did also not succeed segmenting the mastoid, while semi-automatic filters like LifeWire [MB95] usually require too much per-slice user-interaction until the result is achieved. Region Growing (RG) techniques are among the most common segmentation approaches. Seeded Region Growing (SRG) presented in [AB94] and its improved version presented in [MJ97] both exploited the conventional RG algorithm, where the criteria of similarity of pixels is applied. They start with a small number of user-chosen seeds and group them into regions. A 3D region growing technique called Voxel Grow has been also introduced in [LF03].

The complex anatomical structure of the mastoid renders an automatic segmentation procedure as impractical. Henceforth, we are aiming at a segmentation procedure that requires as little as possible user interaction in order to release expensive physician time. Our approach is based on numerous data enhancement and segmentation filters of ITK. We also developed a viewer for the exploration of the volume datasets. In particular, it allows the viewing, visual evaluation, and comparison of different segmentation results. Furthermore, we included a post-processing editor for further processing of the segmentation result, mainly to simplify the removal of incorrectly segmented regions.

5.2 Methods

Our segmentation pipeline is based on a multi-slice CT scan of the head, focusing on the ear region. To minimize user interaction as much as possible, we used 3D implementations of the processing filters and avoided the use of slice-based filters where per-slice interaction is required. Our segmentation procedure, depicted in Figure 5.1, includes three major step:

- 1. Preprocessing:** In order to avoid long, possibly unnecessary computing, we begin by extracting the region of interest out of the input image and apply all further processing on it, rather than on the whole image. This preprocessing step prepares the input image into an appropriate form for segmentation step. In particular, it vacates the mastoid from the sponge-like structures that degrade the success

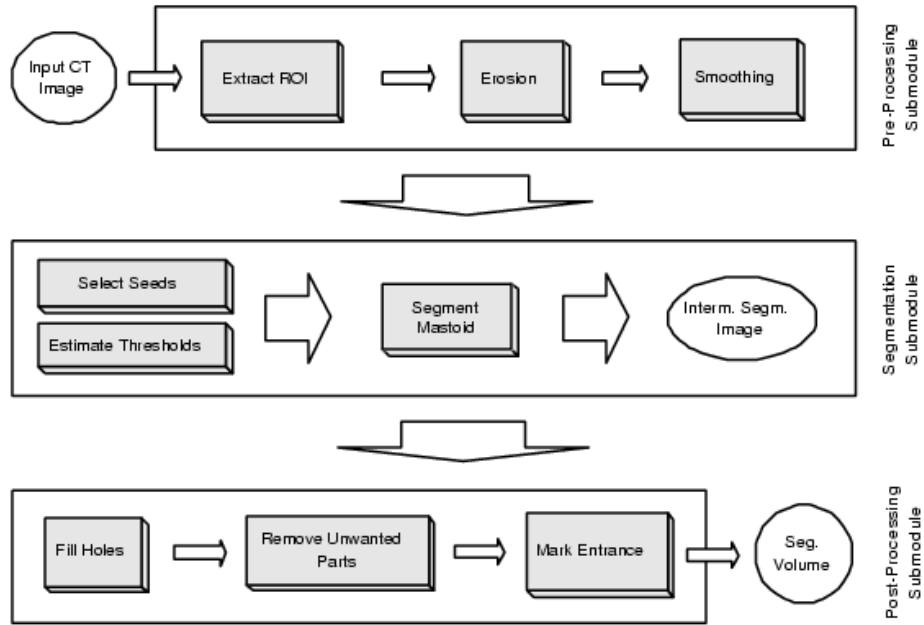


Figure 5.1: System block diagram

of the segmentation filters severely. These thin bone structures are removed using an improved implementation of morphological erosion [SS01]. Remaining structures are resolved by filtering the image with recursive Gaussian smoothing [Der93].

2. Segmentation: This step aims at extracting the mastoid hull. We implemented this step using intensity-threshold based region growing. First, we apply a region growing filter to segment the grey soft tissue region that surrounds most of the mastoid. A second filter is used to segment the bones. We combine the results of these two filters together with the original input image to produce a temporary image, from which a final region growing filter extracts the mastoid. This sequence is outlined in Figure 5.2. Each of the three filters requires two threshold values as input. These values can be entered interactively, but the system can also automatically guess appropriate values by examining the histogram of the smoothed image computed in the previous step.

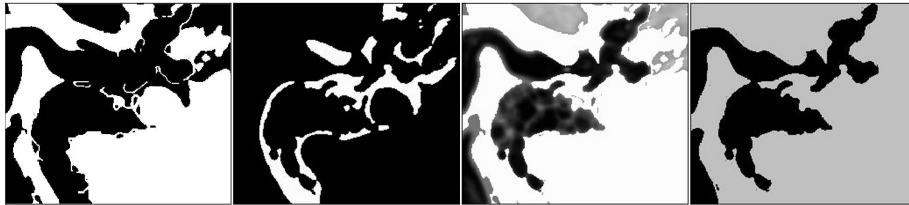


Figure 5.2: Segmentation process (left to right): Grey soft tissue, bones, temporary image, and preliminary mastoid segmentation (black)

Unfortunately, the mastoid segmentation filter may still leak out

through holes, hence false regions may be segmented. Since we do not have pre-knowledge about the size as well as the locations of these holes, we decided to permit this leaking and post-process it later.

3. Post-Processing: This step aims at refining the result of the previous step so that we end up with a clean segmentation of the mastoid. Three main operations are performed in this step:

1. the (possibly existing) internal holes (not-segmented, closed 3D regions within the segmentation) are removed by inverse region growing.
2. unwanted parts are clipped from the segmentation. For this purpose, we developed a semi-automatic editor that detects *connections* to the leaked regions and then removes unwanted parts.
3. an opening of the skull is selected as an entrance for the robot-driven milling process. This is done by marking all skull voxels that (from an out-to-in view) lead to the mastoid.

The second step is actually the most important, and is basically performed as follows: two voxels are selected by the user, one inside the mastoid and the other inside the object to be removed (the leaked region). We then find a path between these voxels in the form of an approximated centerline. Note that the path should go through the connection between the mastoid seed and the object seed. The position along this path where it passes through the narrowest location is located by generating cross sections across the path at equally spaced distances and finding the section that cuts the segmentation with smallest intersection. Finally, the user can accept this position or select an alternative. By placing a clipping plane, the unwanted part is removed. The whole procedure is applied for all parts that we wish to remove.

5.3 Results

The segmentation results of our procedure were evaluated visually using a viewer that we developed specially for this purpose. This viewer enables the viewing of medical images in a traditional, slice-oriented manner. We incorporated this viewer in our application to view intermediate as well as final results. Moreover, it allows the viewing, visual evaluation, and comparison of different segmentation results by aligning slices of the volume data with semitransparent overlays of segmented data.

The upper row of Figure 5.3 shows three slices from a segmented mastoid volume. The lower row shows the realized entrance to the mastoid through the skull.

The execution time of the preprocessing and segmentation steps for full size images (approximately hundred 512x512 slices) take approximately 200 seconds on a 1200 MHz AMD PC. However, this process

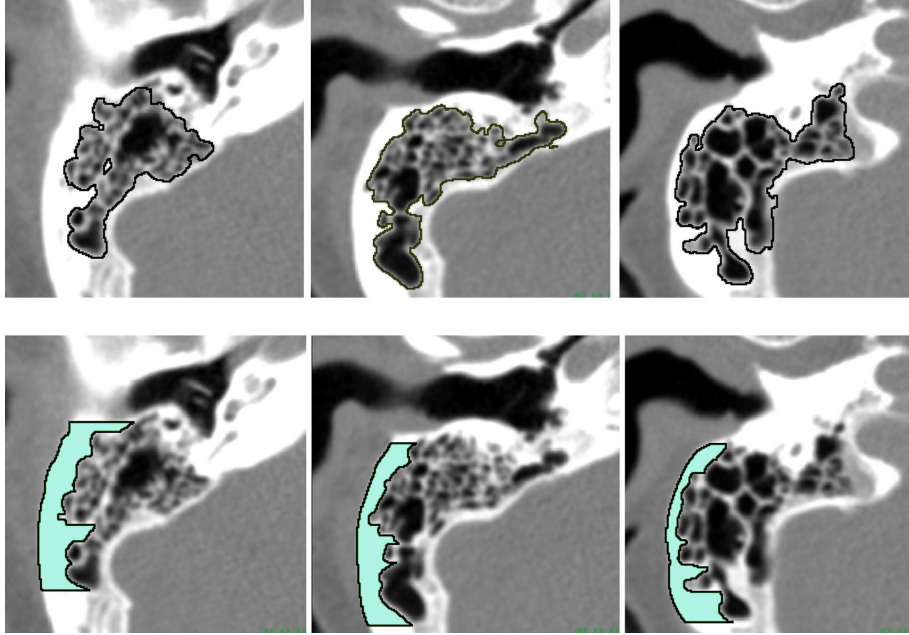


Figure 5.3: Segmented mastoid (above), realizing an entrance to the mastoid (below)

requires only 40-50 seconds if applied on an extracted region of interest (e.g. $250 \times 220 \times 90$). A complete post-processing session time varies, depending on the size of the volume and the complexity of the shape of the resulted segmented object, i.e. on the amount of required cutting. It ranges, however, between two and six minutes of required user-time, in contrast to the previously tedious manual segmentation that took between 20 and 40 minutes.

5.4 Discussion

The segmentation procedure that we presented is used for a robot-supported hearing aid implantation. As our top goal, we were able to reduce the previous segmentation time, in particular the required time for user-interaction. In the near future, the Department of Radiology will conduct a study on the effectiveness of the segmentation pipeline used by clinical personnel.

Acknowledgment

This work has been supported by the German Academic Exchange Service (DAAD) and the Competence Center for Minimally Invasive Medicine and Technology, Tübingen - Tuttlingen.

Bibliography

- [AB94] R. Adams and L. Bischof. Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.
- [Der93] R. Deriche. Recursively Implementing The Gaussian and its Derivatives. Technical Report Technical Report Nr. 1893, INRIA, 1993.
- [LF03] I. Larrabide and S. Fiorentini. Voxel Grow: A Region Growing Segmentation Technique. In *Proc. of International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications*, 2003.
- [MB95] E. Mortensen and W. Barret. Intelligent Scissors for Image Composition. In *Proc. of ACM SIGGRAPH*, pages 191–198, 1995.
- [MHE97] R. Malladi, J. Sethian (H. Hege, and K. Polthier (Eds)). *Visualization and Mathematics: Experiments, Simulations, and Environments*, chapter Level Set Methods for Curvature Flow, Image Enhancement, and Shape Recovery in Medical Images, pages 329–345. Springer Verlag, Heidelberg, 1997.
- [MJ97] A. Mehnert and P. Jackway. An Improved Seeded Region Growing Algorithm. *Pattern Recognition Letters*, 18:1065–1071, 1997.
- [PD00] N. Paragios and R. Deriche. Coupled Geodesic Active Regions for Image Segmentation: A Level Set Approach. In *Proc. of European Conference in Computer Vision*, pages 224–240, 2000.
- [SS01] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, 2001.
- [vBSVN02] C. van Bemmelen, L. Spreeuwers, M. Viergever, and W. Niessen. Level-Set Based Carotid Artery Segmentation

for Stenosis Grading. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 36–43, 2002.

Chapter 6

Live-Wire Revisited

Zein Salah, Jasmina Orman, and Dirk Bartz

Visual Computing for Medicine Group, University of Tübingen

Appeared in the Proceedings of the Workshop Bildverarbeitung für die Medizin 2005.

The live-wire approach is an interactive, contour-based segmentation technique. Generally, the contour of a targeted object (anatomical structure) is built by interactively selecting control points and finding minimal-cost paths between them. By its very nature, this method is applicable only to 2D images. For the segmentation of 3D datasets (volumes), the interactive generation of live-wire contours has to be applied to each slice of the volume. This process can be quite tedious, due to the sometimes intensive user interaction. In this contribution, we propose adaptive propagation as an alternative to individually processing all image slices or shape-based interpolation of live-wire contours.

6.1 Introduction and Related Work

In the context of medical image segmentation, and due to object characteristics as well as image quality, a fully-automatic segmentation is in most cases not possible. Moreover, the results of automatic segmentation methods often need further correction. For example, the complex anatomical structure of the mastoid (Figure 6.3) renders a fully-automatic segmentation as impractical. The segmentation results of the semi-automatic procedure presented in [SBS⁺04] include 10-25% of the slices that require further refinement. However, manual tracing is inaccurate and laboriously unacceptable. Hence, intelligent contour generation methods have been proposed. The *live-wire* [BM96, BM97], also known as *intelligent scissors* [MB95] is one of the most known methods.

Live-wire is applicable to 2D images. For the segmentation of 3D datasets (volumes), the interactive generation of live-wire contours has to be applied to each slice of the volume. This process can be quite tedious, due to the sometimes intensive user interaction. Several extension of the live-wire approach to 3D images (data volumes) have been addressed in the literature. Principally, the goal is to avoid repeating the interactive generation of contours for all slices. An alternative is to use *Shape-Based Interpolation* [HZB92], where live-wire contours are generated only for some *key* slices. Contours of the missing slices are then interpolated from contours of the key slices. In [SPP00, SPP01a, SPP01b], shape-based interpolation is combined with live-wire. Seed points used for computing live-wire contours at the key slices are interpolated to produce seed points for generating live-wire contours of the missing ones.

In this contribution, we propose a new alternative, which we call adaptive propagation. In our method, control points of the live-wire contours are propagated to the missing slices (not interpolated), without requiring significant user interaction. The core idea is to propagate control points through slices, where the positions of these points are automatically adjusted at each slice. Unlike shape-based interpolation, adaptive propagation does not require a closed contour and is henceforth better suited for rapidly changing objects.

6.2 Method

The live-wire approach is slice-based. In the next section, we present a brief overview of the general approach. Adaptive propagation is then described.

6.2.1 Live-Wire Overview

With live-wire, the extraction of an object boundary is based on the computation of minimal cost paths between vertices of a graph. The two-dimensional image is transferred into a weighted graph, where the vertices of the graph correspond to image pixels and the edges connect neighboring pixels. Edge weights are defined in terms of local cost functions. Usually, paths are computed using Dijkstra's graph search algorithms and, as a local cost function, the weighted sum of different component functionals that describe edge features [MB95].

Contour extraction is done by interactively selecting a seed point on the boundary of the targeted object. A second target point is then selected and a minimal cost path from the seed to the target point is computed as described before. Ideally, the path will wrap around the object boundary. If the generated path deviates from the virtual boundary, a new target point is selected nearer to the seed point. Having generated the first path segment, the target point becomes the new

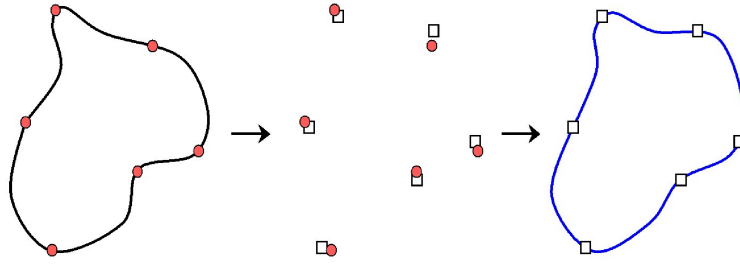


Figure 6.1: Propagation of control points.

seed point and a new path segment is generated similarly. The process is repeated until the whole boundary is outlined. Therewith, the final object contour is composed of multiple minimal-cost path segments.

6.2.2 Adaptive Propagation

The basic idea of adaptive propagation is to reuse information from a current segmented slice to segment the next one. In the first step, we specify the control points of the initial slice using the traditional live-wire approach [MB95]. The resulted segmentation is defined in terms of a piece-wise contour. Each segment of the contour is computed as a minimal-cost path between two control points. Control points between contour segments can be added interactively to refine the contour. Most of the user time required by the algorithms accounts for this choice of points.

For adaptive propagation, we are propagating the control points of a slice S_i to the next slice S_{i+1} and use them to further compute the new live-wire paths. The idea is illustrated in Figure 6.1, where the contour on the left is composed of live-wire segments between interactively-selected control points (red circular bullets). These control points are propagated to the next slice. This is motivated by the shape coherence of anatomical structures, i.e., a structure (an organ) does not tend to change in shape rapidly between slices. Hence, the contours of an organ in two successive slices also deviate not much from each other. In this context, propagated points will not necessarily lie exactly on structure contour, but are assumed to be very close to it. In the next step the positions of propagated points are corrected. Adjusted positions are shown by the square bullets in the middle of Figure 6.1. The idea is simply to move each control point towards the actual contour of the targeted object. Specifically, we search a limited neighborhood of the propagated point for the pixel with the greatest likelihood to lie on the contour, according to the live-wire cost function. As live-wire contours usually resemble edges between an object and its surrounding, we use this heuristic and move the propagated control point to the pixel with the largest gradient magnitude. The concept is illustrated in Figure 6.2.

After adjusting the positions of all control points, a live-wire path is computed between each two successive points as shown in Figure 6.1 (right). The process is then repeated to the next slice, until all selected slices are processed.

6.3 Results

Since the adjustment of control points is heuristic in its very nature, and the shape of an organ might exhibit an unexpected rapid variation between slices, user verification (and may be correction) is still needed. Possibly, new control points must be inserted to refine the contour. However, these situation are relatively rare compared to shape-based interpolation. In our experiments, we tested adaptive propagation with five volume datasets and compared it with shape-based interpolation. On average, 40% of 77 slices from mastoid/ear CT datasets could be processed automatically (with minor corrections) by adaptive propagation, while only less than 20% could be processed automatically by interpolation (also, with minor corrections).

An example is shown in Figure 6.3, where the live-wire contour of slice 34 was generated interactively. The contours in the subsequent slices (35, 36, and 37) were generated using adaptive propagation. Slice 37 was further refined by inserting new control points (images 37a and 37b).

6.4 Discussion

In this contribution, we present an extension of the live-wire methods for 3D dataset segmentation, in which the tedious process of generating live-wire contours at every slice of the volume is significantly reduced. In our method, adaptive propagation, the interactive, full creation of contours is performed to only a fraction of the volume slices. Live-wire contours in the missing slices are generated (not interpolated),

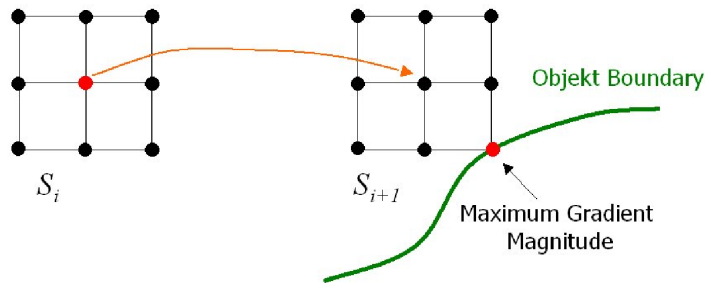


Figure 6.2: Adjusting the position of a propagated control point.

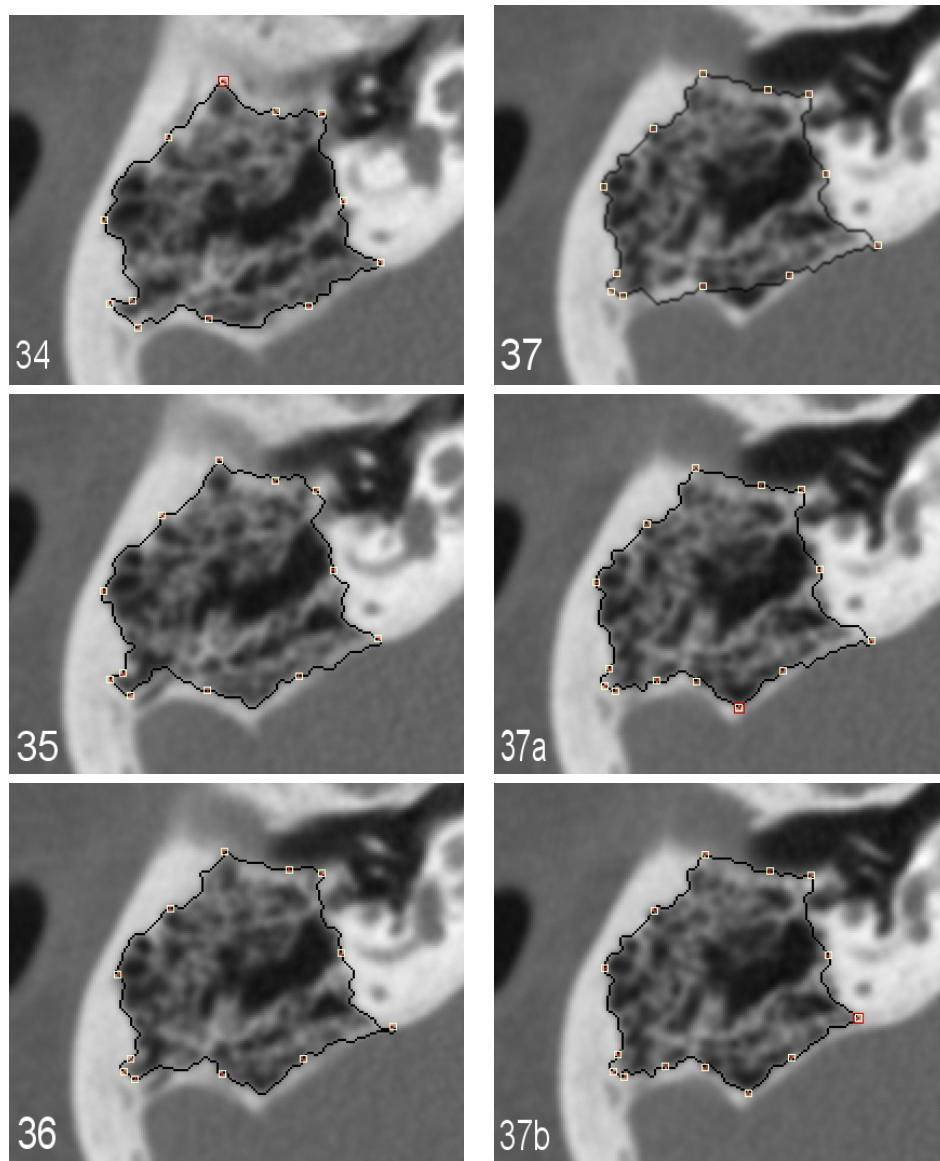


Figure 6.3: Propagation example: the contour in slice 34 was generated interactively. The contours in the slices (35, 36, and 37) were generated using propagation, the last two images show refinements of slice 37 performed by inserting new control points.

but without requiring significant user interaction. The idea is to propagate control points through slices, and at each slices, the positions of these points are automatically adjusted. Compared to the interpolation method, adaptive propagation does not rely on closed contours. Furthermore, the generation of new control points depends on the previous slice and not on the first and last of the interpolated slices. Therefore, it adapts better to local variations and is henceforth better suited for rapidly changing objects. Finally, (semi-)automatically generated

live-wire contours are usually smoother between slices than manually specified live-wire contours.

Acknowledgment

This work has been supported by the German Academic Exchange Service (DAAD) and the DFG focus program 1124 on "Medical Navigation and Robotics".

Bibliography

- [BM96] W. Barret and E. Mortensen. Fast, Accurate and Reproducible Live-Wire Boundary Extraction. In *Proc. of Visualization in Biomedical Computing*, pages 183–192, 1996.
- [BM97] W. Barret and E. Mortensen. Interactive Live-Wire Boundary Extraction. *Medical Image Analysis*, 1(4):331–341, 1997.
- [HZB92] G. Herman, J. Zheng, and C. Bucholtz. Shape-based Interpolation. *IEEE Computer Graphics and Applications*, 12(3):69–79, 1992.
- [MB95] E. Mortensen and W. Barret. Intelligent Scissors for Image Composition. In *Proc. of ACM SIGGRAPH*, pages 191–198, 1995.
- [SBS⁺04] Z. Salah, D. Bartz, E. Schwaderer, F. Dammann, M. Maassen, and W. Straßer. Segmenting The Mastoid: Allocating Space in The Head for a Hearing Aid Implantation. In *Proc. of Workshop Bildverarbeitung in der Medizin, Informatik Aktuell*, pages 45–49, 2004.
- [SPP00] A. Schenk, G. Prause, and H. Peitgen. Efficient Semiautomatic Segmentation of 3D Objects in Medical Images. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 186–195, 2000.
- [SPP01a] A. Schenk, G. Prause, and H. Peitgen. Local Cost Computation for Efficient Segmentation of 3D Objects with Live Wire. In *Proc. of SPIE Medical Imaging*, volume 4322, pages 1357–1364, 2001.
- [SPP01b] A. Schenk, G. Prause, and H. Peitgen. Optimierte Semi-Automatische Segmentierung von 3D-Objekten mit Live Wire und Shape-Based Interpolation. In *Proc. of Workshop Bildverarbeitung in der Medizin, Informatik Aktuell*, pages 202–206, 2001.

Chapter 7

Accurate Volumetric Measurements of Anatomical Cavities

Dirk Bartz¹, Jasmina Orman¹, and Özlem Gürvit²

¹Visual Computing for Medicine Group, University of Tübingen

²Institute for Neuroradiology, University Hospital Frankfurt

Appeared in *Methods of Information in Medicine*, 43(4):331-335, 2004.

Objectives: The volumetric assessment of anatomical cavities is of high relevance for various applications in medicine. Based on 3D scanning (i.e., CT) of these cavities, the volume can be determined by counting the volume elements of a segmentation of that cavity. Unfortunately, elements on the boundary of the segmentation require special treatment to obtain accurate volumetric measurements. In this paper, we propose a novel technique that in particular increases the accuracy of the volume estimation for the boundary elements of segmented anatomical objects.

Methods: Based on a 3D segmentation of an anatomical cavity, we recursively subdivide boundary volume elements into a set of simple situation, where the volume can be estimated easily.

Results: We performed volumetric measurements on seven datasets of phantom models made of plexiglass (see Fig. 7.1) scanned by a biplane angiography unit and assessed the quality of our method by comparing the measured volume by our novel method and by the fluid required to fill the phantom cavities.

Conclusions: Our method calculates a significant more ac-

curate volume of the segmented cavities than previous methods. Nevertheless, it is only slightly more computational expensive.

Keywords: Radiographic Phantom, Angiography, Measurements, Quality Assessment

7.1 Introduction

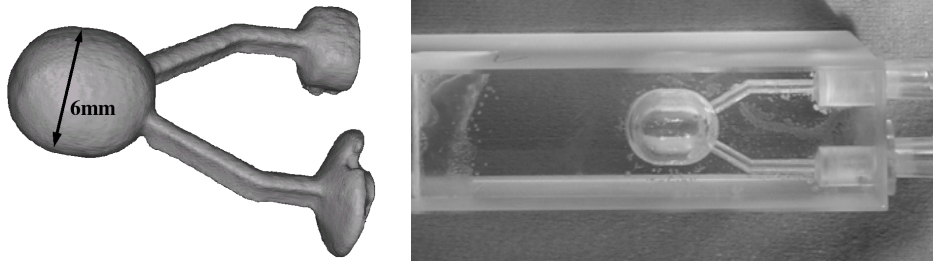


Figure 7.1: 6mm phantom dataset. (a) 3D rendering and (b) photography.

The volumetric assessment of body cavities is of high relevance for many clinical applications in medicine. Of particular interest is the assessment of the size stenosis in blood vessels, of the neck and the volume of aneurysms, and of the volume of the cardiac and cerebral ventricles and other CSF-filled cavities of the central nervous system. In particular the volume of an aneurysm might become meaningful in the near future for the estimation of the quantity of required material for embolization. Other useful applications include the assessment of the orbit volume for posttraumatic orbital reconstruction [GSH⁺02]. Unfortunately, the established methods are limited to an approximated estimation of these measures, based on 2D images - i.e., individual image slices or multi-planar reconstructions (with an arbitrary orientation within the dataset) from medical 3D scanners, such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), rotational angiography, or standard 2D x-ray images. In this paper, we perform volumetric measurements on phantom models of anatomical cavities. Since we can precisely measure the volume of these cavities by measuring the quantity of contrast agent (see Section 4) required to completely fill the phantoms, we can also assess the quality of our computational method based on the 3D volume dataset of the phantoms¹.

¹This is the main reason for using phantom models instead of real data, i.e., a stenosis of a blood vessel in MR angiography data.

7.2 Related Work

Recent research methods do use volumetric information of a complete 3D image stack. Relevant areas are usually differentiated from not relevant areas by a variety of different segmentation methods, like 3D region growing [BSSW99] or Watershed-transformation [HLD⁺01]. Based on the generated segmentation, its volume is calculated by summing up the participating voxels [LSW⁺98]. Hahn et al. calculated the volume based on histogram analysis to take over-segmentation due to partial volume effects and the watershed-transformation into account [HLD⁺01, HMD⁺02]. Unfortunately, no further examination of the actual accuracy of the volume computation is published. Furthermore, the boundary areas of the segmentation are treated with varying accuracy.

7.3 Objectives

Here, we present a three-dimensional volumetry method that is based on standard segmentation of 3D scanned data. It enables the voxel-accurate estimation of important measures such as the volume of a cavity. In particular, we are using the segmentation tools developed in the course of the VIVENDI project for virtual endoscopy [BSSW99]. Furthermore, we assess the quality of our results with measurements of the actual measured fluid volume of several phantom models of aneurysms, blood vessels, and cavities of the heart.

7.4 Methods

For the quality assessment, we used datasets that were previously acquired with a bi-plane angiography unit with a rotation option (Siemens Neurostar) that enables the acquisition of high-resolution 3D datasets with relatively short time of 7-14 seconds. For the phantom models, we used the high dose option for the dose rate and the 22cm image intensifier to acquire 128 to 400 individual images with varying resolutions (128x128 to 512x512). However, only datasets with the highest resolution of 512x512 were used. All phantom datasets have an isotropic pixel and slice distance starting at 0.2mm as lower bound. The phantom models were filled with a non-ionic, iodinated high contrast resolution (300) contrast agent, which was used to reconstruct the lumen of the assessed cavity. The concentration of the contrast agent was 100%. However, later research showed that optimal reconstruction quality rather requires a lower concentration of 60% to reduce beam hardening artifacts [GMS⁺02].

Rotational angiography does not perform a full rotation around the measured area, but only approximately 200 degrees, combined with a

cone beam reconstruction of the volume from the projection image. Consequently, the phantom models need to be positioned in the isocenter of the rotation unit with a longitudinal alignment, which is used also for the regular patient scanning [GMS⁺02].

Note that while the assessments in this paper will be based on these 3D rotational angiography datasets, volume datasets generated by other modalities such as MRI or CT can also be used. Boundary conditions imposed by the rotational angiography do not generally hold for CT or MRI. In contrast to rotational angiography, these modalities are significantly less sensitive to different positions and orientations. However, these modalities do usually acquire an anisotropic volume dataset. Therefore, the model orientation needs to be taken into account when scanning with these modalities.

7.4.1 Segmentation

For a proper volumetric measurement, we rely on a sufficiently accurate segmentation of the respective volumetric object. We use the segmentation tool SEGMedA of the VIVENDI system for virtual endoscopy to identify the relevant voxels [BSSW99]. It is based on a semi-automatic 3D region growing approach where the seed point and the lower threshold needs to be specified. Additionally, a second upper threshold can be specified to limit the segmentation by a voxel value interval (and henceforth by a second material interface). This standard segmentation is sufficient in most cases to identify the relevant voxels of an anatomical structure. This is in particular true for scanning protocols that ensure a high contrast between the cavities and the surrounding tissue. However, several other effects require an additional processing: Presumably due to shortcomings of the volume reconstruction algorithms, the injection of contrast media does not guarantee an even distribution. The resulting signal attenuation in those areas of the cavities renders the pure threshold-based identification impossible, since the respective voxels have a voxel value below the lower bound of the isovalue interval (Fig. 7.2b). Figure 7.2 shows two steps of the segmentation of one of the phantoms. After the simple segmentation of the outer hull with a threshold interval of [isovalue, MAX] (see Fig. 7.2a), we segment the inner core in the interval of [MIN, isovalue] (see Fig. 7.2b). Combined in a segmentation mask, these two areas represent the contrast media filled cavity of the phantom (we only focus on the main cavity of the phantom; the two leg like smaller cavities are omitted).

Unfortunately, it is not straightforward to establish the correct isovalue that limits the outer hull of a phantom. Signal attenuation can cause differences in the required threshold between the various datasets. To establish a standard method to determine a proper isovalue, the user draws a line from the outside of the cavity to the inside. On this line, the system generates a profile histogram how the voxel values evolve

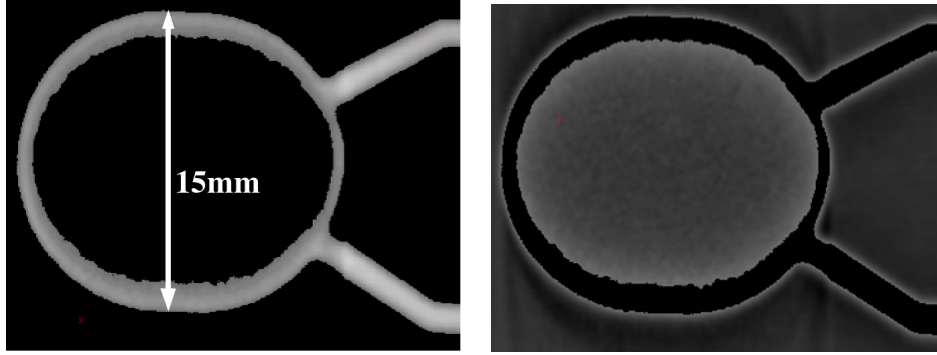


Figure 7.2: Image slice from the segmentation of the main cavity of the 15mm phantom: Bright voxels in (a) are above the lower threshold and show the outer hull (and legs of the phantom). The inner bright voxels in (b) show the inner core below the lower threshold, with significant signal attenuation.

from the outside to the inside. Additionally, we overlay this graph with the scaled gradient magnitude of that profile histogram. The largest change of voxel values indicated by these two curves also depicts the voxel value range, which indicates the material interface [KD98] between the low intensity plexiglass hull of the phantoms, and the high intensity outer hull of the contrast media filled phantom. Based on the discrete integration, the average intensity, and the corresponding variance of this profile area, we determine the isovalue used for the segmentation.

7.4.2 Computation of the Volume

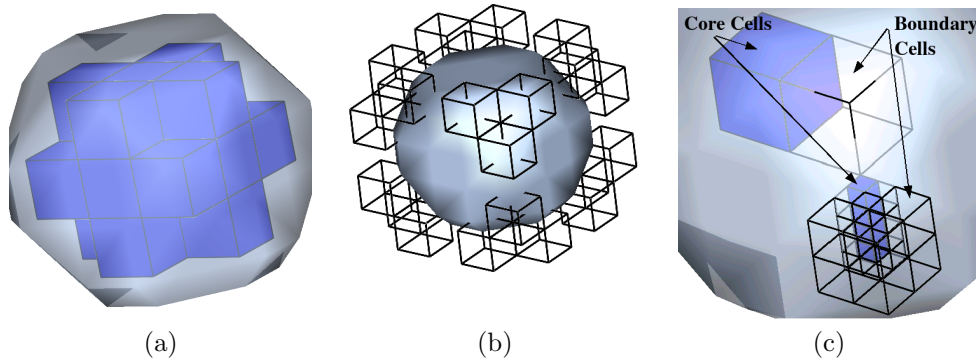


Figure 7.3: Volume computation of a voxel sphere: (a) Core cells (solid dark) fully contribute to the total volume. (b) Boundary cells (wire frame) with only one voxel being part of the segmentation contribute according to their intensity value (see also Fig. 7.4a). (c) Zoomed transparent representation of core cells (solid dark) and subdivided boundary cells (wire frame). Note that boundary cells at different subdivision levels are shown.

Based on the segmentation, all volume cells (blocks of $2 \times 2 \times 2$ voxels), which completely belong to the segmented area (core cells) are summed up to the volume, taking their pixel and slice distance into account (Fig. 7.3a). All cells which are located on the boundary of the segmentation (boundary cells) - only between one and seven voxels of those cells are located within the segmentation - must be treated differently (Fig. 7.3bc). While the number of those boundary cells is rather small in convex structures, they cannot be ignored in more complex structures, such as the cerebral or cardiac ventricular systems. Up to 50 belong to the boundary [LSW⁺98] (see also Fig. 7.5 and Table 7.1). In our examples in Table 7.1, 22% of the actual volume and 40% of all volume cells of the ventricular system are boundary cells. The volume of the boundary cells depends on a classification and the intensity values of the voxels. Similar to the classification into 256 cases of the Marching Cubes algorithms [LC87], we classify boundary cells according to the set of voxels that are located within the segmentation. Simple cases are separated from complicated cases. Fig. 7.4a-c shows three simple cases, where the voxels of a boundary cell that are located within the segmentation are either only on one corner, one edge, or one plane. In these cases, the respective volume of the boundary cells can be calculated according to the trilinear interpolation of their intensity values and summed up into the total volume (see also Fig. 7.3b). Note that the volume calculation for inverse cases (only one voxels are not part of the segmentation, two voxels on an edge are not part of the segmentation, etc.) is similar: the calculated volume of the outside voxels is subtracted from the volume of the full volume cell, and the difference is added to the total volume. Figure 7.4d shows a more complex case, where none of the above solutions can be applied. Instead, the boundary cell is subdivided recursively - where the new interior subvoxels are computed by trilinear interpolation - until the simple cases can be applied again, or until an error threshold is reached (Fig. 7.4c).

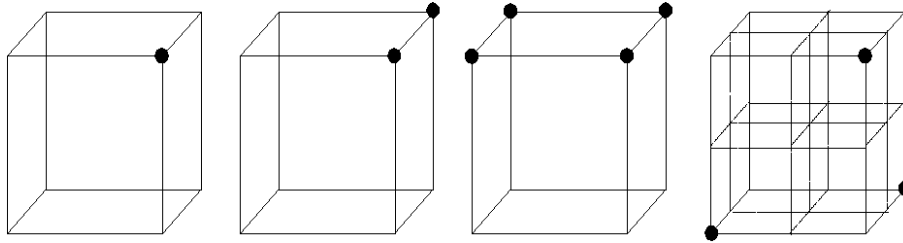


Figure 7.4: Four cases of volumetry classification: (a) Simple case 1, where only on voxel of a boundary cell are part of the segmentation. (b) Simple case 2, where two voxels on one edge are part of the segmentation. (c) Simple case 3, where four voxels on a plane are part of the segmentation. (d) Complex case that is subdivided into simpler cases. The dashed lines show the outlines of the sub-voxel cells.

Other computed measurements, such as distances, are also possible.

In particular the diameter of the neck of aneurysms can be estimated within VIVENDI from an endoscopic point of view.

7.5 Results and Discussion

For our volumetry assessment, we used a variety of different volume datasets, generated by the above described rotational angiography unit. These datasets represent phantom models, made of plexiglass, which model respective body cavities, such as blood vessels with aneurysms, or ventricles of the heart. They were manufactured according to specifications by clinicians in the radiology departments in Marburg and Tübingen. Figure 7.1 shows a 3D rendering and a photograph of such a phantom; it consists of a main cavity (with the transverse diameter depicted in Table 7.1 as dataset name) and two leg-like cavities branching from the main cavity. For the assessment of the volumetric measurements, we considered only the main cavities and clipped the legs from the segmentation.

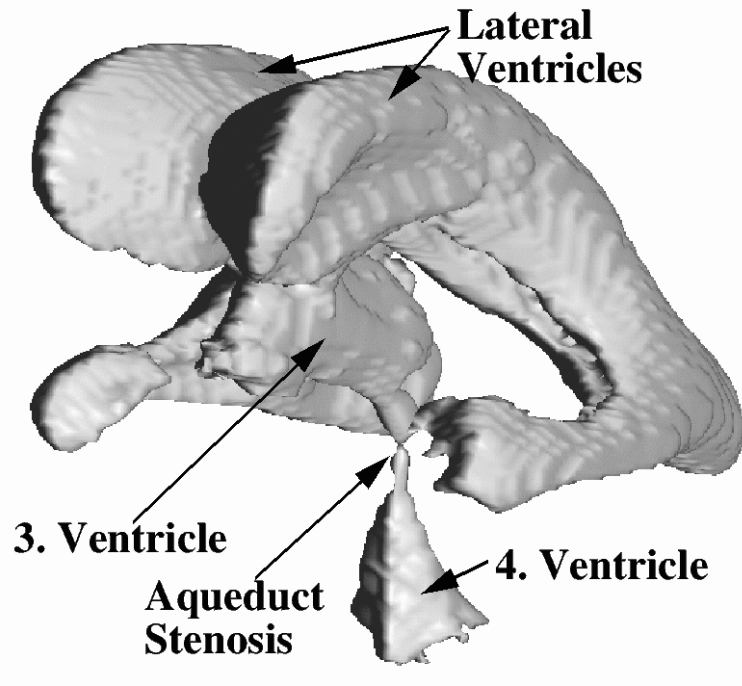


Figure 7.5: 3D model reconstructed from an MRI scan of a cerebral ventricular system with aqueduct stenosis.

In contrast to the daily clinical routine, the phantom model cavities are completely filled with contrast media. Therefore we can perform accurate and direct measurements of the cavity volume of the phantoms. Furthermore, this way of introducing the contrast media and the simple geometry of the models renders the image acquisition process less

Table 7.1: Volumetric measurements of various datasets with different rotation times. The percentage difference between measured and computed volume depend on the threshold. The name of the dataset depicts the transverse diameter of the main cavity of the phantom model (except for the last two datasets). As a comparison, we added the volume computation of a cerebral ventricular system, generated by an MRI scan of a patient (see Fig. 7.5).

Measured Fluid Volume Based on Model		Computed Volume Based on Dataset			
Dataset, Rotation Time	Volume [mm ³]	Threshold (%)	Volume [mm ³]	Difference (%)	Volume of Boundary Voxels [mm ³]
8mm, 8s	0.32	32	0.321	0.31	0.006
10mm, 5s	0.80	47	0.831	3.88	0.064
10mm, 8s	0.80	47	0.826	3.25	0.062
15mm, 8s	2.78	38	2.73	-1.18	0.11
20mm, 5s	5.30	59	5.73	-2.45	0.17
Heart, 8s	6.48	35	6.75	4.17	0.26
Ventricular System, n.a.	n.a.	29	91.00	n.a.	19.99

sensitive to partial volume effects, since no narrow features are present in the datasets. Consequently, we can restrain our measurements from using compensation techniques [HLD⁺01, HMD⁺02]. However, they remain relevant for other modalities or contrast injection protocols that match with the clinical routine.

Table 7.1 gives an overview of the measurements of the volume of the fluid need to fill the main cavity of the phantom models and by the volume computation based on the segmentation.

7.6 Conclusion

In this paper, we introduced a new method to compute the volume of provided segmentation. It treats its boundary with a recursive subdivision method to improve the accuracy of the volume computation of boundary cells. We assess the quality of the computed volume with a series of phantom models of cavities. While the real volume of the cavities of those models is known beforehand - or can be measured by filling them with a contrast agent, we compute the volume of the segmentation of those cavities to compare the computed real volume of the phantoms.

Overall, we showed that our method is able to achieve a high accuracy compared to the real volume of the models. However, some image acquisition methods, such as rotational angiography, need to be used with a fixed imaging protocol to reduce the possible cause of artifacts.

Acknowledgments

We would like to thank Joachim Hornegger of the University of Erlangen-Nürnberg for his support in this project. Furthermore, we thank Martin Skalej of the Department of Neuroradiology and Jürgen Hoffmann of the Department for Maxillo-facial Surgery of the University Hospital Tübingen for helpful discussions and Jan Fischer of the Visual Computing for Medicine group for Word support. This work has been supported by DFG projects CatTrain and Virtue, being part of SPP 1124, and Siemens Medical Solutions.

Bibliography

- [BSSW99] D. Bartz, W. Straßer, M. Skalej, and D. Welte. Interactive Exploration of Extra- and Intracranial Blood Vessels. In *Proc. of IEEE Visualization*, pages 389–392, 547, 1999.
- [GMS⁺02] Ö. Gürvit, M. Müller, R. Siekmann, M. El-Sheik, K. Klose, M. Skalej, and H. Alfke. Limitationen der Rotationsangiographie. In *RöFö 2002: Abstracts of the 83rd Deutscher Röntgen Kongress*, page 179, 2002.
- [GSH⁺02] N. Gellrich, A. Schramm, B. Hammer, S. Rojas, D. Cufi, W. Lagrééze, and R. Schmelzeisen. Computer-assisted Secondary Reconstruction of Unilateral Posttraumatic Orbital Deformity. *Plastic Reconstructive Surgery*, 110(6):1417–1429, 2002.
- [HLD⁺01] H. Hahn, M. Lentschig, M. Deimling, B. Terway, and H. Peitgen. MRI Based Volumetry of Intra- and Extracerebral Liquor Spaces. In *Proc. of Computer Assisted Radiology and Surgery*, pages 384–389, 2001.
- [HMD⁺02] H. Hahn, W. Millar, D. Durkin, O. Klinghammer, and H. Pleitgen. Cerebral Ventricular Volumetry in Pediatric Neuroimaging. In *Proc. of Workshop Bildverarbeitung in der Medizin*, Informatik Aktuell, pages 59–62, 2002.
- [KD98] G. Kindlmann and J. Durkin. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering. In *Proc. of IEEE/ACM Symposium on Volume Visualization*, pages 79–86, 1998.
- [LC87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. of ACM SIGGRAPH*, pages 163–169, 1987.
- [LSW⁺98] A. Luft, M. Skalej, D. Welte, R. Kolb, K. Burk, J. Schulz, T. Klockgether, and K. Voigt. A New Semiautomated, Three-dimensional Technique Allowing Precise Quantification of Total and Regional Cerebellar Volume Using MRI. *Magnetic Resonance in Medicine*, 40(1):143–151, 1998.

Chapter 8

Hybrid Segmentation and Exploration of the Human Lungs

Dirk Bartz¹, Dirk Mayer², Jan Fischer¹, Sebastian Ley²,
Anxo del Río¹, Steffi Thust², Claus Peter Heussel²,
Hans-Ulrich Kauczor², and Wolfgang Straßer³

¹Visual Computing for Medicine Group, University of Tübingen

²Department of Radiology, University Hospital Mainz

³WSI/GRIS, University of Tübingen

Appeared in the Proceedings of IEEE Visualization 2003.

Segmentation of the tracheo-bronchial tree of the lungs is notoriously difficult. This is due to the fact that the small size of some of the anatomical structures are subject to partial volume effects. Furthermore, the limited intensity contrast between the participating materials (air, blood, and tissue) increases the segmentation difficulties.

In this paper, we propose a hybrid segmentation method which is based on a pipeline of three segmentation stages to extract the lower airways down to the seventh generation of the bronchi. User interaction is limited to the specification of a seed point inside the easily detectable trachea at the upper end of the lower airways. Similarly, the complementary vascular tree of the lungs can be segmented. Furthermore, we modified our virtual endoscopy system to visualize the vascular and airway system of the lungs along with other features, such as lung tumors.

Keywords: Tracheo-bronchial tree, segmentation, multi-slice CT, virtual endoscopy

CR Categories: I.3.7: Three-Dimensional Graphics and Realism: Virtual Reality; I.3.8: Application: Virtual Medicine;

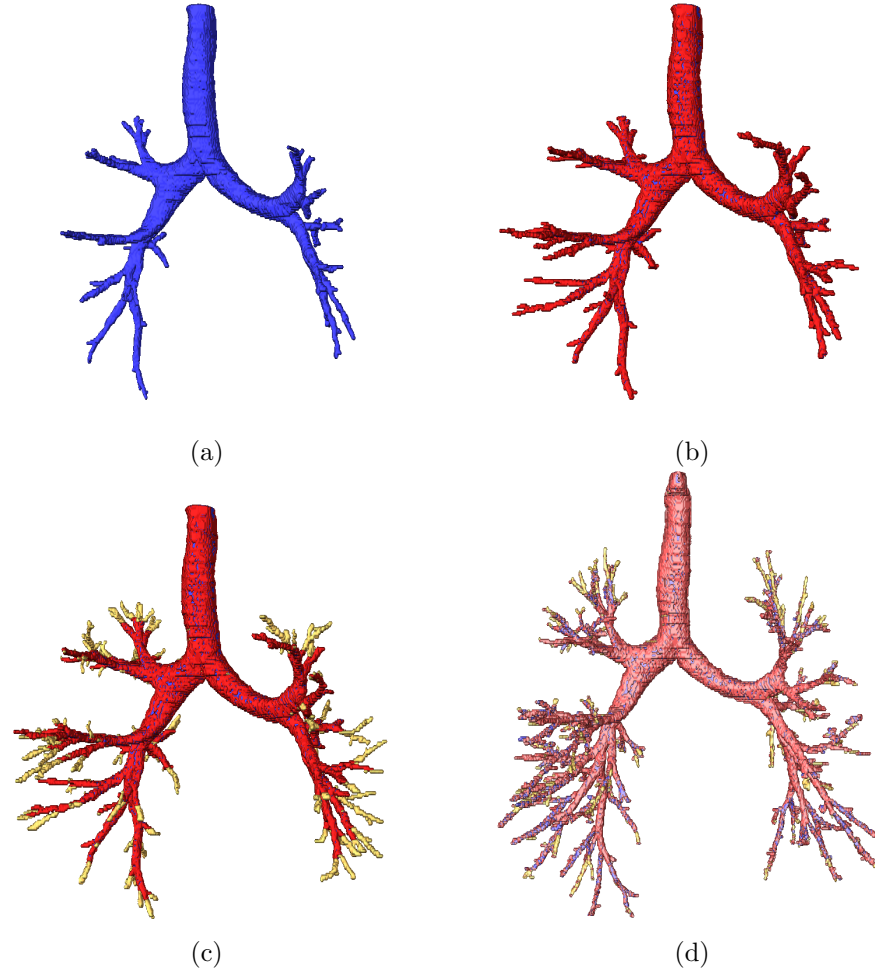
J.3: Life and Medical Sciences: Pulmology, Radiology;

Figure 8.1: Results of three segmentation stages: (a) Region growing, (b) 2D wave propagation, (c) 2D template matching, (d) final segmentation result after six iterations, taking about 76s on a PC with a P3 @ 850 MHz.

8.1 Introduction

The human body consists of several organ systems which control the numerous body functions. Among them are the metabolism, the circulation of the blood and other body fluids (ie., lymphatic fluids), and the respiratory functions. The latter one transports air into the body (inhaling) and removes exhausted air from the body during the exhale. The exchange of the air takes place in the lungs, which are a complementary system of airways and blood vessels. Both systems are supplied through large pipe-like structures which split successively into smaller ones, thus creating the tracheo-bronchial (airways) and blood

vessel tree.

The tracheo-bronchial tree is connected to the outside through the trachea which splits into the main bronchi (left and right lungs) at the main bifurcation. The inhaled air is distributed through the bronchial tree down to the alveoli, where the oxygen/carbon-dioxide exchange between air and blood takes place. The exhausted air (enriched with carbon-dioxide) is then transported back up to the trachea during the exhale. The tracheo-bronchial tree is complemented by a system of pulmonary venous and arterial blood vessels which transports the blood to and from the heart into the lungs.

Several pathologies can jeopardize a sufficient lung function. Among them are tumors, pulmonary embolism, collapse of the lungs (atelectasis), pneumonia, emphysema, asthma, and many more. For a proper diagnosis and treatment, the respective pathologies need to be identified and in some cases quantified. In the case of lung-surgery (ie., for tumor treatment), this information is necessary for the intervention planning where the anatomical relation of diseased bronchi to non-diseased areas is required pre-operatively, ie. to provide a safe distance to essential structures and to determine resectability. In these cases, virtual bronchoscopy with a representation of the vascular structures is in particular useful to represent the spatial and functional relationship in the target area. Another example of a useful application of virtual bronchoscopy is modeling and simulation of inhaled therapies where the exact amount of drug deposition in the lungs is investigated [COP]. Here, virtual bronchoscopy can be used to visualize the local drug deposition in the specific bronchies. Finally, virtual bronchoscopy can be used for the assessment of complex esophagus-trachea syrx that can be difficult to assess in its 3D relationship with regular 2D imaging.

The current gold-standard to identify the respective lung parenchyma and airways is computed tomography (CT) that is performed prior to a bronchoscopy, a tool for inspections of the trachea and central bronchi and deriving tissue samples. Due to the recent technical development improving resolution and scan velocity, multi-slice CT – in connection with virtual bronchoscopy – became a promising alternative to bronchoscopy, if tissue samples are not required. This is amplified by the fact that optical bronchoscopy is limited by smaller, lower airways (third generation and up) or by obstructions. Nevertheless, even smaller structures of the lower airways are extremely difficult to segment from CT datasets, due to leakages, caused by the notorious partial volume effect and due to the lack of sufficient contrast to surrounding tissue or air. However, note that the clinical benefits of virtual bronchoscopy are in principle limited to the data acquisition techniques; features that cannot be detected by it (here multi-slice CT), can also not be exposed by virtual bronchoscopy. Furthermore, the add-on value of a virtual bronchoscopy in contrast to the CT imaging for tumor diagnosis is still disputed.

In this contribution, we introduce a hybrid segmentation method which is based on a pipeline of three segmentation stages to extract the lower airways up to the seventh generation of the lungs (with decreasing sensitivity), starting with the trachea as generation zero. User interaction is limited to the specification of a seed point inside the easily detectable upper airways¹. Similarly, the complementary vascular tree of the lungs can be segmented. Furthermore, we modified our virtual endoscopy system [BSG⁺01] to visualize the vascular and airway system of the lungs along with other features, such as lung tumors, from an endoscopic point of view.

In the following parts of this paper, we briefly review related work in the fields of virtual bronchoscopy and segmentation of the airways in the next section. Afterwards, we will introduce the hybrid segmentation method, including all its pipeline stages in Section 8.3.1. After briefly describing the modified virtual endoscopy system (Section 8.3.6), we present our results and an evaluation of the segmentation quality for a diverse set of datasets from healthy subjects, and subjects who are suffering from diseases of the lungs (Section 8.4). Finally in Section 8.5, we present some conclusions and point to future research directions.

8.2 Related Work

Virtual endoscopy is currently one of the most popular research fields in medical imaging and is slowly moving into the clinical routine. Considering the extensive work developed in the last years on the field of virtual endoscopy, here we will limit ourselves to describe only that work which is directly related to virtual bronchoscopy.

The application of virtual endoscopic techniques to the airways, has been proposed since 1994 [VSH⁺94, MHT⁺94]. Typically it uses helical CT as imaging modality which provides short acquisition times and thus minimizes the motion artifacts due to breathing and heart beating [RKR⁺97]. The rendering of the volumetric data can be performed through two common ways: shaded surface rendering (polygonal rendering) and (direct) volume rendering [RRJAF98]. For shaded surface rendering, a polygonal representation of the surface must be extracted from the volume data. Most of the approaches to virtual endoscopy in general and virtual bronchoscopy in particular, make use of threshold-based region growing methods as basic segmentation technique to obtain the surface voxels [SFH⁺96, RRJAF98, MHS⁺98, FBC01]. Once the surface has been identified, the polygonal representation is generated with the Marching Cubes algorithm [LC87]. However, with these techniques and due to the complexity of the bronchial structure, only the upper bronchi could be properly segmented [SFH⁺96]. Law and Heng use a combination of region growing and centerline extraction to

¹Other parameters are pre-defined and can be modified by the users.

enhance the understanding of the 3D structure of the bronchial tree [LH00]. More recently, Kiraly et al. use a combination of an adaptive 3D region growing, 2D mathematical morphology, and an optional 2D median filter for increasing the robustness of the segmentation algorithm while improving the quality of the results [KHM⁺02]. Also recently, Kitasaka et al. used a regular region growing approach and controlled leaking and bifurcation problems by a complex use of local volume of interest templates that limit the region growing area [KMHT02].

In contrast, Rodenwaldt et al. adopted a different approach for the reconstruction of the 3D surface from CT data [RKR⁺97]. They use the “Navigator” software program (GE Medical Systems, Buc, France), which employs a threshold-based modified ray casting technique to determine and render the walls of the inner lumen.

Also Fung and Heng use direct volume rendering through ray casting for their virtual bronchoscopy application [FH99]. They argue that direct volume rendering has the advantage that no information present in the original data is discarded during a segmentation process. However it comes at the cost of requiring intensive parallel rendering in order to provide interactive frame rates. A different alternative is presented by Wegenkittl et al. where an image based rendering technique is used, based on cubic environment mapping of six video sequences [WVH⁺00]. Although this method produces real time frame rates with a considerable image quality, it is restricted to navigation through a fixed path and presents high pre-processing requirements. During this time intensive pre-processing, the video sequences are computed either using shaded surface rendering, or volume rendering techniques. A further overview of existing visualization techniques applied to airway diseases can be found in [GBAF⁺02].

8.3 Methods for Virtual Bronchoscopy

The segmentation and visualization is based on CT data acquired by a Siemens Somatom Volume Zoom multi-slice CT scanner (acquires four spirals at a time, convolution kernel 50, collimation of 1.25mm and increment of 1mm). For a thorax exam, it typically generates a stack of 250-300 images with a matrix of 512 x 512 pixels resolution and a spacing which varies in the sub-millimeter range (i.e., 0.6mm - 0.7mm) for the pixel distance and 1mm for the slice distance. Based on this dataset, we segment the tracheo-bronchial and the blood vessel tree of the lungs using the segmentation system SegoMeTex, and other structures of interest (ie., a tumor), if available.

Subsequently, we reconstruct the inner surface of these structures and generate additional data-structures needed for virtual endoscopy. Finally, we explore the dataset by a virtual bronchoscopy procedure using the VIVENDI system [BS99].

8.3.1 Segmentation

The segmentation pipeline consists of three stages (Fig. 8.1 and Fig. 8.2). In the first stage, the trachea and central bronchi are segmented using standard 3D region growing methods.

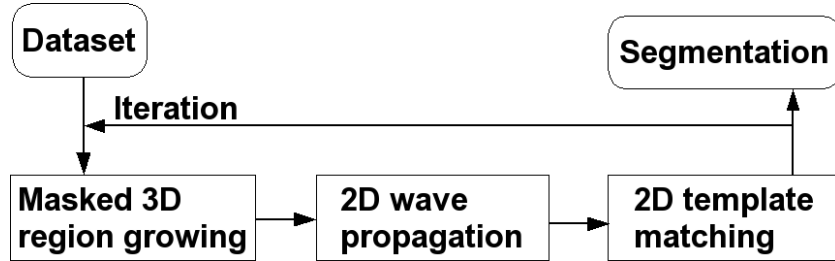


Figure 8.2: Segmentation pipeline.

Partial volume effects and limited resolution of the CT scan (which essentially cause this effect) render this method as not satisfactory for segmentations of further generations of the bronchi, since bordering voxels cannot be sufficiently differentiated from tissue voxels². Therefore, a 2D wave propagation is initiated to complete the upper and central branches. Finally, a 2D template matching procedure is used to segment small lumen, which might be only a single voxel large. A feedback loop of the whole pipeline repeats the stages until no meaningful additions could be made to the previous segmentation (Fig. 8.2). Figure 8.1d shows the final results of five iterations. However, some datasets might require up to 15 iterations.

8.3.2 Masked 3D Region Growing

If we analyze the histogram of the CT dataset of the thorax, we can differentiate the intensity values in the voxels into three categories (Fig. 8.3). All values below -950 HU (Hounsfield Units) can be classified as **definitely airway**, and the intensity values above -775 HU as **non-airway**. All voxels values between this isovalue interval (from -950 HU to - 775 HU) can possibly belong to airway, thus they are classified as **uncertain**.

Based on this analysis, the 3D region growing algorithm extracts all voxels which are definitely airway, starting at the seed point in the trachea. To prevent the leaking into the parenchyma of the lungs in smaller airways (ie., in emphysema), we use a masking technique from texture analysis; if the average gray value of a $3 \times 3 \times 3$ voxel cube centered at the current voxel is within the save range (below -950 HU), we consider this voxel as being part of the airway. Otherwise, the respective voxel is not classified as airway in this stage. While this

²The partial volume effect is averaging low intensity air values with middle intensity bronchi wall or tissue values. The resulting voxel value cannot safely classified as airway or wall.

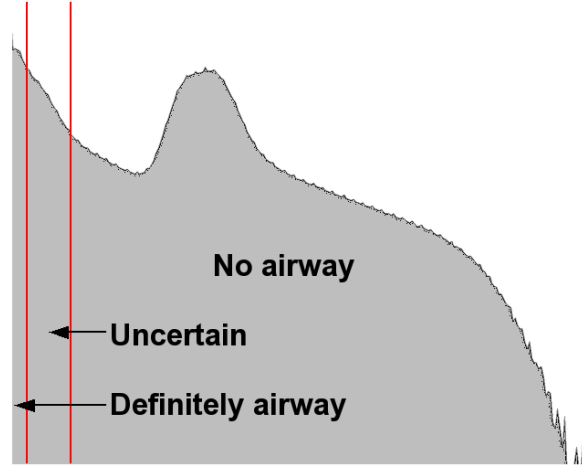


Figure 8.3: Logarithmic histogram of CT dataset.

masking technique prevents leakage, it also impedes the segmentation of smaller bronchi. However, we usually accomplish the segmentation of the bronchial tree up to the fifth generation, whereas the bordering voxels are often not included, since their voxel values belong to the uncertain voxel value interval (see Fig. 8.1a/b and Fig. 8.4).

In the second iteration of the segmentation pipeline, the 3D region growing algorithm runs with the same threshold on the bordering voxels of the previously selected voxels.

8.3.3 2D Wave Propagation

Starting from segmented voxels of the previous step, 2D wave propagation tries to reconstruct bronchi walls within a single CT slice. It starts at each boundary voxel of the airway voxels from 3D region growing and propagates waves to detect the walls of the bronchi (Fig. 8.4b). Voxels at position X in the **uncertain** areas are classified by fuzzy logic rules that consider the density value $V(X)$ (in Hounsfield), the largest local N4 neighborhood (in 2D) gradient $G(X)$, and if voxels in the local N4 neighborhood are already classified as wall pixels (no airway) in a previous wave $W(X)$:

$$f_{wave}(X) = c_v * V(X) + c_g * G(X) + c_w * W(X), \quad (8.1)$$

with $c_v = 1, c_g = 1, c_w = 0.75$.

where $V(X)$ and $G(X)$ are mapped into the closed interval $[1.0, 0.0]$, and $W(X)$ is either 1 – if there is a classified wall pixel in the N4 neighborhood — or 0 – otherwise. Essentially, if $f_{wave}(X) \geq c_{wall}$, the voxel is classified as wall³.

Critical to the wave propagation is the evaluation of the classified airway areas, if they really belong to the airways. To achieve this

³Typically, $c_{wall} = 1.74$.

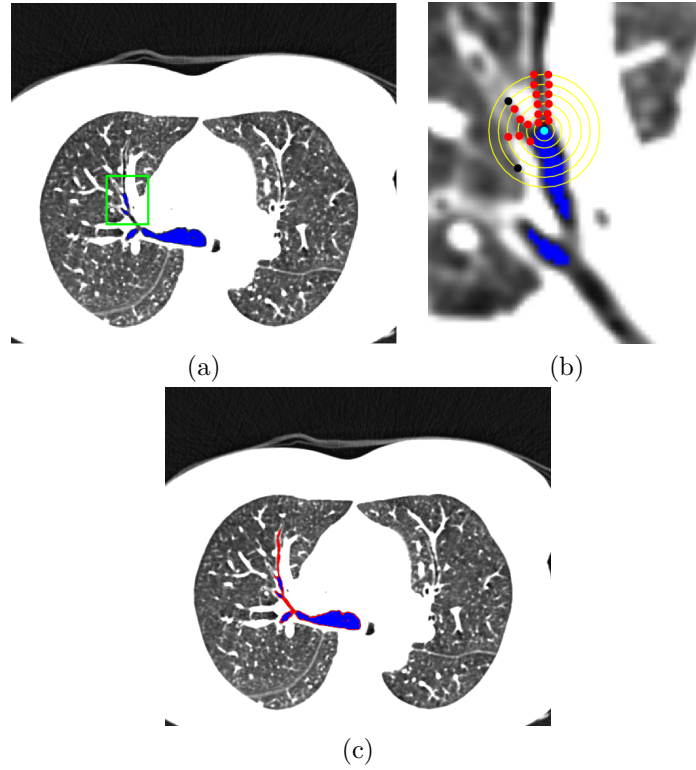


Figure 8.4: Completion of bronchi walls; (a) shows the result of the 3D region growing (blue). The green rectangle marks the zoomed area shown in (b). (b) shows the wave propagation in progress. The cyan boundary voxel is chosen as starting point. The yellow circles mark the propagated waves and the red points mark the airway candidate voxels. The black points failed the leaking test, since the number of voxels of that wave was increasing too fast. Note that the actual waves have a Rhombus-like shape, driven by the N4-neighborhood. (c) shows the completion of that segmentation by 2D wave propagation (red). The voxels (within the body) in the iso-range of **definitely airway** are marked in blue, of the **uncertain** range in black/grey, and of the **no airway** range in white.

goal, the additional voxels segmented by each wave are monitored by a protocol that verifies the shape and size of each bronchus candidate, using a set of default parameters (Fig. 8.4b). As metric, we count the number of voxels selected by the n waves propagating within a plane (BPS_n for BronchusPlainSize), and the wave diameter (WD_n) of the current wave n as the number of selected voxels of wave n . Furthermore, we define the average number of voxels of the first n waves (AWD_n for Average Wave Diameter).

Segments of the tracheo-bronchial tree are identified by sequences of as airway classified voxels in a wave. Figure 8.4b shows two sequences marked by the red points, thus depicting a bifurcation. The shape rules essentially assume that no wave detects segment splits in more than two subsequent segments at a bifurcation. A third segment (of not yet selected voxels) in one 2D wave propagation test (in one slice)

is henceforth considered as leakage into the lungs and is considered invalid. At each bifurcation, the segment identification process starts again recursively.

As closer examination of the datasets showed, two very close bifurcations were never located close enough to be detected as a third segment by the 2D wave propagation, thus they were not falsely identified as leakage.

$$WD_n > d_{max} \quad (8.2)$$

$$WD_n / WD_{n-1} > d_{WD_{ratio}} \quad (8.3)$$

$$BPS_n > d_{size} \quad (8.4)$$

$$AWD_n - AWD_{n-1} > d_{AWD_{current}} \quad (8.5)$$

$$AWD_{max} - AWD_{min} > d_{AWD_{longterm}} \quad (8.6)$$

The size rules limit the growing of the wave propagation⁴. If the diameter of a bronchi candidate exceeds a certain size (Equation 8.2), or if the wave diameter is increasing too fast from the previous wave (Equation 8.3, see also black points in Fig. 8.4b), the respective segment recursion is terminated and the results are considered invalid. Furthermore, if candidates grow spontaneously (while shrinking before) or the overall in plane voxel size BPS_n of the candidate becomes unrealistically large (Equation 8.4), the recursion is again terminated and the results are set invalid. The last two rules (Eqn. 8.5 and 8.6) test the current and long-term growth of the wave front. Specifically, they test if the segments are shrinking (as assumed) or growing. The protocol starts testing after the first three waves, since they frequently show an unstable behavior. During the wave propagation, all invalid results are removed from the segmentation. However, initial correct results (ie., for the first p waves) are preserved.

To follow a bronchus through several slices, virtual waves are propagated in neighboring slices. If one of these virtual waves is similar to the shape and size of the wave propagation in the current slice, another recursive wave propagation in the neighboring slice is initiated. Specifically, the recursive testing of waves in neighboring slice is initiated only for no-branching segments that have classified wall elements from 2D wave propagation. Furthermore, these wall elements may only differ by one voxel to the wall elements of the new neighboring slice segment.

Similar to the first step of the pipeline, 2D wave propagation uses almost the same parameters in the subsequent iteration; only the peripheral bronchi diameter is reduced since the lower airways (higher generations) only grow smaller.

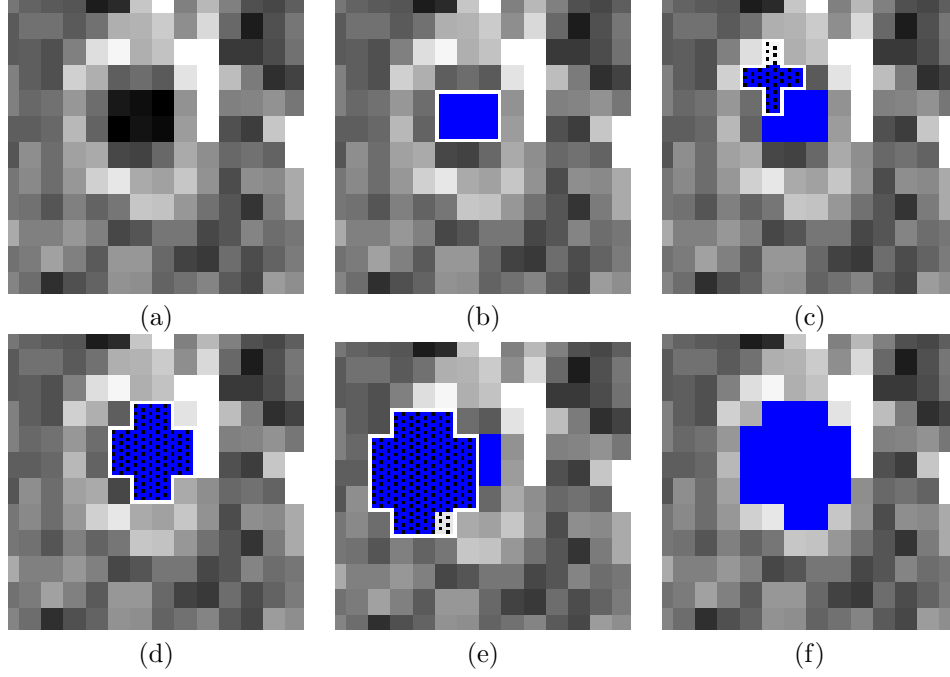


Figure 8.5: Template matching: (a) shows a peripheral airway. 2D seeding is started on pixels of the category **uncertain**. The result is marked blue (b). From there the seeding area templates (white polygon with black dot pattern) are formed and tested on different locations around the seeding. The seeding is repeated on the category uncertain and template (marked by a pattern, (c) to (e)). (f) shows the best result after the classification.

8.3.4 2D Template Matching

Without the used careful validity testing, the previous two stages would leak into the surrounding area, if the airways become too small to be picked up, in particular in areas where the airways might have the size of only one voxel. To select these voxels, but still prevent the leaking, we apply a 2D template matching technique which evaluates the candidate area below templates with the isovalue category **uncertain** (between -950 HU and -775 HU). This stage is organized in two steps; the first step establishes templates that are used in the second step to evaluate the local voxel neighborhood.

First, 2D template matching applies 2D region growing starting from the boundary voxels of the previous segmentations (Fig. 8.5). The thresholds are varied – from the upper threshold of the **uncertain** isovalue interval (-775 HU) – until the number of selected voxels is below the critical limit (ie., 35 voxels), since it can be assumed that they did not leak out. Based on this selected voxel area, circular templates of varying sizes are generated.

In the second step, we apply a 2D region growing and use the tem-

⁴We use $d_{max} = 6.1mm$, $d_{WD_{ratio}} = 1.75$, $d_{size} = 500mm^2$, $d_{AWD_{current}} = 1.13$, $d_{AWD_{longterm}} = 3.0$.

plates to differentiate the thresholds; below the template, we are using the upper **uncertain** threshold (-775 HU), while we are using the original template threshold outside of the template. By moving the templates over the local area, we generate various segmentation candidates (see Fig. 8.5b-e) which are again evaluated by a set of fuzzy rules. This time, we consider the average density value $V'(X)$ of the template area and the (filtered⁵) average gradient $G(X)$ to the surrounding voxels in the N8 neighborhood (within a single slice). The best possible result is then selected and added to the segmentation (Fig. 8.5f and Fig. 8.1d).

$$f_{template}(X) = c_{ave} * V'(X) + c_{grad} * G(X), \quad (8.7)$$

with $c_{ave} = 0.25$ and $c_{grad} = 0.75$.

Here, $V'(X)$ is mapped from [-1024,-775] to [0.0,1.0], and $G(X)$ is mapped into the range [0.0, 1.0]. Illustratively, this means that accepted candidates have a low average density value and a high boundary contrast. The candidate with the largest $f_{template} \geq 0.7$ (Eqn. 8.7) is accepted as airway (Fig. 8.5f and Fig. 8.1d).

However, if the size of the template controlled area is larger than twice as much as for the previous slice, a leak-out is assumed, thus the area of the current slice is assumed invalid.

In the subsequent iterations, voxels which have already been unsuccessfully tested for inclusion, are excluded from template matching. This is mainly to save time – 2D template matching is the single most time consuming stage of the segmentation pipeline – and they usually do not contribute in later iterations.

8.3.5 Segmenting Blood Vessels

To segment the blood vessels, we are using the same three stage segmentation pipeline, based on a dataset that was scanned while injecting an iodine-based contrast agent to enhance the contrast of the blood filled vessels to the neighboring tissue. In addition, the lower (**definitely airway**) and upper (**no airway**) thresholds are adapted to the intensity values of the contrast agent filled blood vessels (130 HU, 250 HU). Note that this time we are looking for a higher intensity structure, in contrast to the very low intensity of the airways.

Similar to the tracheo-bronchial tree, the pulmonary artery tree has the tendency that its diameter is decreasing while descending to the arteriole level of the arterial tree. If the segmentation proceeds to the complementary venous vessel tree, the diameter will increase again, triggering the leakage detectors of the segmentation pipeline. Thus, only the arterial tree is segmented. However, the varying quality of the contrast agent distribution in the blood vessels causes also a varying quality of its segmentation. Therefore, no fixed Hounsfield thresholds – in contrast to the airway segmentation – can be used; they need to

⁵In order to reduce data artifacts, we clamp the gradient range.

be adapted to the specific datasets. Furthermore, additional manual editing may be required to achieve the necessary quality (see also Section 11.5 and Fig. 8.8c).

8.3.6 Virtual Endoscopy

Based on the generated segmentation, we reconstruct the iso-surface of the segmentation (see Fig. 8.7c) using the standard Marching Cubes algorithm [LC87]. Here we use the segmentation as a template to restrict the iso-surface extraction to the volume cells that contain segmented voxels. However, the direct reconstruction from a (binary) segmentation can result in a bumpy appearance, caused by interpolation artifacts due to high density differences between the voxels of the segmentation and the not selected surrounding voxels. Therefore, we add one layer/shell of voxels to the boundary voxels of segmentation that reduces this effect. The application of more continuous distance fields can further reduce this effect. However, it might also reduce reconstructed segmentation details. In some cases (ie., 7th generation bronchi in Fig. 8.7c), this can result into holes that are clearly visible from the outside, but negligible for most of the interior viewpoints. A fall-back to the binary segmentation template can reduce this artifact.

For the virtual endoscopic exploration, we use the reconstructed models (ie., tracheo-bronchial tree and pulmonary blood vessel tree) and render those independently. Usually, we assume a viewpoint inside a bronchus, hence we render its surface semi-transparent and the pulmonary vessels opaque. However, we can also assume a viewpoint inside the blood vessels. Consequently, the transparency parameters would be switched. Independently from the chosen viewpoint location, the enclosing structure is rendered semi-transparent, while the outside structure should be rendered opaque. Note that also other structures (i.e., a tumor) can be added to the scene. In this case, the rendering parameters need to be chosen accordingly.

To determine the visibility of the scene components (tracheo-bronchial tree, pulmonary arteries, and tumor(s)), we render the front-to-back sorted opaque components using visibility driven rendering algorithm as described in [BS99], where we use an octree-based scene decomposition to organize the scene in hierarchical octree blocks, in which the leaf blocks contain the actual scene geometry (see Fig. 8.6 and Table 8.1). Finally, the semi-transparent component – usually the tracheo-bronchial tree – is rendered. If we use the mentioned visibility driven rendering for this component, we trade off rendering quality⁶ and rendering speed [BSG⁺01].

For the actual virtual bronchoscopy, we start in the trachea (Fig. 8.7a) and traverse the tracheo-bronchial tree (Fig. 8.7b) to the target region

⁶The occlusion culling part of the visibility calculation assumes opaque rendering. If we nevertheless render components semi-transparent, we will experience minor popping artifacts [BSG⁺01].

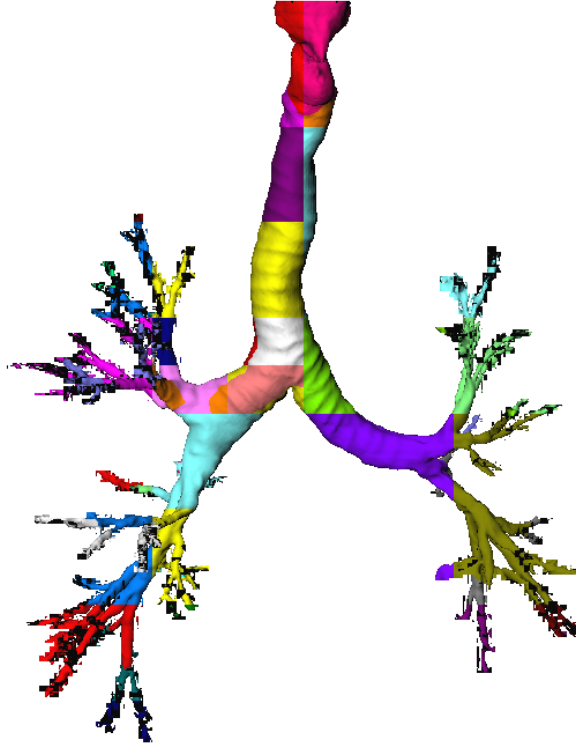


Figure 8.6: Reconstruction of iso-surface of dataset 2. The different octree leaf blocks are coded in different colors.

of interest. Interesting structures or pathologies can be documented for further evaluation. As we will show in the next section, we traverse the tracheo-bronchial tree and see the pulmonary arteries and a tumor of the lungs (Fig. 8.8).

We choose an indirect volume rendering technique for our visualization component due to the required high sustained, full quality rendering performance of more than 30fps. Therefore, we did not use any mesh reduction approach, but visibility driven rendering.

For the same reasons, we opted against a direct volume rendering approach that might be able to generate higher quality images. A texture-mapping-based volume rendering approach will also encounter severe memory problems, since the label (segmentation) volume and the volume dataset itself are required for the proper rendering. Finally, note that in contrast to a common misconception, we still need a segmentation pipeline for the successful segmentation of the structures of interest; a classification by (multi-dimensional) transfer functions is not able to differentiate between the low intensity airways, the empty space between the airways in the lungs, and the space outside of the body.

8.4 Results

The segmentation method was successfully tested on datasets from 22 patients (approximately 6600 dataset slices), who were separated into three groups of normal subjects ($n=8$), emphysema ($n=5$), and lung diseases with increased density, such as pneumonia ($n=9$). The results of the segmentation were compared to segmentations using only the threshold/region growing method – which is still the widely used state-of-the-art for segmentation of the tracheo-bronchial tree – and assessed by an experienced chest radiologist, who also identified false positive segmentation results⁷.

The reliability of the segmentation is statistically measured by the sensitivity and the positive predictive value. The sensitivity describes how reliable those bronchi branches are correctly found (correctly found bronchi branches / existing bronchi branches detected by the radiologist). In contrast, the positive predictive value (PPV) describes the accuracy that **only** existing branches are detected by the segmentation pipeline (correctly found bronchi branches / (correctly found bronchi branches + found false positive bronchi branches)). Therefore, one can interpret the PPV as the specificity (which describes the reliability that negative cases are detected as negative) of leaked (invalid) bronchi branches. The actual specificity could not be determined, since the number of “correct negative” bronchies (no bronchies) cannot be measured.

The presented hybrid segmentation method identified bronchi up to the fifth generation with a sensitivity of more than 85%, up to the sixth generation with a sensitivity of more than 58%, and a PPV of more than 90%. Beyond the sixth generation, the sensitivity drops just below 30%, while the PPV maintains its high level. For emphysema patients, the PPV was slightly reduced due to the damage of the alveoli which increased the leakings into false positive branches. For pneumonia patients, the sensitivity (and to a smaller extent the PPV) are significantly reduced compared to the other datasets. This is due to the increased density in the airways which increases the difficulties of tracking the inner surface tremendously.

While almost all parameters (propagation diameter, etc.) are pre-specified – except the seed point –, they can be optimized for individual datasets. However, the segmentation result is only slightly improved compared to the “standard” setting. We consider this as stability feature of our segmentation approach.

Two of the segmentation pipeline stages are working only on individual slice images. Consequently, the identification of structures which grow orthogonal to the slices is not always possible in the same iteration. As it turns out, however, the masked 3D region growing of the

⁷Usually, an experienced radiologist can identify the tracheo-bronchial tree up to the eighth generation.

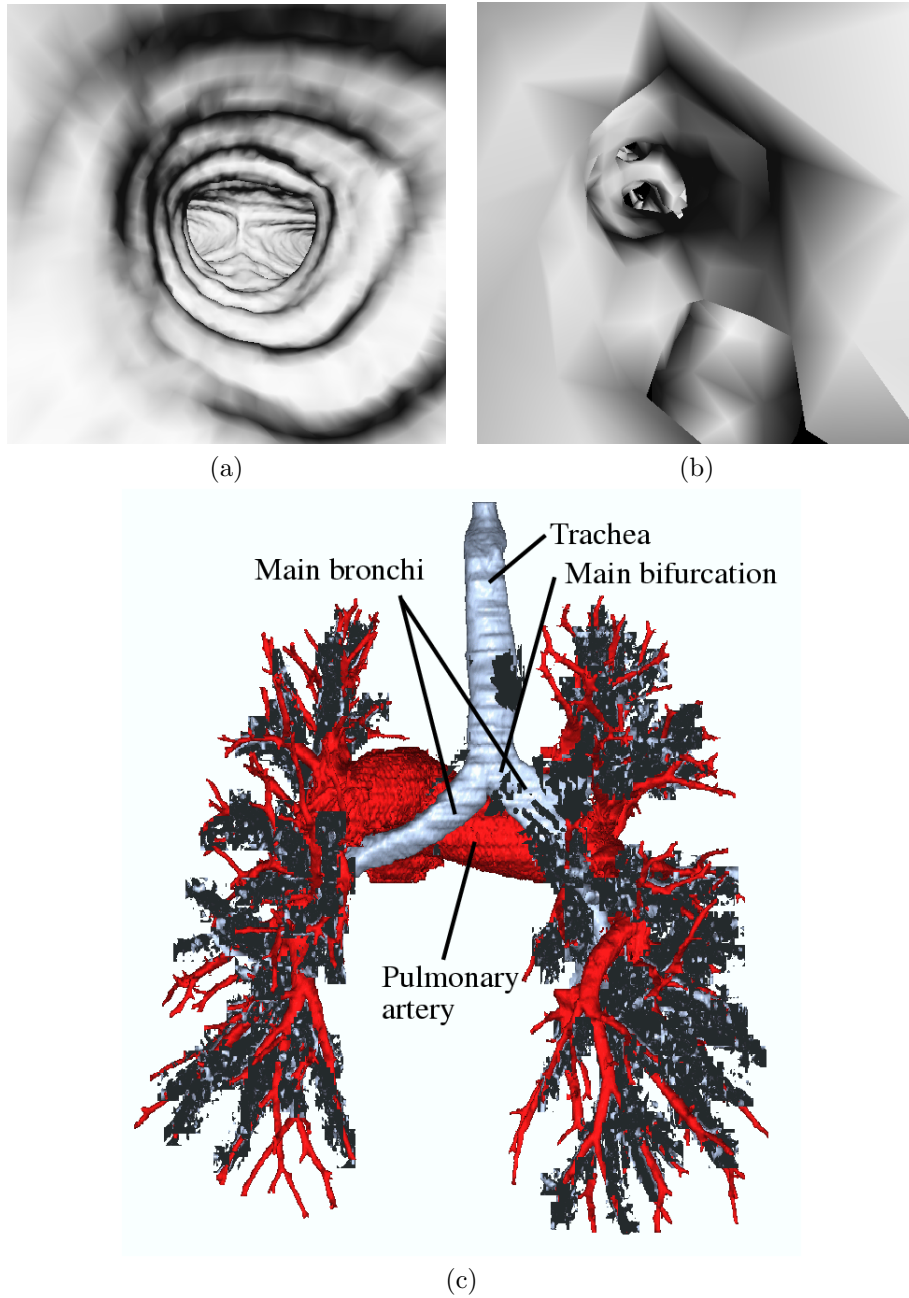


Figure 8.7: Virtual endoscopy of dataset 1: (a) View in trachea down to main bifurcation, (b) approximately the seventh generation of the lower airways, mainly segmented by 2D template matching. The diamond shape interpolation artifacts are due to high zoom factor in this close-up. They are very typical for Marching Cubes reconstructed iso-surfaces close to the dataset resolution (c) shows an annotated outside representation with the tracheo-bronchial tree (grey) and pulmonary arteries (red) from a view behind the patient body (left image side is left patient side).

next iteration – followed by the other 2D pipeline stages – is able to compensate for this effect. As noted before, usually five to seven it-

erations are sufficient for a segmentation of the tracheo-bronchial tree, which is usable in our application context. Only in few cases – ie., like the previously mentioned pathological cases –, the algorithm requires up to 15 iterations. Overall, a typical segmentation process takes between 20 and 100 seconds on a PC with an Intel PIII CPU running at 850MHz. A breakdown of this this time can be found in Table 8.1.

Table 8.1: Breakdown of typical scene complexity, segmentation and rendering costs (dataset 1). Note that due to organizational reasons the timings for the segmentation were generated on a PC with a P3 @ 850 MHz, while the rendering was performed on a PC with a P4 @ 2.8 GHz.

Model Complexity	# Polygons	# Octree Leaf Blocks
Tracheo-bronchial Tree	1,148,032	222
Pulmonary Artery Tree	822,906	116
Segmentation Pipeline	Time	# Voxels [%]
Masked 3D Region Growing	3%	70
2D Wave Propagation	11%	27
2D Template Matching	86%	3
Total Segmentation Costs	76s	100
Rendering Performance	Frames per Second	
Opaque Display	53	
Transparent Display	36	

The iso-surface extraction for dataset 1 (shown in Figure 8.7) generated 1.1M triangles for the tracheo-bronchi tree and 822K triangles for the pulmonary artery tree (Table 8.1). In contrast, the polygonal complexity of the dataset 2 (Figure 8.8) is significantly smaller, mostly due to less refined segmentation results in the upper generations; the tracheo-bronchi tree consists of less than 300K triangles, the pulmonary artery tree of 340K triangles, and the tumor of only 44K triangles. Since we successfully apply a visibility driven rendering approach (Section 8.3.6), we achieve a sufficient rendering performance of the full resolution polygon models.

After the segmentation, we explored a subset of the 22 datasets interactively (more than 30fps on a standard PC⁸, see also Table 8.1) using the virtual endoscopy software [BS99]. The exploration exposed a high quality reconstruction, even of small structures throughout the dataset. In Figure 8.7, we show two snapshots from a regular virtual bronchoscopy. The left image (Fig. 8.7a) shows the endo-view from the trachea looking down to the main bifurcation, where the tracheo-bronchial tree splits into the left and right lungs. Figure 8.7b shows

⁸In the full opaque mode, a Linux PC equipped with an P4 CPU running at 2.8GHz and an ATI Radeon 9700Pro graphics accelerator rendered more than 1.1M triangles (tracheo-bronchial tree is visible only) at 53fps. In the semi-transparent mode, it achieved 36fps rendering the tracheo-bronchial and the pulmonary artery trees of almost 2M triangles. All geometry is arranged in triangle strips, compiled in OpenGL display lists.

the limits of the currently possible segmentation; the virtual endoscope is located approximately in the seventh generation of the lower airways. Due to the high zoom-factors, the typical diamond-shape interpolation artifacts of trilinear interpolation on the voxel cell grid are exposed. Blood vessels were visualized to provide more context information (Fig. 8.7c).

In Figure 8.8, we show a more complex situation, where the surrounding arterial blood vessels are visible through the inner surface of the tracheo-bronchial tree. Figure 8.8c shows a tumor in the right lung in green. At the same time, poor contrast and beam hardening artifacts of the voxels of the pulmonary arteries expose segmentation difficulties; the blood vessel tree is only incompletely segmented (Fig. 8.8c).

8.5 Conclusions and Future Work

In this contribution, we discussed a novel segmentation algorithm which consists of a pipeline of three segmentation stages. The segmentation results were combined with VIVENDI, a virtual endoscopy system to provide a (virtual) bronchoscopic exploration mode. Based on a high-quality segmentation, meaningful features could be visualized. Whereas the segmentation provided good results for a large variety of patients, it encountered problems identifying all airways with pneumonia patients, or pulmonary blood vessel structures.

Virtual bronchoscopy is a valuable tool for the localization and measurement of stenosis for treatment planning. However, mild stenosis, submucosal infiltration, and superficial spreading tumors cannot be identified, since they are usually not picked up by the CT data, and hence they are not visualized through the virtual endoscopy system.

Future Work

Future work will focus on the extension of the last two pipeline stages into 3D. While we do not expect a better segmentation result, 3D operations will reduce the number of required iteration steps, thus it will reduce the required time. Furthermore, we are looking into improving the stability of the algorithms for datasets with poor tissue contrast, like the blood vessels in Figure 8.8.

ACKNOWLEDGEMENTS

This work was supported by the European Commission (IST-1999-14004: "COPHIT", the DAAD/British Council, and by DFG Projects CatTrain and Virtue. We would like to thank Özlem Gürvit of the University Hospital Marburg for helpful discussions and Mike Doggett

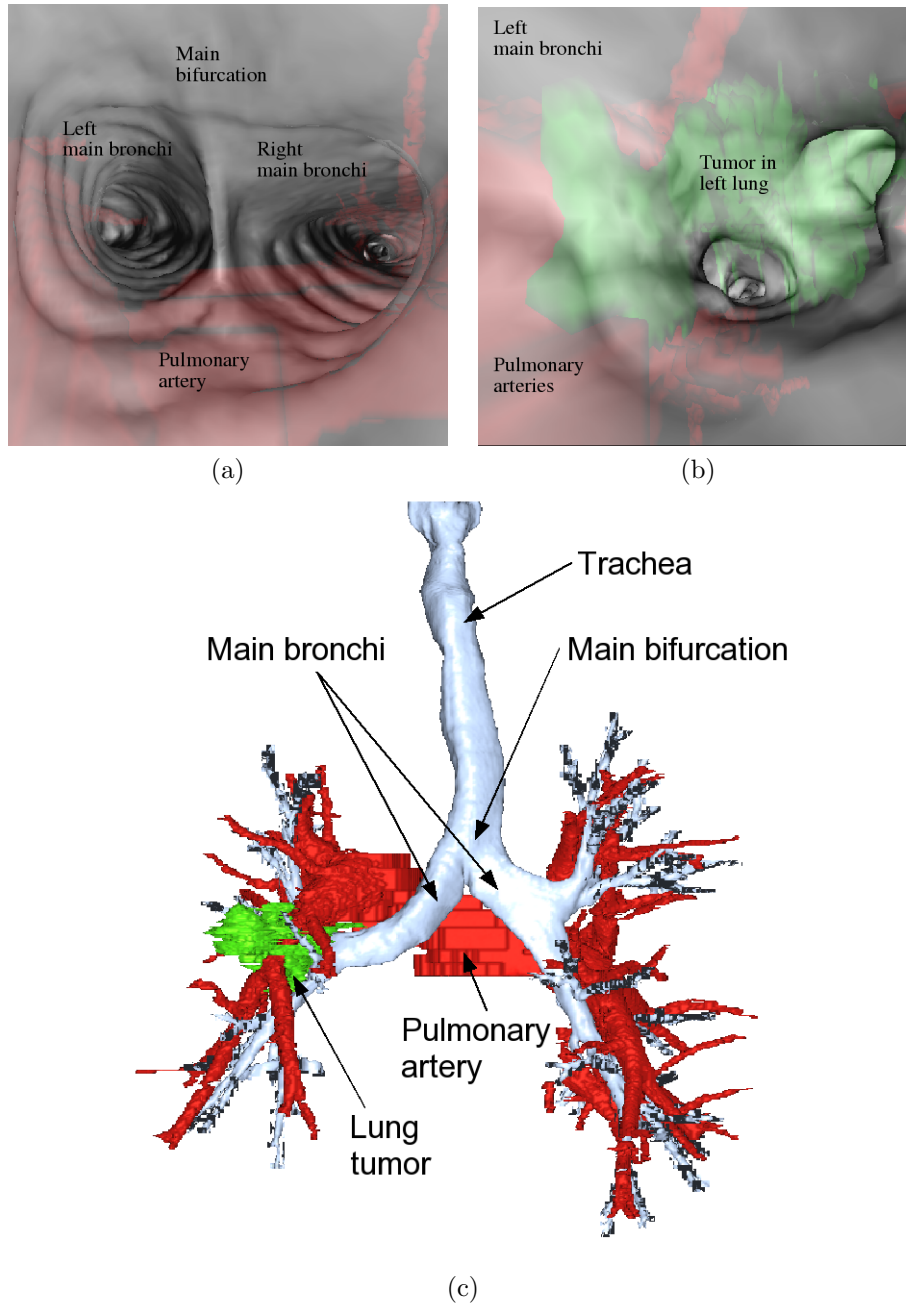


Figure 8.8: Virtual endoscopy of dataset 2: (a) Semi-transparent rendering of view in trachea at main bifurcation. The pulmonary arteries (red) are visible through the inner surface of the tracheo-bronchial tree. (b) shows the same situation further down the bronchi. The tumor (green) encloses the bronchi (grey). (c) shows an annotated outside representation of another dataset with the tracheo-bronchial tree (grey), pulmonary arteries (red), and a lung tumor (green) in the left lung (view from behind the patient).

of ATI providing the most recent Linux drivers for the Radeon 9700Pro in time. Finally, we thank Urs Kanus for proof-reading.

Bibliography

- [BS99] D. Bartz and M. Skalej. VIVENDI - A Virtual Ventricle Endoscopy System for Virtual Medicine. In *Data Visualization (Proc. of Eurographics/IEEE VGTC Symposium on Visualization)*, pages 155–166, 324, 1999.
- [BSG⁺01] D. Bartz, W. Straßer, Ö. Gürvit, D. Freudenstein, and M. Skalej. Interactive and Multi-modal Visualization for Neuroendoscopic Interventions. In *Data Visualization (Proc. of Eurographics/IEEE VGTC Symposium on Visualization)*, pages 157–164, 2001.
- [COP] COPHIT Consortium. EU COPHIT Project: Computer-Optimised Pulmonary Delivery in Humans of Inhaled Therapies. <http://www.cophit.co.uk>.
- [FBC01] G. Ferretti, I. Bricault, and M. Coulomb. Virtual Tools for Imaging of the Thorax. *European Respiratory Journal*, 18:381–392, 2001.
- [FH99] P. Fung and P. Heng. High Performance Computer Simulated Bronchoscopy with Interactive Navigation. In *Proc. of Computer Assisted Radiology and Surgery*, pages 161–165, 1999.
- [GBAF⁺02] P. Grenier, C. Beigelman-Aubry, C. Fetita, F. Preteux, M. Brauner, and S. Lenoir S. New Frontiers in CT Imaging of Airway Disease. *European Radiology*, 12(5):1022–1044, 2002.
- [KHM⁺02] A. Kiraly, W. Higgins, G. McLennan, E. Hoffman, and J. Reinhardt. Three-Dimensional Human Airway Segmentation Methods for Clinical Virtual Bronchoscopy. *Academic Radiology*, 9(10):1153–1168, 2002.
- [KMHT02] T. Kitasaka, K. Mori, J. Hasegawa, and J. Toriwaki. A Method for Extraction of Bronchus Regions from 3D Chest X-ray CT Images by Analyzing Structural Features of the Bronchus. *Forma*, 17(4):321–338, 2002.

- [LC87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. of ACM SIGGRAPH*, pages 163–169, 1987.
- [LH00] T. Law and P. Heng. Automatic Centerline Extraction for 3D Virtual Bronchoscopy. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1935, pages 786–795, 2000.
- [MHS⁺98] K. Mori, J. Hasegawa, Y. Suenaga, J. Toriwaki, H. Anno, and K. Katada. Automated Labeling of Bronchial Branches in Virtual Bronchoscopy System. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1496, pages 870–878, 1998.
- [MHT⁺94] K. Mori, J. Hasegawa, J. Toriwaki, S. Yokoi, H. Anno, and K. Katada. A Method to Extract Pipe Structured Components in Three Dimensional Medical Images and Simulation of Bronchus Endoscope Images. In *Proc. of 3D Image Conference*, pages 269–274, 1994.
- [RKR⁺97] J. Rodenwaldt, L. Kopka, R. Roedel, A. Margas, and E. Grabbe. 3D Virtual Endoscopy of the Upper Airways: Optimization of the Scan Parameters in a Cadaver Phantom and Clinical Assessment. *Journal of Computer Assisted Tomography*, 21(3):405–411, 1997.
- [RRJAF98] J. Remy, M. Remy-Jardin, D. Artaud, and M. Fribourg. Multiplanar and Three-dimensional Reconstruction Techniques in CT: Impact on Chest Diseases. *European Radiology*, 8:335–351, 1998.
- [SFH⁺96] R. Summers, D. Feng, S. Holland, M. Sneller, and J. Shelhamer. Virtual Bronchoscopy: Segmentation Method for Real-Time Display. *Radiology*, 200:857–862, 1996.
- [VSH⁺94] D. Vining, R. Shifrin, E. Haponik, K. Liu, and R. Choplin. Virtual Bronchoscopy (abstract). In *Radiology*, volume 193(P), page 261, 1994.
- [WVH⁺00] R. Wegenkittl, A. Vilanova, B. Hegedüs, D. Wagner, M. Freund, and E. Gröller. Mastering Interactive Virtual Bronchoscopy on a Low-End PC. In *Proc. of IEEE Visualization*, pages 461–465, 2000.

Chapter 9

Scaffolding-based Segmentation of Coronary Vascular Structures

Dirk Bartz¹ and Sarang Lakare¹

¹Visual Computing for Medicine Group, University of Tübingen

²Center for Visual Computing, Stony Brook University

Appeared in the Proceedings of the International Workshop
on Volume Graphics 2005.

The coronary arteries are essential for the proper function of the human heart. However, they are generally difficult to segment in volume datasets separately from the other blood-filled cavities of the heart. The major reason for these difficulties is the lack of sufficient spatial resolution and partial volume effects.

In this paper, we present a method to mark the coronary arteries by a virtual endoscopic traversal. Virtual endoscopy enables a significantly easier visual identification of the blood vessels in comparison to outside views or slice-by-slice examination methods. Furthermore, we use this marking as a scaffold for the actual segmentation of the coronary arteries.

Keywords: Virtual Medicine, Cardio-Vascular System, Segmentation, Virtual Endoscopy

CR Categories: I.3.7: Three-Dimensional Graphics and Realism: Virtual Reality; I.3.8: Application: Virtual Medicine; J.3: Life and Medical Sciences: Cardiology, Radiology;

9.1 Introduction

The vascular system is one of the most important and challenging organ systems in the human body. It ensures circulation and distribution of oxygen, nutrients, hormones, and cells of the immune defense system through all parts of the body. While doing so, it regulates fundamental physiological measures, such as body temperature, hormonal and acid-base balance.

The centerpiece of the vascular system is the heart, which pumps blood through the vascular system. Any disturbance of this system can have a significant impact on body functions. In particular disorders of coronary blood vessels – which supply the heart muscle – can lead to destructions of the respective muscle areas. The resulting malfunction of the heart muscle can cause severe disturbances of the heart rate, which influences the ejection rate of the heart, in particular of the left ventricle (see Section 9.2).

For the timely diagnosis of pathologies of coronary blood vessels, cardio-vascular imaging is, next to other diagnostic means, essential. Unfortunately, imaging of the heart is an extremely challenging task. The frequency of the heartbeat varies usually between 60 and 80 beats per minute, while this number can be higher or lower, depending on body activity. This is too fast for all regular medical imaging modalities to avoid motion artifacts. Thus, scanning is usually triggered by the peak of the ECG, which measures the electric activity of the heart [FO01]. Currently, multi-slice CT scanners, which acquires parts of four to sixteen spirals at a time [KRSF⁺99, FO01], are considered state-of-the-art. Synchronized by the ECG, it scans only in specific parts of the heartbeat cycle to minimize motion artifacts.

The blood-filled cavities in the heart are enhanced by an injected contrast agent. However, effects like beam hardening [Ame05], due to high contrast agent density, can reduce the quality of data significantly. Fortunately, this effect is usually limited to the right atrium of the heart, which is less important for the assessment of coronary arteries. Of more concern is the fact that coronary blood vessels are extremely difficult to segment due to the limited spatial resolution of the scan. In this paper, we propose a marking method to accentuate coronary blood vessels in visualizations of the heart. We use this marked region to generate a segmentation of the coronary arteries. The marking itself is based on the virtual endoscopic exploration of coronary blood vessels.

In the course of this paper, we provide a short overview of the structure and function of the heart to understand the causalities that can lead from a stenosis to a failure of the heart function (Section 9.2). In Section 9.3, we review related work and present our approach for the representation and visualization of coronary blood vessels, and the respective results in Section 9.4. Finally, we conclude in Section 9.5.

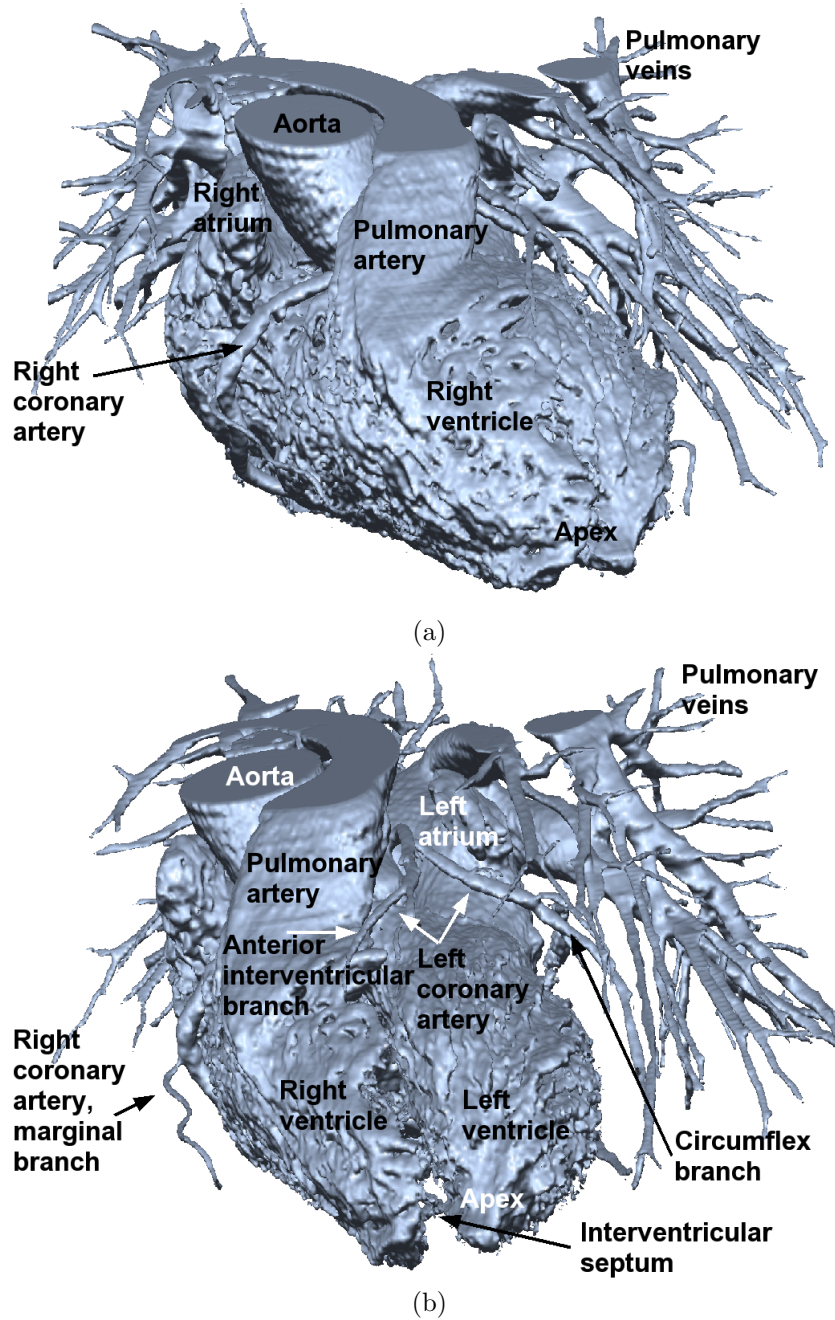


Figure 9.1: Reconstructed blood-filled cavities of the heart; (a) anterior (front) view, (b) sinister (left) view.

9.2 Structure and Function of the Heart

The human heart is divided into four chambers; the left and right atrium and the left and right ventricles, whereas the left and right side of the heart are not directly connected with each other [HR85]. The right part of the heart (see also Fig. 9.1a) – right atrium and right ven-

tricle – are connected through the atrioventricular valve (or tricuspid valve). The atrium receives the venous blood through the venae cavae from the body, whereas the ventricle pumps that blood through the pulmonary valve and through the pulmonary artery to the lungs. After the blood is enriched with oxygen in the lungs, it is received through the pulmonary veins in the left atrium, which is connected through the left atrioventricular valve (or mitral valve) with the left ventricle (see Fig. 9.1b). The ventricle pumps the enriched blood through the aortic valve into the aorta, which distributes the blood to the other body arteries. The (left and right) coronary arteries branch from the aorta and supply the heart muscle with enriched blood through a system of blood vessels that branch from the coronary arteries. They are complemented by the cardiac veins, which collect the blood and merge into a large vein (coronary sinus) that leads to the right atrium.

The conducting system of the heart consists of the sinuatrial node (which generates the electric trigger of the heartbeat), the atrioventricular node (AV node), the AV bundle, which distributes the conduction of the impulse to the ventricles, and at the end, the Purkinje fibers. Due to this arrangement, the contraction of the heart starts at the apex and moves quickly towards the aorta and pulmonary artery, and thus pumps the blood from the ventricles to the respective arteries. A more detailed description of the heart can be found in anatomy text books [HR85].

Cardiac problems often arise from a stenosis of a coronary artery, which is a stricture of this blood vessel that limits the possible blood flow through that part of the vessel and – depending on its severeness – can cause serious cardiac problems, up to an *angina pectoris*, a temporary insufficient supply of oxygen to the heart muscle. Over time, it might even develop into an occlusion of the respective coronary arteries, which causes a serious disturbance of the coronary blood flow. The resulting dysfunctional parts of the respective heart muscle jeopardize the synchronized contraction of the heart, lead to a completely uncoordinated partial contraction, and henceforth reduce the pumping performance of the heart dramatically. This is one of the major causes of heart failure. Other possible causes of cardiac problems are direct damages to the conducting system, which jeopardize the electrical function of the heart, or malfunctions of the numerous valves in the blood flow controlling heart.

9.3 Related Work

The heart is of central interest in the medical community. Consequently, a large number of related publications are available, which focuses on the various aspects from assessment, diagnosis, treatment, imaging, and modeling. In this section, we can only give a brief overview of aspects that vary from imaging of relevant features to the assessment of the

function of the cardio-vascular system, which might be supported by computer graphics methods. These aspects furthermore range from methods with direct clinical application to methods that are still a research topic.

Echo-Cardiography (or Doppler Myocardial Imaging) is a non-invasive ultrasound based technique that is used for the evaluation of the speed of the blood flow, i.e. the blood flow through the valves of the heart, and other assessments [WDGM01]. With the invasive intra-vascular ultrasound imaging, a probe is introduced into the blood vessels of interest [WMR⁺01]. Other methods include MRI Angiography (MRA) or MRI perfusion imaging to assess the blood flow. The blood flow itself can be simulated by CFD methods, based on the derived geometry of the heart [FKL⁺00]. Fenlon et al. described a system that simulates flexible artificial heart valves to evaluate the performance of the various designs [FDW00].

Another direction of research aims at the modeling of the mechanical, electrical, and cell-based functions of the heart, which often is in the domain of biomedical engineering. Park et al. [PMA⁺99] studied the motion (contraction) of the ventricles of the heart to assess the function and Montillo et al. focused on the segmentation of the left and right ventricles [MMA02]. Sachse et al. investigate the electro-mechanical properties of the left ventricles [SSW02] and McCulloch et al. looked into these properties from cellular level up to organ level [MBHN98]. Most of the research done in modeling of the heart is based on the extensive analysis of image data. Sachse et al. for example used a segmentation of the visible male [SSW02]. Virtually all methods that include modeling have in common that up to now, an application to patient data is technically possible, but not practical, in particular for an application in a clinical routine setting.

Essential for a patient oriented presentation of blood vessels are appropriate segmentation methods. Numerous techniques are available for various kinds of blood vessels. Young et al. presented next to their own, shape-based propagation blood vessel extraction algorithm an overview of other approaches [YPW01]. However, almost all of these approaches are targeting blood vessels in areas like the head [BAL⁺99], the lungs [BMF⁺03], the liver [SPSP02], the neck, or the extremities [KWF⁺01], where frequently a graph representation of the vessel tree based on its skeleton is generated [BAL⁺99, SPSP02]. La Cruz et al. presented a model-based parameterization model for the reconstruction blood vessels [CSK⁺04]. Unfortunately, coronary blood vessels pose a significantly different challenge, due to the signal similarities of the blood (and contrast agent) filled heart and blood vessels. Hence, many of the presented methods cannot differentiate adequately between cavities of the heart and the coronary blood vessels. They are therefore not suited for coronary blood vessels.

Recent methods however, have been successfully applied to the coro-

nary blood vessels of the heart. Lorenz et al. proposed a three level segmentation method of the coronary blood vessels using fast-marching wave front propagation [LRSB03]. The three levels (voxels, segments, tree) trim the segmentation to represent the coronary blood vessel tree. This segmentation information can be combined to generate a 3D graph representation of the vessel tree, like it has been done for cerebral blood vessels [BAL⁺99]. In contrast, we are aiming at situations, where (semi-)automatic methods fail to reconstruct coronary blood vessels, due to various artifacts. Sherbondy et al. presented an approach that also employs interaction to improve the segmentation result [SHN03]. They visualize the segmentation result and control similar to the approach presented here segmentation leaks with a mask based on the segmentation seeds.

The volume data for the patient oriented visualization of the relevant features of the heart is acquired by helical CT scanners that are synchronized with an ECG. With the introduction of multi-slice CT scanners, a higher spatial resolution became available that enables better reconstructions of anatomical features [KRSF⁺99, FO01]. These patient datasets can be used for advanced visualization of features of the coronary arteries. In 2001, we used virtual endoscopy to explore features such as stenosis of coronary arteries [BGL⁺01] (see also Fig. 9.4). Schroeder et al. [SKO⁺02] presented a study where virtual endoscopy (virtual coronary angiography) was used to detect lesions of the coronary arteries. In contrast, we use virtual endoscopy here to mark the visited coronary blood vessels for further post-processing.

Please note that although we examined only CT data, the presented methods are not limited to this kind of data and could be used for all other types of volume data of the cardio-vascular system (if the anisotropy is adequately addressed). However, CT scanners are the current state-of-the-art in analyzing the morphology of the heart (our focus), while MRI scanners usually focus on functional aspects.

9.4 Reconstruction of Coronary Blood Vessels

In heart datasets, we can find several different kinds of blood vessels. Easy distinguishable – although still difficult to segment – are the large veins, the aorta, and the pulmonary artery and veins (see Fig. 9.1). Significantly smaller, and hence more difficult to identify is the complex system of coronary blood vessels. Whereas segments of them can be identified while taking a closer look at the data (see Fig. 9.1) or using specific blood vessel tracing algorithms [BFC02], it is challenging to locate the whole system of coronary arteries. Most of the problems are caused by partial volume effect artifacts and the lack of sufficient spatial resolution – the multi-slice CT scan (four slices) provides a slice distance of less than 1.3mm, and a pixel distance of about 0.5mm. While the latter one diminishes some of the boundaries between the ventricles of

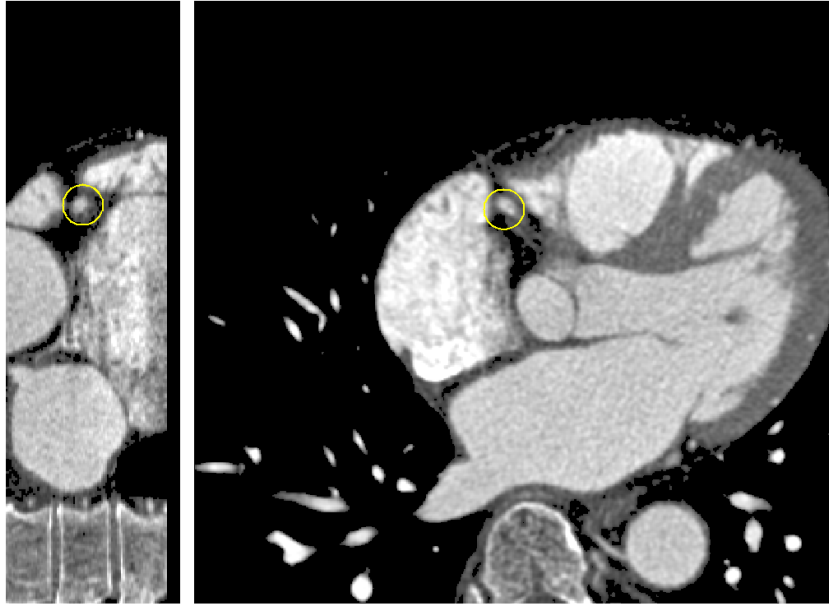


Figure 9.2: Slice views of heart dataset. Sagittal (left) and axial slice (pre-segmented) (right) with coronary arteries - the current position (within the *right coronary artery*) is marked with the circle.

the heart and the coronary vessels (see circled area in Fig. 9.2), the partial volume effect leads to artificial connections or holes that are not present in the scanned subject. These connections or holes are resulting from averaging operations that compensate violations of the sampling theorem at volume reconstruction (undersampling) in regions with high intensity differences between neighboring voxels.

From an endovascular view (see Fig. 9.3 and 9.4) fractions of the affected blood vessels are still visible and usually provide sufficient hints on their shape, which cannot easily be spotted from an outside view. Therefore, the paths of these blood vessels are significantly easier to identify from an endovascular view (see Fig. 9.3). Therefore, we visit the coronary arteries starting from their common starting point in the aorta successively using our virtual endoscopy system [BS99].

9.4.1 Virtual Endoscopy

Our virtual endoscopy system VIVENDI [BS99] enables the virtual exploration of patient volume data from endoscopic viewpoints. First, the voxels of interest are segmented with standard threshold and 3D region-growing methods. Unfortunately, bone structures and contrast agent enhanced blood-filled cavities have approximately the same density values. In some cases, this requires additional cut-off operations to remove false connections. Furthermore, the complex tree-like structure of the pulmonary veins often obstructs the outside view on the coronary

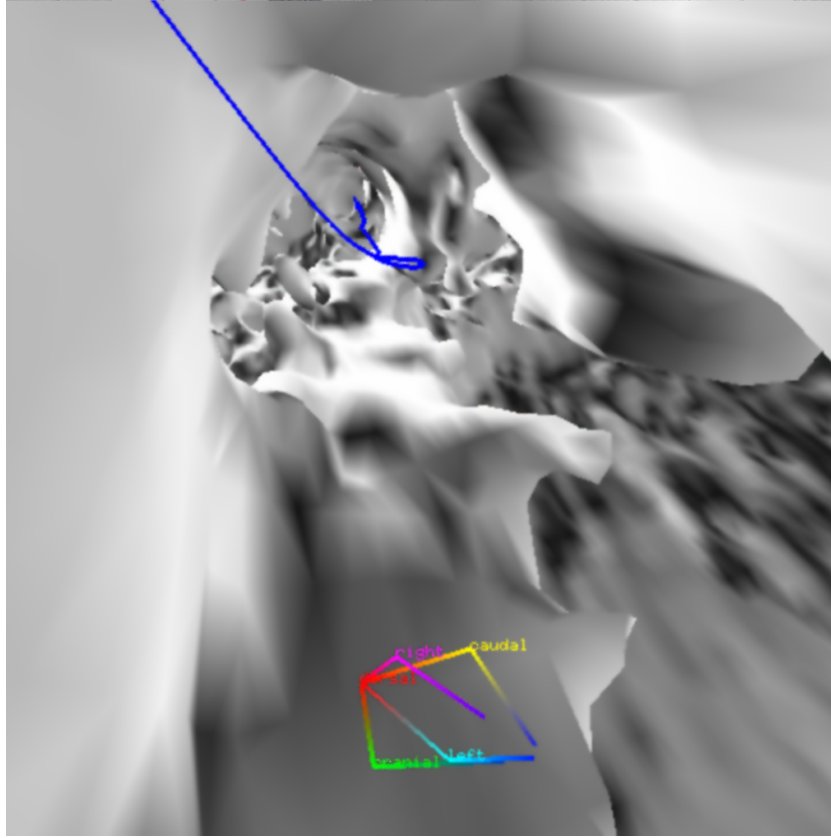


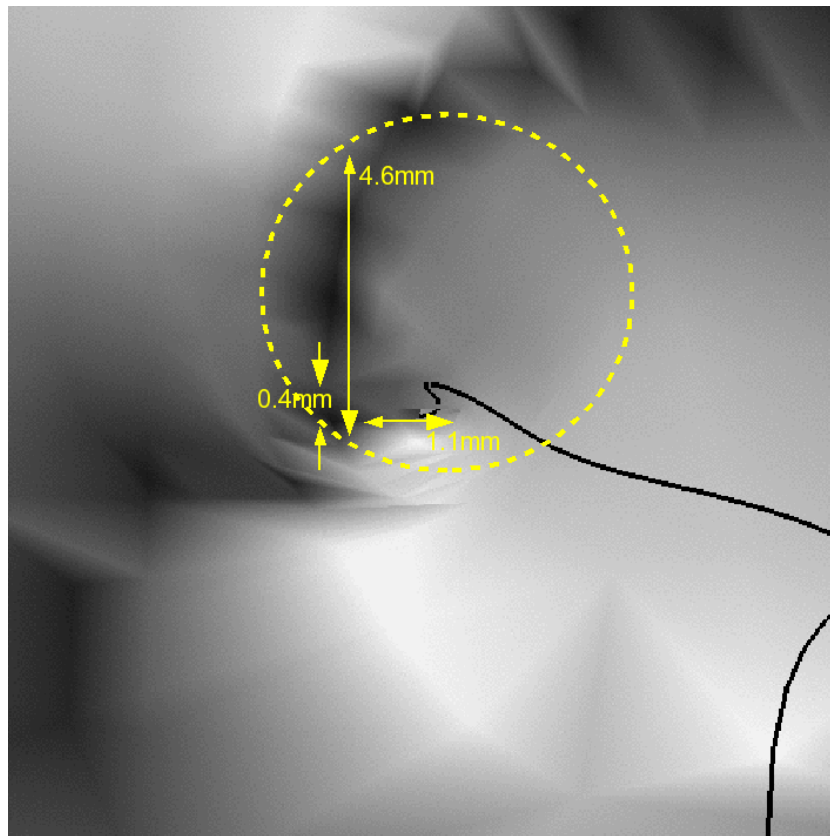
Figure 9.3: Virtual endoscopic view through coronary arteries; the blue/black line is the marked path through the arteries. The reconstruction of the *circumflex branch* of the *left coronary artery* is incomplete due to the lack of spatial resolution and partial volume effects.

arteries. Since they are usually not in the focus of coronary imaging, we also cut-off these structures. In total, the standard segmentation takes about one minute. If additional cut-off operations are required, segmentation takes approximately five to 15 minutes.

Based on the segmentation, the inner surface of the cavities is reconstructed and represented by a polygonal mesh that is organized hierarchically by an octree decomposition of the volume dataset. Also based on the segmentation of the body cavities of interest, we generate a set of potential fields that implement a collision avoidance system¹ [BS99].

Note that this segmentation is extracting all of the blood-filled cavities of the heart, and is therefore not suited for the segmentation of the coronary blood vessels alone.

¹The potential fields also implement user-guidance to a specified target area [BS99]. However, the semi-automatic exploration of the coronary arteries based on this user-guidance would require an already successful segmentation and is thus not practical in our scaffolding approach.



(b)

Figure 9.4: Virtual endoscopic view through coronary arteries; the black line is the marked path through the arteries. The figure shows a stenosis of *left anterior descent (LAD)* and the progress of the stenosis compared to the regular vessel diameter (regular vessel shape marked with yellow circle).

9.4.2 Virtual Coronary Angioscopy

We are using the described virtual endoscopy system to interactively visit the coronary blood vessels. Starting from the aorta – the coronary arteries branch from the aorta –, we successively traverse the coronary arterial system. If the current blood vessel branches, we recursively visit any of its branches and return to the branching point, which was previously bookmarked manually through the scout module of VIV-ENDI [BS99]. In cases where the wall of the blood vessels could not be successfully reconstructed – lacking sufficient spatial resolution – tracing of the direction becomes more difficult (see Fig. 9.3). Although segmentation methods will usually not succeed segmenting this blood vessel, the direction and shape of it are still apparent from an endoscopic viewpoint (Fig. 9.3). Hence, we can continuously traverse the artery. In rare cases, the blood vessel data completely resolves into the neighboring chamber of the heart. Fortunately, such location rarely exhibits sudden changes of direction of that blood vessel, which in turn

simplifies the task of finding the blood vessel, once it appears again.

At the time of our traversal, we mark the locations of our virtual camera, similar to Greek myth of the Minotaur, where Ariadne's thread leads out of its labyrinth. This marking can later be used to visualize the journey of the camera through the vascular system (see Fig. 9.7 and 9.8). Since the camera travels through the volume cells of the patient dataset, we can also generate a voxelized path. For this voxelization, we simply select the nearest-neighbor voxel to the current camera position. The smooth interpolation and the physically-based motion model of the virtual camera by the guided navigation system [BS99] ensures a continuous and smooth voxel path (see Fig. 9.3 and 9.4). However, if the step size of the camera is too large, we need to connect subsequent camera positions to generate a voxel path. This is achieved by attaching line segments between the positions to be connected. Since these line segments are always short (if needed at all), they still preserve a smooth appearance. Afterwards, we assign a voxel environment (or tube) to that path, which represents the scaffold for the later segmentation of the visited blood vessels (see Fig. 9.6b). Using inverse *onion peeling* [HKW⁺95], we add n layers of voxels around the voxel path². n should reflect the typical voxel radius of the coronary blood vessels, possibly considering the rare case of aneurysms. Note that a large n may also include too many voxels into the scaffold that do belong to neighbor blood-filled cavities of the heart.

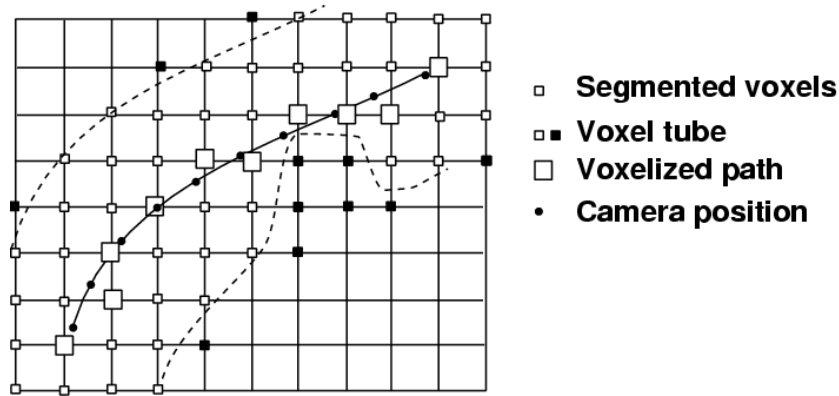


Figure 9.5: Path voxelization and segmentation: The positions of the camera path (solid line) are used for the voxelized path in the blood vessel (dashed lines). Subsequently, a 26-neighborhood voxel tube is created around the voxelized path, representing the segmentation scaffold. A stenosis is located at the right part of the blood vessel and excludes voxels of the scaffold (black cubes) from the segmentation.

A distance field based voxel tube generation [ST94] would ensure a smoother appearance at higher computational costs. Since this is not really required, we opted for the simpler *onion peeling*. Figure 9.5 gives an overview of the path voxelization and segmentation scaffold.

²Each voxel of the next layer is in a 26-neighborhood of the previous layer.

Based on the generated segmentation scaffold, we can apply simple 3D region growing segmentation, using the relevant thresholds to identify contrast agent filled blood vessels. The scaffold acts like a mask that limits the selection of voxels for the segmentation to the voxel tube represented by the scaffold (Fig. 9.5). A stenosis of these blood vessels – a major target feature for coronary imaging – is covered by the segmentation scaffold, but the threshold interval controlled segmentation process also restricts the selection to the voxels within the relevant threshold interval (see black scaffold voxels in Fig. 9.5)³. Overall, our algorithm selects all voxels along the voxel path of the structures visited by the virtual endoscopy system, limited by the segmentation scaffold and the relevant intensity interval of the contrast agent enhanced blood vessels. However, the current approach does not compensate

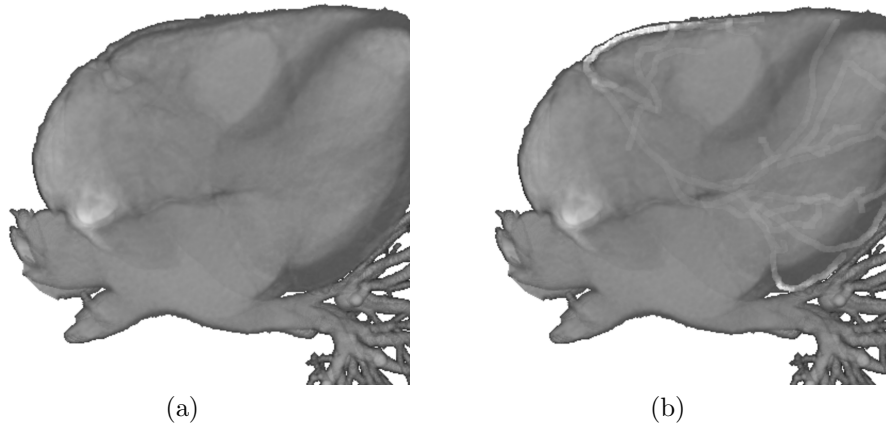


Figure 9.6: Axial view of heart dataset; (a) pseudo x-ray image of heart data without marking, (b) pseudo x-ray image with voxel environment enhanced coronary arteries.

for anisotropic data spacing. Hence, the implicitly uniform expansion of the segmentation scaffold will not ensure an uniform rendering of the vessel segmentation (see Fig. 9.7). It will grow larger into slice direction, where the voxel spacing (slice distance) is larger than within one slice (pixel distance). Fortunately, this effect is not very apparent in multislice CT data, since the slice distance is only two times larger than the pixel distance (0.6mm vs. 1.2mm).

Besides extracting the arterial vascular system of the heart, we can also segment the venous system. However, this system is usually less relevant for the diagnosis of coronary disorders.

³Note that partial volume effects and limited resolution can create artificial connections between neighboring high-contrast areas. The scaffold only limits their influence – resulting in leaking of the segmentation –, it does not completely remove it.

9.4.3 Results

For the actual rendering, the segmented voxels are mapped into an isorange beyond the used isorange. For 8bit data, the scalar data range is downsampled to a smaller range (ie., $[0, 240]$). The remaining value range of $[241, 255]$ in our example) is used for the segmentation, where the maximum value is assigned to the voxel path, whereas the voxel shells around the path has successively decreasing scalar values from $max - 1$ (254) down to $(max-n)$ (241 with $n = 14$). If more than n voxel shells are required, the volume needs to be downsampled into a smaller data range, the voxel shells are assigned to a smaller range, or more bits per voxel are required. In our examples, the voxel shells fit into the specified range.

For high quality visualization of the coronary blood vessels, it is important to maintain the fuzzy boundary around the vessels where they interface with the surrounding objects. A loss of fuzzy boundary can lead to stair-casing artifacts in the rendered images. We extract the vessels with the intensity flipping technique [LK03] and then merge them with the original dataset. This allows us to visualize the smooth extracted blood vessels along with the heart.

The color plates of this paper (Fig. 9.7 and 9.8) demonstrate some of the rendering options with the enhanced coronary blood vessels. Figure 9.7a shows a volume rendered image (using the VIZARD II ray casting hardware [MKW⁺02]) of a heart dataset that is combined with the segmentation of the voxel environment of the marked coronary blood vessels. In Figure 9.7b, we see a close-up of the vessel tree of the left coronary artery, and the respective LAD stenosis. Figures 9.8a and b show images of the same heart, exposing a stenosis in the *left anterior descent* branch of the *left coronary artery*. The yellow lines show the camera paths of the virtual endoscopy traversals through this dataset (Fig. 9.8a). Figure 9.8b shows an overview of the zoom area of (a). Note that these camera paths are used to enhance the coronary blood vessels in Figure 9.7.

Figures 9.3 and 9.4 shows snapshots from virtual endoscopy. In the top image (a), we can see the incompletely reconstructed wall of the circumflex branch of the left coronary artery. Figure 9.4 shows a stenosis of left anterior descent (LAD) from an endoscopic viewpoint.

The total time requirements split into a data preprocessing part, which reconstructs the heart cavities and coronary blood vessels from the CT dataset, the interactive traversal of the coronary blood vessels, and finally the scaffolding-based segmentation. While data preprocessing accounts for approximately 5-15 minutes, depending on the acquisition quality of the dataset, the interactive traversal may take between 10 and 15 minutes depending on the experience of the user and the complexity of the dataset. The final scaffolding-based segmentation is only a matter of seconds.

Overall, this method is tremendously faster than manually performing a slice-by-slice segmentation of the coronary arteries – in particular of small arteries –, but still too time consuming for clinical routine.

9.5 Conclusions and Future Work

In this paper, we presented a non-standard method for the segmentation of coronary blood vessels. Due to the limited spatial resolution, neighboring parts of the heart are commonly very difficult to separate from these coronary blood vessels. We use virtual endoscopy to traverse relevant blood vessels, thus marking them as blood vessels of interest. Based on the marking, we generate a scaffold for the segmentation of these vessels.

The major drawback of this method is the time consuming traversal of the blood vessels itself. Depending on the cardiac experience of the user and the quality of the volume data, it can take a significant amount time, which is usually not available in the daily clinical practice. However, it is still tremendously faster and easier to use – hence more practical – than the manual slice-by-slice segmentation used so far.

Future work will focus on a better representation of the segmentation. Currently, we simply specify the respective voxels and mask the volume dataset with the segmentation. However, a smoother representation can be generated, if we apply smooth voxelization methods, ie. the Vxt library [SK99]. With this approach, we can also adapt to the anisotropic nature of many datasets, in particular datasets from MRI and single-slice CT scanners.

Last but not least, this segmentation approach needs to be validated on phantom data and on a larger number of CT heart datasets.

Acknowledgments

The multi-slice CT datasets are courtesy of Siemens Medical Systems and the Department of Radiology of the University Hospital in Tübingen. We would like to thank Axel Küttner and Andreas Kopp of the Department of Radiology at the University Hospital in Tübingen and Stefan Schaller of Siemens Medical Solutions in Erlangen for providing datasets of the heart. We also would like to thank Özlem Gürvit at the Department of Radiology at the University Hospital in Marburg for helpful discussions, and Urs Kanus of the University of Tübingen for proof reading.

The work has been supported by the project CatTrain of the German Research Council (DFG), the Center of Competence for Minimally Invasive Medicine and Technology Tübingen-Tuttlingen (MITT), and the Department of Neurosurgery of the University of Tübingen.

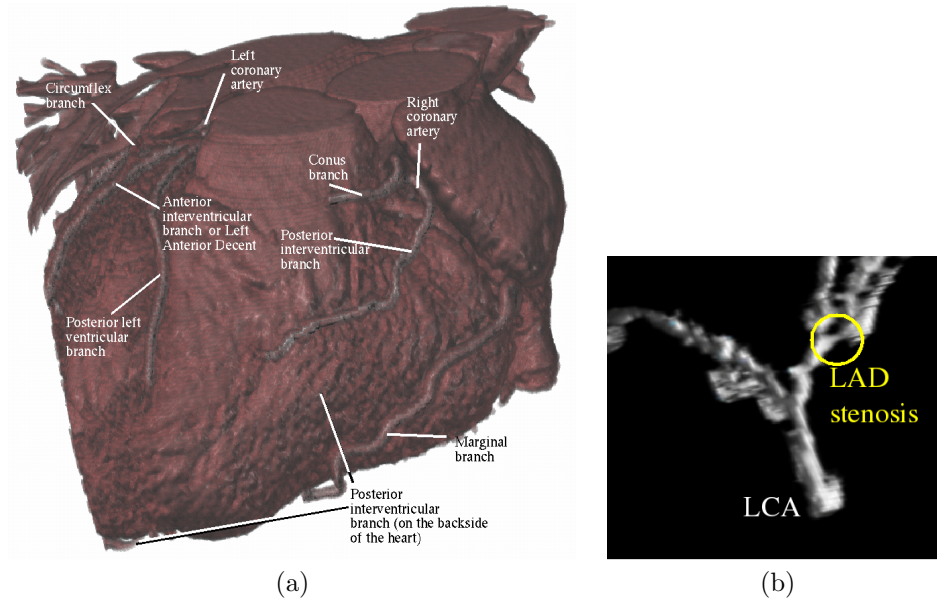


Figure 9.7: Cardio vascular imaging: Volume rendered reconstruction of full tree of coronary arteries; the segmented arteries are classified in **white**. (a) shows an overview image of the heart, (b) shows a close-up of the LAD stenosis in the segmented *Left Coronary Artery (LCA)*. For this image, we removed all the non-artery voxels.

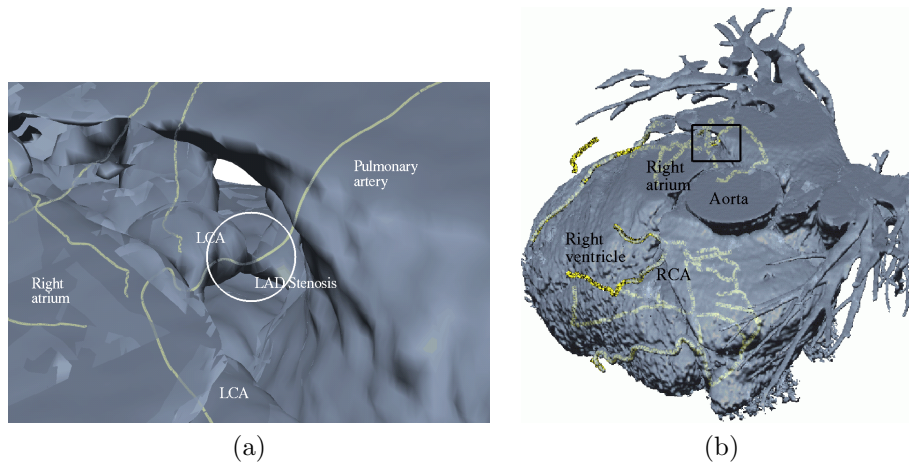


Figure 9.8: Cardio vascular imaging: Stenosis of the *anterior interventricular branch* of the left coronary artery (also known as *RIVA - Ramus Intraventricularis Anterior* or *LAD - Left Anterior Descent*) The coronary arteries are marked in **yellow** in this example for indirect volume rendering. (a) Close-up to LAD segment with stenosis (white circle). (b) Overview - the black rectangle marks the zoomed area.

Bibliography

- [Ame05] Amersham Health. Medcyclopaedia - Encyclopaedia of Medical Imaging. <http://www.amershamhealth.com/med-cyclopaedia/medical/>, accessed 2005.
- [BAL⁺99] E. Bullitt, S. Aylward, A. Liu, J. Stone, S. Mukherji, C. Coffey, G. Gerig, and S. Pizer. D Graph Description of the Intracerebral Vasculature from Segmented MRA and Tests of Accuracy by Comparison with X-ray Angiograms. In *Proc. of Information Processing in Medical Imaging, LNCS 1613*, pages 308–321, 1999.
- [BFC02] K. Bühler, P. Felkel, and A. La Cruz. Geometric Methods for Vessel Visualization and Quantification - A Survey. Technical Report 35, VRVis Wien, Austria, 2002.
- [BGL⁺01] D. Bartz, Ö. Gürvit, M. Lanzendörfer, A. Kopp, A. Küttner, and W. Straßer. Virtual Endoscopy for Cardio Vascular Exploration. In *Proc. of Computer Assisted Radiology and Surgery*, pages 960–964, 2001.
- [BMF⁺03] D. Bartz, D. Mayer, J. Fischer, S. Ley, A. del Río, S. Thust, C. Heussel, H. Kauczor, and W. Straßer. Hybrid Segmentation and Exploration of the Human Lungs. In *Proc. of IEEE Visualization*, pages 177–184, 2003.
- [BS99] D. Bartz and M. Skalej. VIVENDI - A Virtual Ventricle Endoscopy System for Virtual Medicine. In *Data Visualization (Proc. of Eurographics/IEEE VGTC Symposium on Visualization)*, pages 155–166, 324, 1999.
- [CSK⁺04] A. La Cruz, M. Straka, A. Koechi, M. Sramek, E. Gröller, and D. Fleischmann. Non-linear Model Fitting to Parameterize Diseased Blood Vessels. In *Proc. of IEEE Visualization*, pages 393–400, 2004.
- [FDW00] A. Fenlon, T. David, and J. Walton. An Integrated Visualization and Design Toolkit for Flexible Prosthetic Heart Valves. In *Proc. of IEEE Visualization*, pages 453–456, 2000.

- [FKL⁺00] A. Forsberg, R. Kirby, D. Laidlaw, G. Karniadakis, A. van Dam, and J. Elion. Immersive Virtual Reality for Visualizing Flow Through an Artery. In *Proc. of IEEE Visualization*, pages 456–460, 2000.
- [FO01] T. Flohr and B. Ohnesorge. Heart Rate Adaptive Optimization of Spatial and Temporal Resolution for Electrocardiogram-Gated Multislice Spiral CT of the Heart. *Journal of Computer Assisted Tomography*, 25:907–923, 2001.
- [HKW⁺95] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang. 3D Virtual Colonoscopy. In *Proc. of IEEE Symposium on Biomedical Visualization*, pages 26–32, 1995.
- [HR85] W. Hollinshead and C. Rosse. *Textbook of Anatomy*. Harper & Row, Publishers, Philadelphia, USA, 4th edition, 1985.
- [KRSF⁺99] K. Klingenberg-Regn, S. Schaller, T. Flohr, B. Ohnesorge, A. Kopp, and U. Baum. Subsecond Multi-slice Computed Tomography: Basics and Applications. *European Journal of Radiology*, 31(2):110–124, 1999.
- [KWF⁺01] A. Kanitsar, R. Wegenkittl, P. Felkel, D. Fleischmann, D. Sandner, and E. Gröller. Computed Tomography Angiography: A Case Study of Peripheral Vessel Investigation. In *Proc. of IEEE Visualization*, pages 477–480, 2001.
- [LK03] S. Lakare and A. Kaufman. Anti-Aliased Volume Extraction. In *Data Visualization (Proc. of Eurographics/IEEE VGTC Symposium on Visualization)*, pages 113–122, 2003.
- [LRSB03] C. Lorenz, S. Renisch, T. Schlathölter, and T. Bülow. Simultaneous Segmentation and Tree Reconstruction of the Coronary Arteries in MSCT Images. In *Proc. of SPIE Medical Imaging*, volume 5031, 2003.
- [MBHN98] A. McCulloch, J. Bassingthwaite, P. Hunter, and D. Noble. Computational Biology of the Heart: From Structure to Function. *Progress in Biophysics and Molecular Biology*, (69):153–155, 1998.
- [MKW⁺02] M. Meißner, U. Kanus, G. Wetekam, J. Hirche, A. Ehlert, W. Straßer, M. Doggett, P. Forthmann, and R. Proksa. VIZARDII: A Reconfigurable Interactive Volume Rendering System. In *Proc. of Eurographics/SIGGRAPH Workshop on Graphics Hardware*, pages 137–146, 2002.

- [MMA02] A. Montillo, D. Metaxas, and L. Axel. Automated Segmentation of the Left and Right Ventricles in 4D Cardiac SPAMM Images. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 620–633, 2002.
- [PMA⁺99] J. Park, D. Metaxas, L. Axel, Q. Yuan, and A. Blom. Cascaded MRI-SPAMM for LV Motion Analysis During a Whole Cardiac Cycle. *International Journal of Medical Informatics*, 55(2):117–126, 1999.
- [SHN03] A. Sherbondy, M. Houston, and S. Napel. Fast Volume Segmentation With Simultaneous Visualization Using Programmable Graphics Hardware. In *Proc. of IEEE Visualization*, pages 169–176, 2003.
- [SK99] M. Sramek and A. Kaufman. Alias-free Voxelization of Geometric Objects. *IEEE Transactions on Visualization and Computer Graphics*, 3(5):251–266, 1999.
- [SKO⁺02] S. Schroeder S, A. Kopp, B. Ohnesorge, H. Loke-Gie, A. Küttner, A. Baumbach, C. Herdeg, C. Claussen, and K. Karsch. Virtual Coronary Angioscopy Using Multislice Computed Tomography. *Heart*, 87:205–209, 2002.
- [SPSP02] D. Selle, B. Preim, A. Schenk, and H. Peitgen. Analysis of Vasculature for Liver Surgery Planning. *IEEE Transactions on Medical Imaging*, 21(11):1344–1357, 2002.
- [SSW02] F. Sachse, G. Seemann, and C. Werner. Modeling of Electro-Mechanics in Left Ventricle. In *Computers in Cardiology*, pages 705–708, 2002.
- [ST94] T. Saito and J. Toriwaki. New Algorithms for Euclidean Distance Transformation of an N-Dimensional Digitized Picture with Applications. In *Pattern Recognition*, pages 1551–1565, 1994.
- [WDGM01] I. Wolf, R. DeSimone, G. Glombitza, and H. Meinzer. EchoAnalyze - A System for Three-Dimensional Echocardiographic Visualization and Quantification. In *Proc. of Computer Assisted Radiology and Surgery*, pages 902–907, 2001.
- [WMR⁺01] A. Wahle, S. Mitchell, S. Ramaswamy, K. Chandran, and M. Sonka. Visualization of Human Coronary Arteries with Quantification Results from 3-D and 4-D Computational Hemodynamics based upon Virtual Endoscopy. In *Proc. of Computer Assisted Radiology and Surgery*, pages 877–882, 2001.

- [YPW01] S. Young, V. Pekar, and J. Weese. Vessel Segmentation for Visualization of MRA with Blood Pool Contrast Agent. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 491–498, 2001.

Part V

Navigation

Chapter 10

A Navigated Virtual Endoscopy System for Oral-, Maxillo-Facial Surgery and Neurosurgery

Dirk Bartz¹, Jan Fischer¹, Angel del Río¹, Dirk Freudenstein², and Jürgen Hoffmann³

¹Visual Computing for Medicine Group, University of Tübingen

²Department of Neurosurgery, University Hospital Tübingen

³Department of Maxillo-Facial Surgery, University Hospital Tübingen
Appeared in the Proceedings of 3D Modeling 2003.

The combination of minimally-invasive endoscopic methods of surgery with techniques from virtual endoscopy and image-guided navigation promises a significant improvement of the three-dimensional spatial orientation during surgical procedures. In particular the better representation of risk structures like blood vessels or nerve fibers enables a better planning, and hence reduces the risk of serious complications during an intervention.

Keywords: Endoscopy, virtual Endoscopy, image-based Navigation, Computer Graphics, virtual Medicine.

10.1 Introduction

The virtual endoscopy system VIVENDI has been developed over the past years by the working group for “Visual Computing in Medicine” at the WSI/GRIS of the University of Tübingen, in collaboration with the departments of neuroradiology and neurosurgery of the University Hospital Tübingen. VIVENDI is an interactive system which can be used

for the visual simulation of endoscopic examination. Furthermore, it enables a truly three-dimensional representation of the respective organ systems.

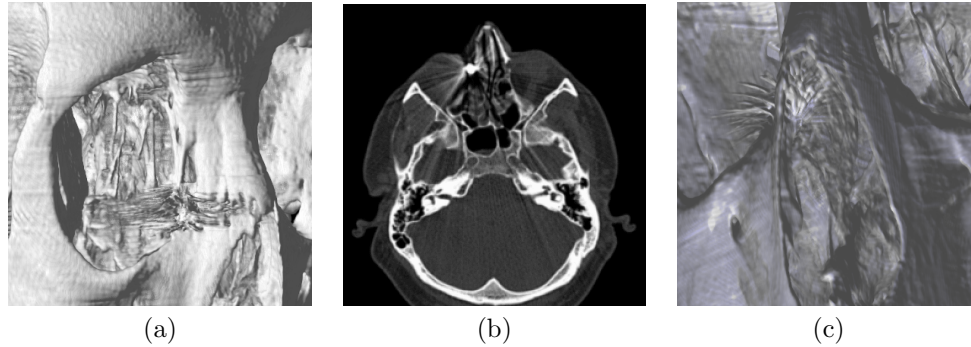


Figure 10.1: Bullet of an air rifle in the ethmoid on the right side. Using image-based navigation, it could be removed by nasal endoscopy, while an extensive bicoronal approach could be avoided. (a) View on the right facial skull (bullet is marked); (b) Slice from multi-slice CT with bullet (arrow); (c) view through nose skeleton on bullet (marked).

Currently, VIVENDI is in a pre-clinical test phase, where it is used mainly for the planning of the actual intervention. Furthermore, it can be used for measurements of anatomical structures and for the volumetric evaluation of various cavities. In this context, VIVENDI has been used for the examination of the colon [HMK⁺97], of the ventricular system of the brain [BS99, FBSD01], as well as for the vascular systems of the heart [BGL⁺01] and the cerebral nervous system [BSSW99].

Systems and techniques of image-based navigation have been used at the University Hospital of Tübingen for quite some time. In particular, it has been used in the departments of neurosurgery [DDSG94, FBSD01] and in the oral and maxillo-facial surgery [KHea00]. During the intervention, the position and orientation of the referenced optical endoscope are mapped into a pre-operative acquired stack of images (volume dataset) of a medical 3D-scanner (i.e., CT or MRI). Navigated optical endoscopic procedures are used for diagnostic and therapeutic interventions, i.e., in the area of the cerebral ventricular system [DDSG94, FBSD01], the treatment of fracture reduction and osteosynthesis of the facial bones [KHea00], or for the navigation-based removal of foreign bodies (see Fig. 10.1). Catheter-based interventions are also performed in interventional neuroradiology, however no truly three-dimensional spatial representation of anatomical cavities are provided for interventions of this kind. Furthermore, the influence of the brain shift is largely ignored, since it either has only a limited impact, or the focus of the orientation is on the bone structures that are not affected by it.

The goal of this project is the efficient combination of virtual, and navigated-optical endoscopy for the clinical application for neuro- and

maxillo- facial surgical interventions. This combination enables the three-dimensional visual representation of the organ system – based on the pre-operative 3D-scan of the patient –, coupled with the respective optical representation through the endoscope. In case a minor bleeding reduces the optical visibility, we keep the spatial orientation of the OR situ. Therefore, the virtual representation (Fig. 10.2) can still ensure the safe retraction of the endoscope. With the integration of multi-modal data from CT, MRI, or 3D rotational angiography, additional information can be overlaid on the virtual representation (Fig. 10.2), i.e., blood vessels or nerve fibres. Serious complications, such as an injury of major blood vessels – which are usually not visible through optical endoscopy –, can be avoided using this information. Furthermore, if unusual anatomy or pathological variations increase the difficulties of orientation or identification of structures, a virtual endoscopic examination can access these areas which are not accessible for the optical endoscope.

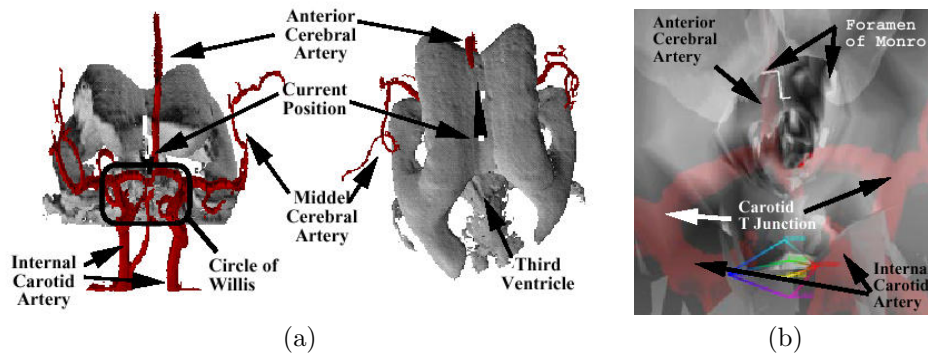


Figure 10.2: Multi-modal representation of the cerebral ventricular system with participating arterial blood vessels. Left: Front and top overview representation; right: virtual-endoscopic representation of the third ventricle.

10.2 Material and Methods

Figure 10.3 shows an overview of the project architecture. The VIVENDI-based planning of the intervention is based on a volume dataset of the patient (MRI in this example). Initially, the optical endoscope is referenced to the volume dataset, which enables the current 2D navigated imaging through the navigation system. This system provides the orientation and position data of the navigated endoscope to VIVENDI, which uses the information to synchronize the virtual view with the optical view. With this synchronized view, the surgeon can guide the endoscope to the target.

To achieve this goal, various problems need to be addressed. The combination of navigation and virtual endoscopy requires the real-time

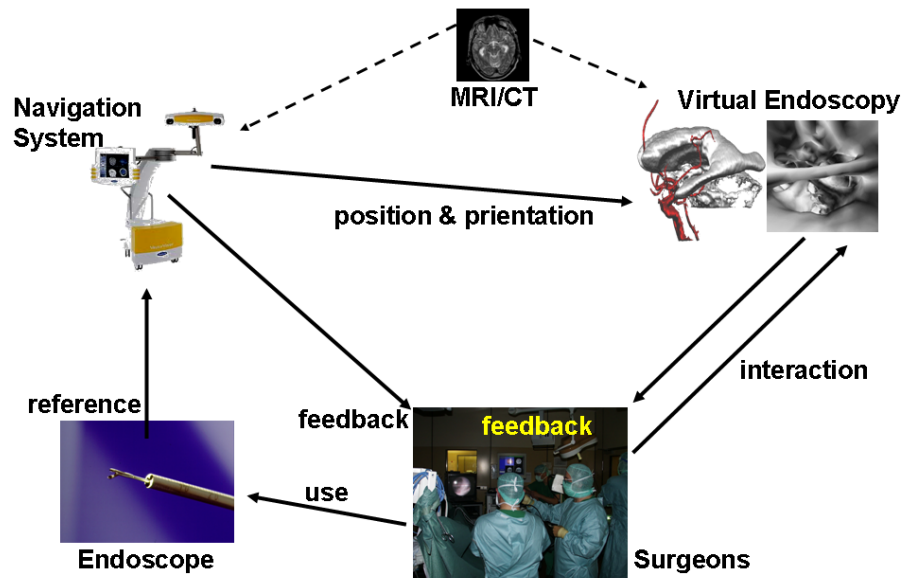


Figure 10.3: Overview of the architecture of the integration of virtual and navigated endoscopy.

representation of the respective organ structures. VIVENDI uses a visibility driven rendering, which removes currently non-visible parts of the geometric representation of the organs. Other problems include the smoothing of jitter in the data transfer of the orientation- and position information, segmentation tasks, and the registration of the various datasets. A special focus is also the integration of the roll of the endoscope, which is not always detected by the navigation system. Finally, the improvements of recent scanner technology require new data management strategies to cope with the increased size of the data. In particular, data compression and out-of-core methods can be used to address these problems.

10.3 Discussion

In summary, the combined representation of navigated, optical endoscopy with virtual endoscopy can significantly increase the share of useful information. This in turn reduces the risks of serious complications. Furthermore, the improved planning and navigation possibilities increase the range of potential endoscopic interventions.

Acknowledgements

This project is financed by DFG CatTrain, DFG VIRTUE, and the EU IST programme.

Bibliography

- [BGL⁺01] D. Bartz, Ö. Gürvit, M. Lanzendörfer, A. Kopp, A. Küttnert, and W. Straßer. Virtual Endoscopy for Cardio Vascular Exploration. In *Proc. of Computer Assisted Radiology and Surgery*, pages 960–964, 2001.
- [BS99] D. Bartz and M. Skalej. VIVENDI - A Virtual Ventricle Endoscopy System for Virtual Medicine. In *Data Visualization (Proc. of Eurographics/IEEE VGTC Symposium on Visualization)*, pages 155–166, 324, 1999.
- [BSSW99] D. Bartz, W. Straßer, M. Skalej, and D. Welte. Interactive Exploration of Extra- and Intracranial Blood Vessels. In *Proc. of IEEE Visualization*, pages 389–392, 547, 1999.
- [DDSG94] F. Duffner, W. Dauber, M. Skalej, and E. Grote. A New Endoscopic Tool for the CRW Stereotactic System. In *Stereotactic and Functional Neurosurgery*, volume 67(3-4), pages 213–217, 1994.
- [FBSD01] D. Freudenstein, D. Bartz, M. Skalej, and F. Duffner. A New Virtual System for Planning of Neuroendoscopic Interventions. *Computer Aided Surgery*, 6(2):77–84, 2001.
- [HMK⁺97] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual Voyage: Interactive Navigation in the Human Colon. In *Proc. of ACM SIGGRAPH*, pages 27–34, 1997.
- [KHea00] M. Krimmel, J. Hoffmann, and D. Gülicher et al. Bedeutung der Videoendoskopie für die operative Versorgung von Jochbeinfrakturen. In *28. Jahrestagung der Deutschen Gesellschaft für Plastische und Wiederherstellungschirurgie*, 2000.

Chapter 11

Medical Augmented Reality based on Commercial Image Guided Surgery

Jan Fischer¹, Markus Neff², Dirk Freudenstein³, and Dirk Bartz¹

¹Visual Computing for Medicine Group, University of Tübingen

²BrainLAB AG, Heimstetten

³Department of Neurosurgery, University Hospital Tübingen

Extended version of the paper, which appeared in the Proceedings of the Eurographics Symposium on Virtual Environments 2004.

Utilizing augmented reality for applications in medicine has been a topic of intense research for several years. A number of challenging tasks need to be addressed when designing a medical AR system. These include the import and management of medical datasets and preoperatively created planning data, the registration of the patient with respect to a global coordinate system, and accurate tracking of the camera used in the AR setup, as well as the respective surgical instruments. Most research systems rely on specialized hardware or algorithms for realizing augmented reality in medicine. Such base technologies can be expensive or very time-consuming to implement. In this paper, we propose an alternative approach of building a surgical AR system by harnessing existing, commercially available equipment for image guided surgery (IGS). We describe the prototype of an augmented reality application, which receives all necessary information from a device for intraoperative navigation. It is shown that all of the afore-

mentioned problems can be solved by image guided surgery. Furthermore, we propose a new method for transforming pose information of a tracked surgical instrument into camera pose information and the necessary one-time calibration step based on conventional marker tracking.

CR Categories: H.5.1: Information Interfaces and Presentation: Artificial, augmented, and virtual realities; I.3.7: Three-Dimensional Graphics and Realism: Virtual Reality; I.3.8: Application: Virtual Medicine; J.3: Life and Medical Sciences: Medical information systems;

11.1 Introduction

Augmented reality (AR) denotes a set of techniques aimed at combining real and virtual image elements in real-time. One particular aspect of AR is the requirement of correct three-dimensional placement and orientation of virtual graphical objects. Two main types of augmented reality can be distinguished. Whereas optical see-through systems use special translucent display devices, video see-through AR is based on mixing a live video stream from a camera with the graphical scene elements. An overview of augmented reality is given by Azuma in [Azu97].

11.1.1 Medical Augmented Reality

The planning, preparation, and support of medical treatment has always been among the most important applications of augmented reality. Surgical interventions in particular have been in the focus of many research efforts. Modern methods of surgery, like minimally-invasive procedures, can gain significantly from added information visualization and augmented views to balance the increased difficulty of these interventions.

A complete AR application for ultrasound-guided needle biopsies has been described by State et al. [SLH⁺96]. The system offers stereoscopic rendering and accurate tracking of both the user and the ultrasound probe, but relies on a number of proprietary technologies. These include a hybrid optical-magnetic tracker for the user's head and a mechanical arm for probe registration.

More recently, several research groups have worked on developing augmented reality support for various medical application scenarios. Examples include the Medarpa project [SSW02, SS04], which uses a special translucent display device mounted on a swivel arm. Here tracking is performed using a hybrid scheme based on active infrared and electromagnetic components. The AR system prototype of Splechtna et al. [SFW02] consists of an optical tracking system comprising two cameras and a head-mounted display. It is capable of augmenting the

surgeon's view with preoperative volume data or live ultrasound images. A semi-transparent mirror display mounted directly over the patient is used for reality augmentation in the ARSys-Tricorder described by Goebbels [Goe03]. It also provides stereoscopic rendering, with infrared cameras being used for tracking the surgeon's head and components of the AR system. Scheuering et al. have presented a two-stage approach to registration for augmented reality [SRSB⁺02]. In addition to rigid tracking using a hybrid optical-magnetic system, a non-linear deformation of patient data is performed based on image processing methods. A concept for supporting navigation in liver surgery with AR has been described by Suthau et al. [SVH⁺02]. A similar application scenario has also been examined by Bornik et al. [BBR⁺03]. Their system uses head-mounted displays for visualization and relies on a specialized optical camera system for tracking.

11.1.2 Drawbacks of Existing Systems

One significant drawback of many existing methods for realizing augmented reality in medicine is their reliance on specific hardware. Such devices, like magnetic tracking systems or specialized displays, often are expensive and can require tedious setup procedures. Many of them have originally been designed for applications in industry or virtual reality and may not be well suited for medical scenarios. In particular stereo shutter glasses and head mounted displays are often considered to be too bulky, and they may deteriorate the user's perception of the surroundings. They are thus not well accepted for medical applications.

Magnetic tracking devices, which are used in many existing medical AR systems, may prevent the transition from an experimental state into the clinical practice. Such magnetic trackers can be disturbed by metallic objects like surgical instruments, and they can also possibly interfere with other sensitive equipment in the operation room.

Another difficulty of medical AR is the fact that a number of problems related to registration, tracking, and calibration need to be solved. These may require a lot of effort for the design, implementation, and test of respective algorithms and software components. The resulting algorithms can be computationally expensive and consume a considerable share of the available computing capacities.

The remaining paper is organized as follows: In Section 11.2, the general tasks for building a medical AR system are discussed. The VectorVision image guided surgery system is introduced in Section 11.3, and Section 11.4 contains a detailed description of our novel augmented reality system. Section 11.5 provides a discussion of the results of our work. Finally, we conclude by giving an outlook on future extensions in Section 11.6.

11.2 Components of a Medical AR System

11.2.1 Camera / Instrument Tracking

It is essential for any augmented reality application to measure in real-time the spatial position and orientation of the camera (video see-through) or of the user's head (optical see-through), respectively. There are various ways of doing this. The most common approaches are marker tracking, magnetic tracking, and specialized optical cameras.

Marker tracking is a video-based method which approximates camera position and orientation by analyzing artificial fiducials in the video stream [KB99]. The main drawback of this method is the fact that both the spatial range and the range of angles that can be measured are very limited. Whereas marker tracking can be implemented cheaply as a software-only solution based on the given video data, magnetic and external optical tracking systems consist of additional hardware. These devices can be very expensive and require complicated setup procedures. However their accuracy and trackable volumes usually surpass the capabilities of marker tracking.

In most medical AR systems, it is necessary to measure the position and orientation of interaction devices in addition to the user's camera. Such interaction devices are often similar to or based on surgical instruments. Instrument tracking can be performed using the same methods as described above.

11.2.2 Patient Registration

In order to correctly overlay virtual elements for the display of medical information, the patient needs to be registered with respect to a global coordinate system. The tracking methods described in Section 11.2.1 usually deliver positions and orientations in an abstract coordinate system, in relation to which the actual placement of the patient has to be determined. Patient registration is necessary so that previously acquired medical image data can be positioned and oriented correctly. This is one of the more challenging tasks of medical AR, which has been researched quite intensively [MV98, SRSB⁺02].

11.2.3 Creation of Preoperative Planning Data

Image data in medicine usually originates from 3D scanning devices like CT or MRI. This data is usually stored according to the DICOM standard [DIC03], which can be notoriously complex. One reason for this is the fact that modifications and derivations from the standard are often encountered.

For many applications it is useful to provide preoperative planning data in addition to the medical image data. Such information can con-

sist of trajectories and target points for an intervention, or three-dimensional objects resulting from a particular segmentation step. It is thus necessary to either import this information from third-party software or provide interaction capabilities for creating such data within the AR system. An example of preoperatively created planning data is symmetry information that is produced by our planning tool for maxillofacial surgery, *Schneewittchen* [FMdR⁺04].

11.3 The VectorVision IGS system

In the past few years, Image Guided Surgery (IGS) has become a very important domain of Computer Aided Surgery, and *VectorVision*[®] (see Figure 11.1a) is the current passive, optical-tracking IGS platform of *BrainLAB*.

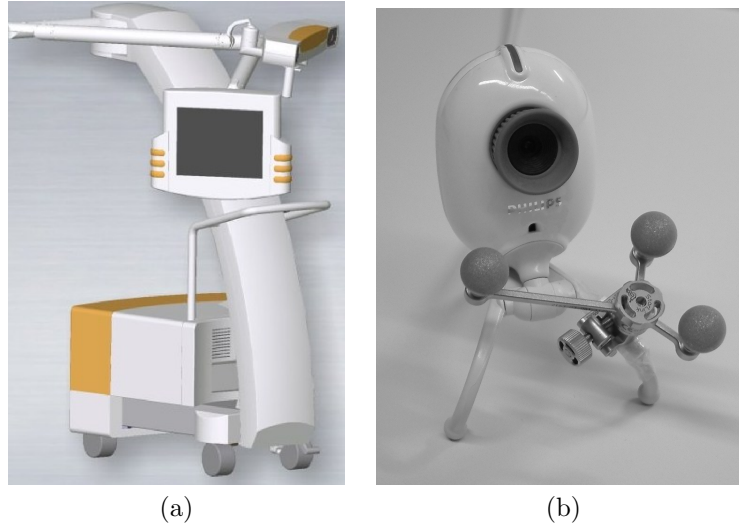


Figure 11.1: (a) VectorVision IGS system, (b) Webcam with attached passive infrared markers.

Among other things, IGS systems provide a 3D display of preoperative CT and MR scans of the patient and therefore can assist surgeons in multiple ways; first of all, certain disorders can be identified much easier by looking at the whole 3D volume from different directions instead of mentally reconstructing and orienting the volume from the 2D images. On the other hand, the surgeon can benefit from an IGS system during operation planning and during the actual procedure. With the help of the 3D model of the region of interest, the surgeon is supported in finding a way to access a tumor, while limiting the damage of the encompassing tissue or vital neighboring organs.

The IGS system consists of a PC, a touchscreen display and two infrared cameras. An infrared source integrated into each camera emits infrared waves. Special marker spheres that are rigidly attached to every tracked object reflect the infrared light back to the cameras. The

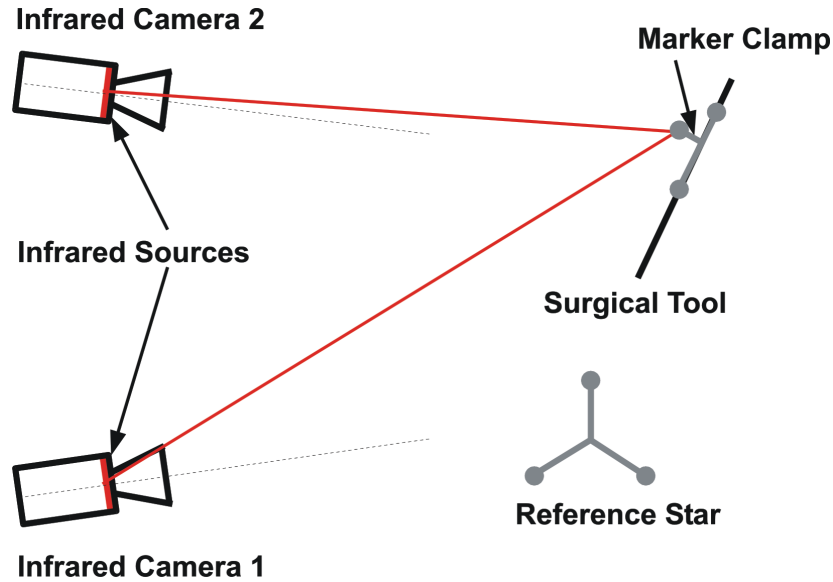


Figure 11.2: Infrared tracking of surgical instruments. The instrument is identified and tracked by the **infrared cameras** via the **marker clamp**, which has a specific configuration of three marker spheres. The **reference star** defines the reference coordinate system of the whole system and the surgical intervention.

position of each marker sphere is calculated via triangulation, which allows the touchscreen to display the 3D model of the calibrated surgical tool with a connected marker clamp at the correct location in the 3D dataset. The triangulation step is depicted in Figure 11.2.

Disregarding the fact that soft tissue can slightly modify shape and position over time, in particular between the image acquisition and the surgical intervention, the technically possible accuracy achievable today is in the order of magnitude of 1 mm.

11.3.1 VectorVision Link

For an AR system like the one described in this paper, it is important to have access to state-of-the-art medical imaging equipment and to be able to access intermediate data of an IGS system. Sometimes it might also be necessary to integrate early prototypes of new subsystems or algorithms for the purpose of clinical validation or verification.

To enable the interaction of research (sub)systems with an existing IGS product, it is desirable for the developers to be provided with specific, well-suited interfaces for this task; *VectorVision Link (VVL)* is an easy to use TCP/IP-based client-/server interface integrated into the BrainLAB *VectorVision® Cranial* IGS system. The API is based on the Visualization Toolkit (VTK) [SML98] and can be accessed through C++, but also through Java and the two scripting languages Tcl/Tk and Python. VTK was chosen because it is well established in research community, thus making interaction with VectorVision® easy.

Among other information, the 3D medical image sets of the patient and the positions of the currently tracked tools can be interactively downloaded for processing by an external system. In return, new image sets and informative 2D bitmaps can be uploaded to the IGS system and used by the surgeon during the intervention. More in-depth information about VVL can be found in [ND03].

11.4 The New Augmented Reality System

In this paper, we describe a novel system for medical augmented reality. The system is based on a VectorVision image guided surgery device produced by the BrainLAB company [Bra02]. We show that all of the tasks described in Section 11.2 can be performed by this IGS system.

Our medical AR application is designed as a video see-through system. In order to facilitate the implementation of a prototype, we have not used special display devices so far. Currently the augmented video stream generated by our software is displayed on a conventional computer monitor. We also plan to examine possibilities for building a stereo augmented reality system based on a new, experimental autostereoscopic LCD display.

11.4.1 Camera Tracking

An off-the-shelf webcam is used for the acquisition of the video images. One central aspect of the design of the new AR system is to devise a method for tracking the webcam using the given equipment. The VectorVision system is capable of tracking surgical instruments using its built-in infrared camera system. We harness this capability by attaching an instrument marker clamp to the webcam. The setup is shown in Figure 11.1b.

The position and orientation of the instrument clamp measured by the IGS system is received using the VectorVision Link interface. Although the latency depends on the system load of the VectorVision device and the current network traffic, real-time transfer of the data can be achieved most of the time.

Tracked Instrument Pose Data

The major difficulty of using tracked instrument data for tracking the webcam is the fact that the pose information delivered by the IGS system does not directly correspond to that of the camera. In order to utilize a marker clamp, it has to be calibrated, as if it was attached to an actual surgical instrument. This is done using a special procedure aimed at ensuring accuracy and correct placement of medical tools within the patient's coordinate system. Figure 11.3 shows how a mock-up of a surgical tool is calibrated with respect to a so-called reference star

(seen at the left). The same marker clamp used for the webcam in Figure 11.1b is attached to the instrument mock-up.

Since the calibration procedure requires the instrument to have a fixed physical tip, it is not possible to directly calibrate the webcam at the reference star. Thus the AR system can only obtain position and orientation information in relation to the instrument mock-up, as shown in Figure 11.4. In this figure, **pos** denotes the position of the instrument's tip. The direction of the instrument's trajectory is given as vector **dir**, with **norm** being a plane normal perpendicular to it. By combining the information contained in **pos**, **dir**, and **norm**, the complete six degrees of freedom pose information can be obtained. It is then necessary to transform this pose, which is relative to the instrument used for the initial reference star calibration, into the camera pose required for the AR image composition. We have devised an additional one-time calibration step for computing this transformation.

One-time Calibration

A matrix describing the transformation from the tracked instrument to the camera is computed. In order to be able to perform this computation, a common reference coordinate system is established at a given point of time. This reference coordinate system contains both the tracked instrument and the webcam in absolute positions and orientations. A transformation between the two can then be easily derived. For this one-time calibration, conventional marker tracking based on the ARToolKit by Kato et al. is employed [KB99].

An ARToolKit marker is placed in or near the trackable volume of the VectorVision infrared camera. To make the measurements and the

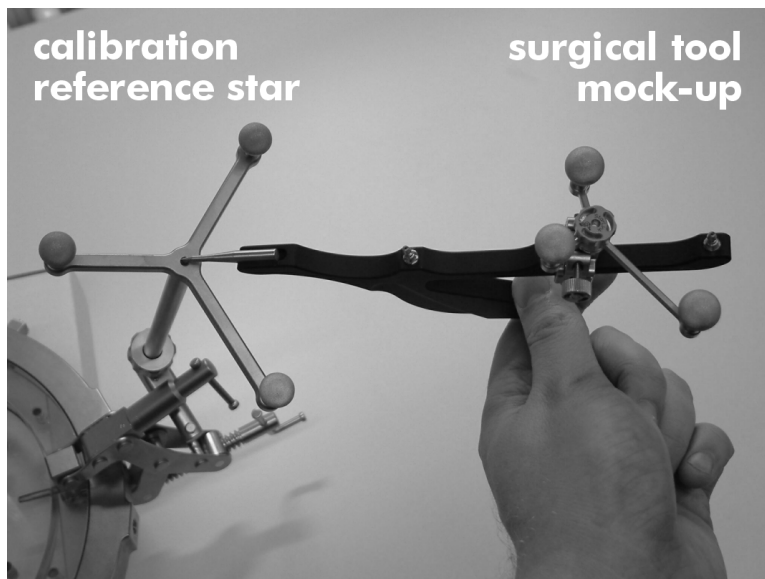


Figure 11.3: Calibration of a surgical tool in the IGS system.

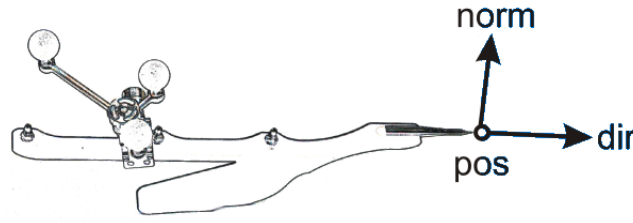


Figure 11.4: 6-DOF instrument data delivered by image guided surgery.

calibration stable and repeatable, a physically fixed relation between the reference star and the marker is ensured. The user then defines the positions of the marker corners in the coordinate system of the infrared camera; a standard pre-calibrated pointer tool is used to indicate the spatial positions of the corners. The pointer tool is automatically recognized by the image guided surgery system, and the position of its tip can be retrieved using the VectorVision Link. Figure 11.5 shows the definition of two of the marker's corners.

The user is prompted to define the corners in clockwise order as shown in Figure 11.6. Thus the transformation matrix from the VectorVision coordinate system to the coordinate system of the marker can be easily constructed. The marker coordinate system is the same one as used in the ARToolKit; vectors \mathbf{u} and \mathbf{v} in Figure 11.6 correspond to the x and y axes, respectively. The z axis points upward towards the viewer, creating a right-handed coordinate system.

The transformation matrix from the VectorVision to the marker coordinate system, $A_{vvctomarker}$, is computed according to Equation 11.1. The origin of the marker coordinate system is the geometric centroid of the four defined corner points. Note that both \mathbf{u} and \mathbf{v} are normalized. Other considerations concerning the scaling of the coordinate systems are not necessary, since both VectorVision and the ARToolKit define a coordinate unit as equivalent to 1 millimeter.

$$\begin{aligned}\mathbf{u} &:= \frac{P_4 - P_1}{\|P_4 - P_1\|} & \mathbf{v} &:= \frac{P_2 - P_1}{\|P_2 - P_1\|} \\ \mathbf{c} &:= \frac{P_1 + P_2 + P_3 + P_4}{4}\end{aligned}\tag{11.1}$$

$$A_{vvctomarker} := \left(\begin{array}{c|c|c|c} \mathbf{u} & \mathbf{v} & \mathbf{u} \times \mathbf{v} & \mathbf{c} \end{array} \right)^{-1}$$

In the next step, the actual one-time calibration procedure is triggered by the user. At one point in time, the webcam has to be positioned so that the optical marker can be recognized by the ARToolKit,

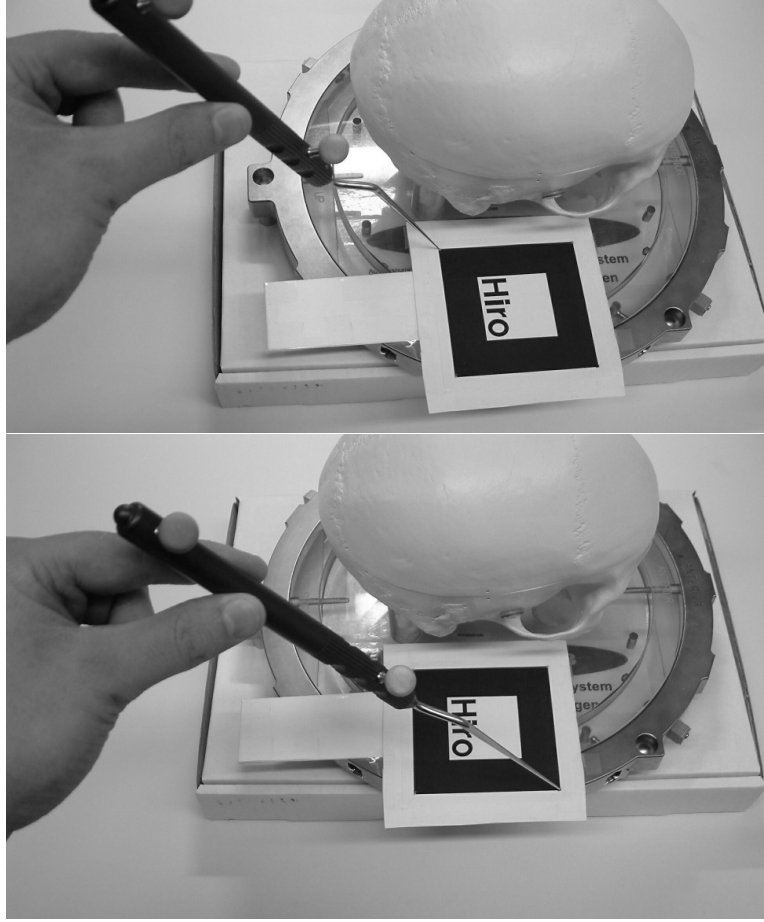


Figure 11.5: Two of the marker's corners are indicated by the user with the pointer tool.

while the infrared markers are seen by the VectorVision cameras simultaneously. This is illustrated in Figure 11.7. The computation of the necessary transformation matrices is then triggered with a single keystroke.

We now calculate the matrix for the transformation from the VectorVision coordinate system to the coordinate system of the tracked instrument marker clamp. When the calibration is performed, the current tracked instrument data, as shown in Figure 11.4, is recorded. This pose information directly leads to the construction of matrix $A_{vvtoinsm}$ (see Equation 11.2), which describes a right-handed coordinate system with the (imaginary) instrument tip at its origin.

$$A_{vvtoinsm} := \left(\begin{array}{c|c|c|c} \mathbf{dir} & \mathbf{norm} & \mathbf{dir} \times \mathbf{norm} & \mathbf{pos} \end{array} \right)^{-1} \quad (11.2)$$

Finally the marker transformation matrix is retrieved from the AR-ToolKit. It is then inverted, yielding the transformation $A_{camtomarker}$

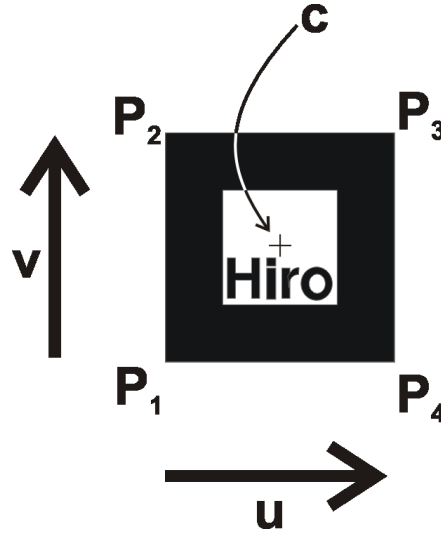


Figure 11.6: Marker corners are defined in a clockwise order from P_1 to P_4 . The marker shown here is taken from the ARToolKit [KB99].

from the coordinate system of the webcam to the marker coordinate system. Using the ARToolKit marker as the common point of reference, it is now possible to compute the transformation from the marker clamp to the webcam coordinate system. Figure 11.8 illustrates the relations between the transformation matrices involved in the calibration procedure.

$$A_{insmtocam} = A_{camtomarker}^{-1} \cdot A_{vvctomarker} \cdot A_{vvctoinsm}^{-1} \quad (11.3)$$

Equation 11.3 shows how the result of the calibration, matrix $A_{insmtocam}$, is computed. Note that this transformation only needs to be computed once. As long as the physical relation between the instrument marker clamp and the webcam remains unaltered, the matrix remains valid. A new calibration is only required when the marker clamp is moved relative to the webcam.

Operation of the Camera Tracking System

After the one-time calibration step, only the data from the VectorVision infrared camera is required for tracking the webcam. The vision-based marker tracking is not used for the operation of the augmented reality application itself. For every frame that is generated by the AR system, the current tracked tool data is retrieved using the VectorVision Link. An $A_{vvctoinsm}$ matrix is computed as described in Equation 11.2. The final transformation matrix used for OpenGL camera setup is the result of a simple multiplication as shown in Equation 11.4.

$$A_{vvctocam} = A_{insmtocam} \cdot A_{vvctoinsm} \quad (11.4)$$



Figure 11.7: The ARToolKit marker is used as the common point of reference.

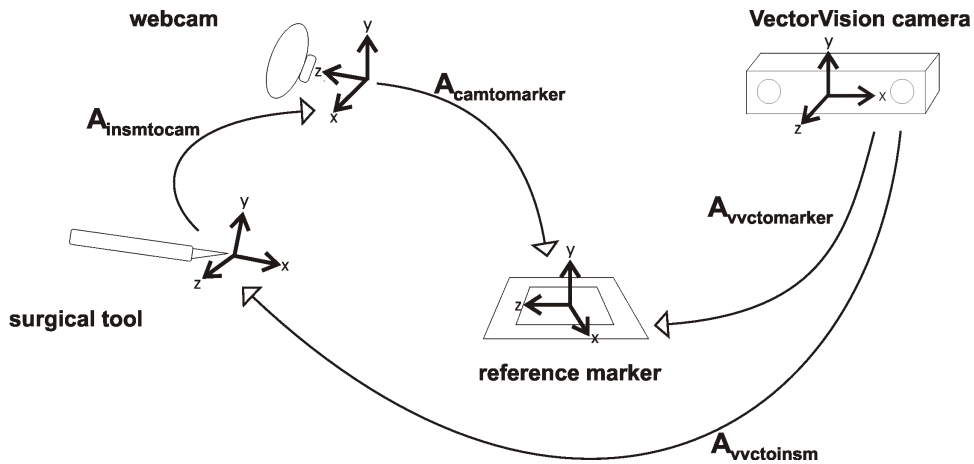


Figure 11.8: Overview of the transformation matrices involved in the calibration and tracking process.

11.4.2 Additional Support by the IGS system

In addition to the camera tracking described above, the image guided surgery system also provides support for the remaining tasks of medical AR. Surgical tools are tracked in the same way as the webcam's marker clamp (see Section 11.4.1). Several instruments can be tracked simultaneously, each delivering 5-DOF or 6-DOF pose information.

The VectorVision system offers several methods for patient registration. These include algorithms based on the matching of anatomical landmarks, artificial fiducials attached to the patient, or threedimensional surface point clouds generated by a special handheld laser device.

Also, a software pipeline for importing and preprocessing DICOM image datasets is provided by the IGS system. It is possible to define target points or trajectories for interventions. Moreover volumes origi-

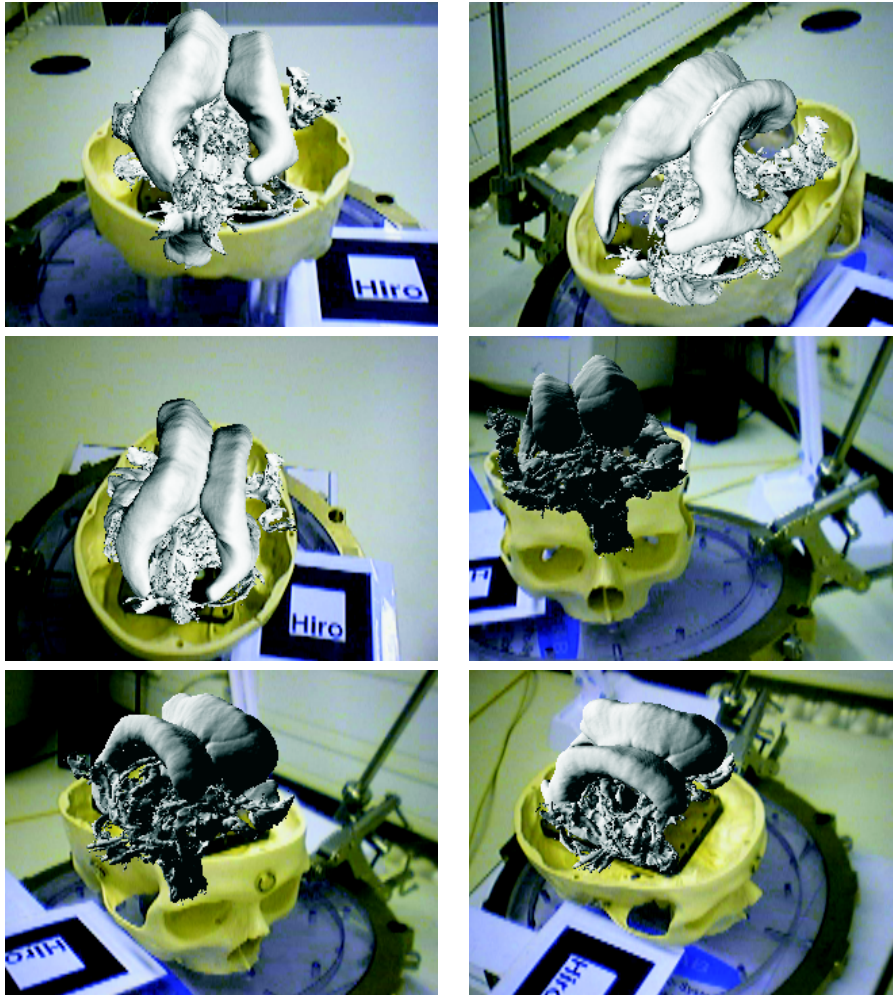


Figure 11.9: Several views generated by the AR system using the ventricle dataset.

nating from different scanning modalities can be registered for use in a single session.

All of this data can be exchanged with the augmented reality system using the VectorVision Link. The AR application can upload and download volume datasets, retrieve patient registration parameters and tracked instrument data, and read and add intervention planning information.

11.5 Results and Discussion

We have implemented a simple application demonstrating the feasibility of IGS-based medical AR. Several images generated by the software during an AR session are shown in Figure 11.9. The surface model of a ventricle generated from an MRI scan is positioned inside the plastic phantom of a human skull. The graphical model was originally used in the virtual endoscopy system VIVENDI [BSW⁺99].

Our test sessions showed that the medical AR system is capable of generating an augmented video stream at an interactive frame rate. Average frame rates of more than 10 fps were achieved over all tests, for which a webcam resolution of 320x240 pixels was selected. We have measured an average latency of approximately 80 ms for receiving the tracked instrument data from the VectorVision system, depending on the number of tracked tools and the current configuration of the VectorVision software.

The camera tracking provided by the image guided surgery system delivers a high position accuracy. Sporadic visual mismatches between virtual and real scene elements result from the time lag between infrared marker registration, webcam image acquisition and generation of the augmented video.

The major drawback of our tracking method is the fact that the tracking range is limited to the viewing volume of the infrared camera pair. Moreover, the system is sensitive to occlusion of the marker clamps from the viewpoint of the infrared cameras. This makes our augmented reality application ill-suited for the use of head-mounted displays. Since the trackable volume is normally centered at the operation situs, the user's head is outside of it most of the time. Depending on the setup of the infrared cameras and the size of the webcam, the marker clamp can become occluded by the webcam itself. This makes a change of the marker clamp configuration and a subsequent re-calibration necessary.

11.6 Conclusion and Future Work

We have presented a novel approach of realizing augmented reality for medical applications. Unlike other research systems, our solution is based on existing equipment for medical visualization and navigation. Devices for image guided surgery are becoming increasingly widespread and can be found in many operation rooms. The transition of augmented reality into the clinical practice thus can be facilitated by the fact that IGS systems are ubiquitous, well tested, stable, certified for medical settings, and easy to use.

Since the VectorVision image guided surgery devices have been designed specifically for medicine, many practical problems like possible magnetic interference or working in a sterile environment have already been taken care of. An IGS-based augmented reality system benefits from these qualities. Utilizing image guided surgery for medical AR also significantly reduces the costs and shortens the required setup procedures. Assuming that IGS equipment exists for a certain medical application, additionally only a standard computer system and webcam are necessary for building a basic AR system. The initialization of the image guided surgery system can be performed within a few minutes by a trained surgeon, while the one-time camera calibration step

(Section 11.4.1) is only required in case of a configuration change.

We plan to extend our AR application into a more complete system for displaying the augmented video stream in a non-HMD configuration on a conventional or autostereoscopic display. The VectorVision Link also provides the possibility to upload and show externally generated images on the IGS display, which could give the surgeon access to AR functionality within the familiar navigation environment. We also intend to eliminate visual mismatches due to the time lag in the rendering loop with predictive tracking methods. An improved tracking accuracy might also be achieved by using pose information of the marker clamp centers instead of virtual tool tips. Finally, we will examine possible applications of IGS-based augmented reality in neurosurgery or maxillofacial surgery.

Acknowledgements

We would like to thank Ángel del Río for helpful discussions concerning the camera calibration step. This work has been supported by project VIRTUE in the focus program on "Medical Robotics and Navigation" of the German Research Foundation (DFG).

Bibliography

- [Azu97] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [BBR⁺03] A. Bornik, R. Beichel, B. Reitinger, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Computer Aided Liver Surgery Planning Based on Augmented Reality Techniques. In *Proc. of Workshop Bildverarbeitung in der Medizin*, Informatik Aktuell, 2003.
- [Bra02] BrainLAB. Neurosurgery Product Brochure. BrainLAB AG, 2002.
- [BSW⁺99] D. Bartz, M. Skalej, D. Welte, W. Straßer, and F. Duffner. A Virtual Endoscopy System for the Planning of Endoscopic Interventions in the Ventricle System of the Human Brain. In *Proc. of BIOS'99: Biomedical Diagnostics, Guidance and Surgical Assist Systems*, volume 3514, pages 91–100, 1999.
- [DIC03] DICOM Standards Committee. *Digital Imaging and Communications in Medicine (DICOM)*. National Electrical Manufacturers Association, 2003.
- [FMdR⁺04] J. Fischer, M. Mekic, A. del Río, D. Bartz, and J. Hoffmann. Prototyping a Planning System for Orbital Reconstruction. In *Proc. of Workshop Bildverarbeitung in der Medizin*, Informatik Aktuell, pages 264–268, 2004.
- [Goe03] G. Goebbels. ARSyS-Tricorder - Entwicklung eines Augmented Reality Systems für die intraoperative Navigation in der MKG Chirurgie. In *Jahrestagung der Deutschen Gesellschaft für Computer-und Robotergestützte Chirurgie e.V. (CURAC)*, 2003.
- [KB99] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *Proc. of IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, 1999.

- [MV98] J. Maintz and M. Viergever. A Survey of Medical Image Registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [ND03] M. Neff and K. Diepold. Design and Implementation of an Interface Facilitating Data Exchange between an IGS System and External Image Processing Software. Diplomarbeit, TU München, 2003.
- [SFW02] R. Splechtna, A. Fuhrmann, and R. Wegenkittl. ARAS - Augmented Reality Aided Surgery System Description. Technical report, VRVis Research Center for Virtual Reality and Visualization, 2002.
- [SLH⁺96] A. State, M. Livingston, G. Hirota, W. Garrett, M. Whittton, H. Fuchs, and E. Pisano. Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. In *Proc. of ACM SIGGRAPH*, pages 439–446, 1996.
- [SML98] W. Schroeder, K. Martin, and W. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 2nd edition, 1998.
- [SRSB⁺02] M. Scheuering, C. Rezk-Salama, H. Barfuss, A. Schneider, and G. Greiner. Augmented Reality Based On Fast Deformable 2D-3D Registration For Image Guided Surgery. In *Proc. of SPIE Medical Imaging*, pages 436–445, 2002.
- [SS04] B. Schwald and H. Seibert. Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality. In *Journal of WSCG*, pages 411–418, 2004.
- [SSW02] B. Schwald, H. Seibert, and T. Weller. A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 261–262, 2002.
- [SVH⁺02] T. Suthau, M. Vetter, P. Hassenpflug, H.-P. Meinzer, and O. Hellwich. A Concept Work for Augmented Reality Visualisation Based on a Medical Application in Liver Surgery. In *Proc. of IAPRS*, 2002.

Chapter 12

Occlusion Handling for Medical Augmented Reality Using a Volumetric Phantom Model

Jan Fischer¹, Dirk Bartz¹, and Wolfgang Straßer²

¹Visual Computing for Medicine Group, University of Tübingen

²WSI/GRIS, University of Tübingen

Extended version of the paper, which appeared in the Proceedings of the ACM Symposium on Virtual Reality Software and Technology (ACM VRST) 2004.

The support of surgical interventions has long been in the focus of application-oriented augmented reality research. Modern methods of surgery, like minimally-invasive procedures, can benefit from the additional information visualization provided by augmented reality. The usability of medical augmented reality depends on a rendering scheme for virtual objects designed to generate easily and quickly understandable augmented views. One important factor for providing such an accessible reality augmentation is the correct handling of occlusion of virtual objects by real scene elements. The usually large volumetric datasets used in medicine are ill-suited for use as phantom models for static occlusion handling. We present a simple and fast preprocessing pipeline for medical volume datasets, which extracts their visual hull volume. The resulting, significantly simplified visual hull iso-surface is used for real-time static occlusion handling in our AR system, which is based on off-the-shelf medical equipment.

CR Categories: H.5.1: Information Interfaces and Presentation: Artificial, augmented, and virtual realities: I.3.7:

**Three-Dimensional Graphics and Realism: Virtual Reality;
I.3.8: Application: Virtual Medicine; J.3: Life and Medical
Sciences: Medical information systems;****12.1 Introduction**

In augmented reality (AR), virtual graphical elements are added to the user's real surroundings. The essential precondition for the realistic overlay of virtual objects is their correct three-dimensional positioning and orientation. There are different approaches to designing an AR system, which are usually divided into optical see-through and video see-through systems. While the former utilize special translucent display devices, the latter achieve reality augmentation by processing a digital video stream [Azu97].

The planning, preparation and support of medical interventions has always been among the most important applications of augmented reality. In many modern surgical procedures, the surgeon's view of the patient anatomy and risk structures can be very limited. This is especially true for minimally-invasive interventions, which often are facilitated by a supplemental graphical display of anatomical structures and operation plans. Dedicated medical equipment for navigation and visual representations during surgical interventions is available as so-called image guided surgery (IGS) devices. Such devices are offered as commercial models by several manufacturers, and have become widespread in certain medical applications [Bra02].

In addition to established standard systems for medical visualization, a number of experimental augmented reality prototypes for the support of medical treatment have been designed in recent years. Although many of these systems have reached a high level of quality in terms of tracking and registration accuracy, image generation latency and graphical quality, they usually rely on specific hardware or sophisticated algorithms. Such base technologies often are not well suited for use in an actual surgical intervention, and may prevent the transition of these experimental systems into clinical practice.

It is essential for any AR application to provide an augmented display that is easy to interpret by the user. This tends to be even more important in any application scenario, where users often are less technically experienced than in the field of augmented reality research. A significant visual cue for understanding the spatial relationships in a scene is the correct occlusion of objects from the user's viewpoint. Whereas occlusion between real elements (optical occlusion) and between virtual elements (z-buffer tests) in an AR scene is automatically correct, it is difficult to determine when a real object should be in front of a graphical object [Kli01]. If the geometry and placement of static objects in the real surroundings are known, the use of so-called phantom models is a common approach for handling the occlusion of virtual

objects [BWRT96].

12.2 Related Work

A number of experimental systems and prototypes for medical augmented reality have been developed. State et al. have presented an augmented reality application for ultrasound-guided needle biopsies [SLH⁺96]. The system offers accurate tracking of both the user and the ultrasound probe as well as stereoscopic rendering, but relies on a number of proprietary technologies. These include a hybrid optical-magnetic tracker for the user's head and a mechanical arm for probe registration.

In recent years, AR systems were created for the support of various medical application scenarios. The AR system prototype of Splechna et al. [SFW02] consists of an optical tracking system comprising two cameras and a head-mounted display. It is capable of augmenting the surgeon's view with preoperative volume data or live ultrasound images. A special translucent display device mounted on a swivel arm is used in the Medarpa project [SSW02]. In this application, tracking is performed with a hybrid system using active infrared and electromagnetic components. A semi-transparent mirror display mounted directly over the patient is used for reality augmentation in the ARSys-Tricorder described by Goebbels [Goe03]. It also provides stereoscopic rendering, with infrared cameras being used for tracking the surgeon's head and components of the AR system.

Several research groups have worked on the detection and handling of occlusion in an augmented reality rendering pipeline. As mentioned above, Breen et al. have suggested a model-based method for handling static occlusion in AR [BWRT96]. An extension of this method for determining how virtual objects are hidden by the user's body was described by Fuhrmann et al. [FHFG99]. Their system combines a previously acquired geometric model with positional data from a tracking system. The AR system of Malik et al. [MMR02] detects the occlusion of black and white marker regions by the user's hand. This is achieved using a histogram-based thresholding process refined by a flood-fill algorithm. A method for detecting dynamic occlusion in front of static backgrounds is described in [FRB03]. This algorithm does not require any previous knowledge about the occluding objects, but relies on a textured graphical model of planar elements in the scene.

Approaches for detecting occlusion in a stereo camera AR system have also been investigated. In some cases, an attempt to solve the occlusion problem using depth information delivered by stereo matching is made [KOTY00, WA95]. The method developed by Gordon et al. [GBB⁺02] can correctly render interaction devices into the scene. Some research has also been done into occlusion in non-real-time augmented reality. Correctly handling occlusion while adding virtual objects to stored video sequences was examined by Berger and Lepetit [Ber97,

LB00].

12.3 Augmented Reality based on Image Guided Surgery

A major disadvantage of many existing medical AR systems is their reliance on specialized system components for display or tracking. Many of these devices are not well suited for use in a medical application. Head mounted displays and stereo shutter glasses are often considered to be too bulky, and they can deteriorate the user's perception of the surroundings. Moreover, connecting cables used for data transmission or power supply can limit the user's range of motion. Electromagnetic tracking systems can be disturbed by metallic objects like surgical instruments, and they can interfere with medical equipment in the operating room. Dedicated optical tracking cameras, like other components of experimental augmented reality systems, often require tedious setup and calibration procedures. The very limited space available in the operating room also prevents the utilization of some types of devices. Furthermore, practical problems like working in a sterile environment and certification for medical settings usually have not been solved for specialized VR and AR tracking and display equipment.

We have proposed an alternative approach to the design of an augmented reality application for medicine [FNBF04]. Our AR system is based on commercially available medical equipment. A VectorVision image guided surgery system manufactured by the BrainLAB company is used for performing many of the tasks required for augmented reality (see Fig. 12.1a). In the remainder of this section, we give a short overview of our system.

12.3.1 Camera Tracking

Our AR system *ARGUS* (Augmented Reality using image Guided Surgery) is a video see-through system for medical augmented reality. It uses hardware and software components of the VectorVision device for supporting the generation of an augmented video stream. An essential task is the real-time measurement of the spatial position and orientation of the camera acquiring the system's input video stream. The IGS device is equipped with an infrared camera pair for the fast and highly accurate tracking of surgical instruments with passive infrared marker spheres. In order to utilize these capabilities for our application, one of the infrared marker clamps is attached to the webcam used in the AR system (see Figure 12.1b).

The 6-DOF tracked instrument pose information is retrieved through a specific network interface for accessing internal data of the IGS device, called VectorVision Link [ND03]. The pose delivered by VectorVision Link corresponds to that of a hypothetical surgical instrument. It is

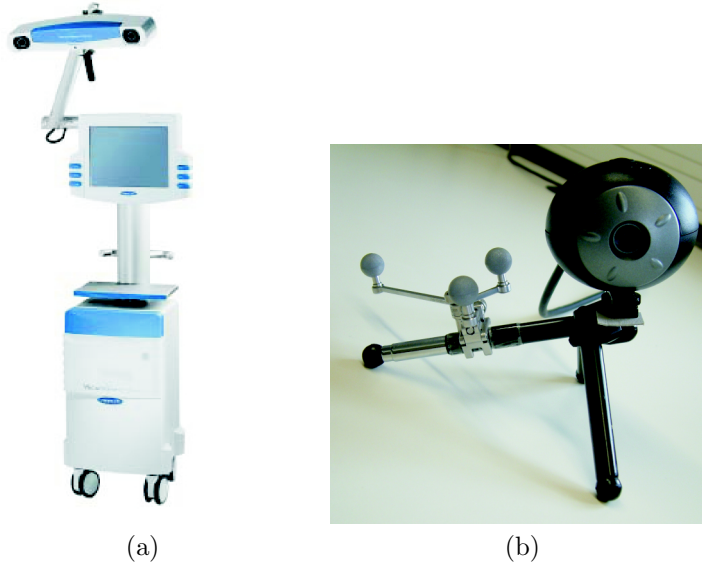


Figure 12.1: (a) VectorVision image guided surgery system (BrainLAB), (b) webcam with attached infrared markers.

thus necessary to compute a transformation from the surgical instrument coordinate system to the webcam coordinate system. We have devised an one-time calibration step for computing the corresponding transformation matrix.

In order to establish a common coordinate system for both the infrared cameras and the webcam, conventional optical marker tracking based on the ARToolKit by Kato et al. is used [KB99]. The *ARGUS* software provides functionality for measuring the spatial positions of the corners of an ARToolKit marker. This is done using a standard pre-calibrated VectorVision pointer tool, yielding 3-d positions in the coordinate system of the infrared cameras. Figure 12.2 illustrates the definition of the marker corners by the user.

The coordinates obtained by the process shown in Figure 12.2 are used for constructing a transformation matrix from the VectorVision camera coordinate system to a coordinate system located at the marker's center. The calculation of this matrix is described in detail in [FNBF04].

For performing the actual calibration, at one point of time the instrument marker clamp must be registered by the infrared cameras, while the webcam can detect the ARToolKit marker simultaneously. The user then triggers the calibration computation, causing the *ARGUS* system to store two more matrices at that specific moment. One matrix is directly retrieved from the VectorVision Link interface and describes the transformation from the coordinate system of the infrared cameras to the hypothetical surgical instrument. Moreover, the transformation matrix from the webcam to the marker coordinate system is obtained from the marker tracking component of ARToolKit. Using these three matrices, the transformation from the instrument clamp to

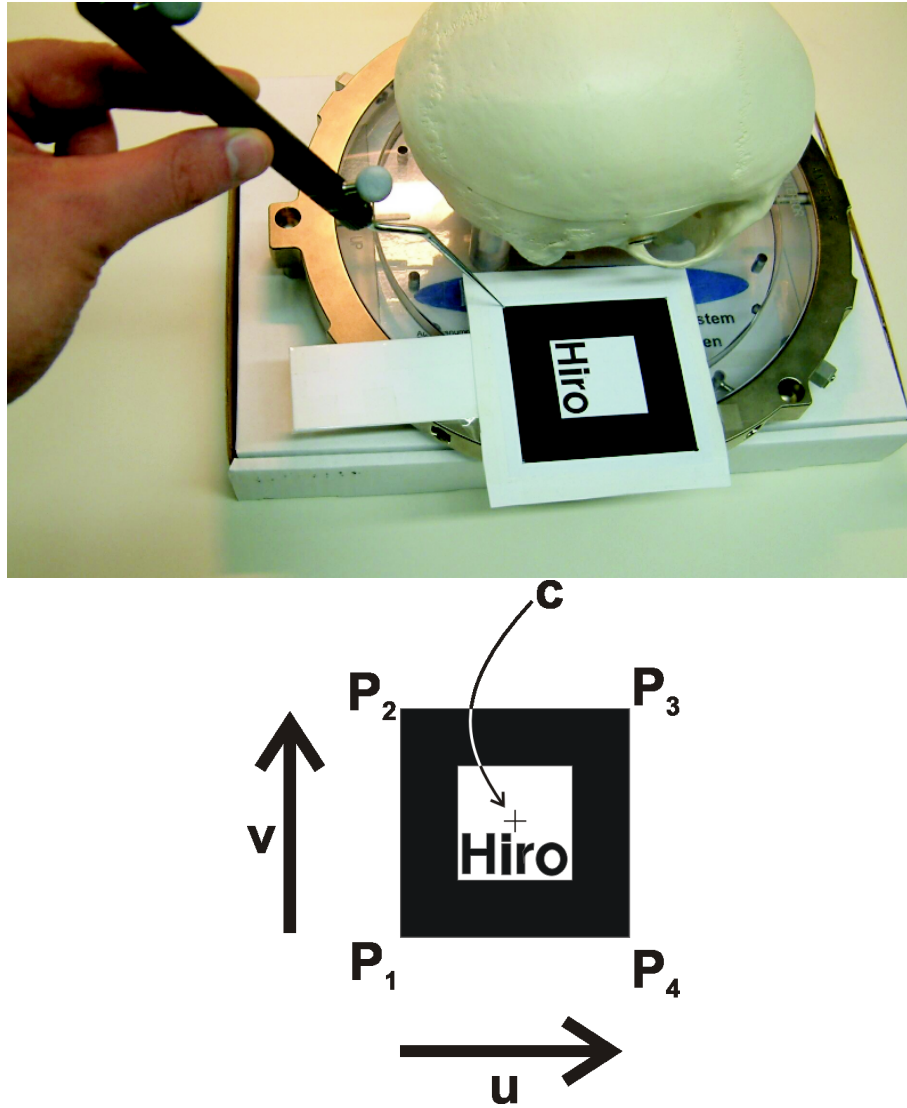


Figure 12.2: Marker corners are defined in a clockwise order from P_1 to P_4 .

the webcam coordinate system can be computed easily.

During the operation of the AR system, the current surgical instrument transformation matrix is downloaded using the VectorVision Link for every video frame to be generated. The global transformation to be used for rendering the virtual objects can then be obtained by simply taking the result of the one-time calibration into account. This means that the application programmer can conveniently specify virtual graphical objects within the same coordinate system in which the patient has been registered. During the actual operation of the *ARGUS* system, only the infrared cameras are used for tracking the webcam. As long as the physical relationship between the webcam and the instrument marker clamp is not changed, a re-calibration is not necessary.

12.3.2 Additional Support by the IGS System

In addition to the tracking of the webcam, the image guided surgery device also provides support for a number of other tasks. Several methods for registering the patient in the global coordinate system of the infrared cameras are available. These include the matching of manually selected anatomical landmarks or artificial fiducials stuck to the patient's skin with respective features in the volumetric medical dataset. In a more advanced mode of registration, a three-dimensional point cloud generated using a special handheld laser device is registered by the infrared cameras and matched to the dataset using an automatic registration algorithm. The IGS system has a built-in software pipeline for the import and preprocessing of medical datasets in the standard DICOM format [DIC03]. Moreover, operation plans comprising target points, planned trajectories and separately segmented 3-d objects can be created and displayed with the VectorVision system. All of this data can be exchanged with the AR application, facilitating the generation of a realistic and useful reality augmentation for medicine.

An augmented image generated by the *ARGUS* system is shown in Figure 12.3. In the image, a manually placed virtual tumor model is displayed with correct spatial positioning and orientation relative to a plastic skull mock-up.

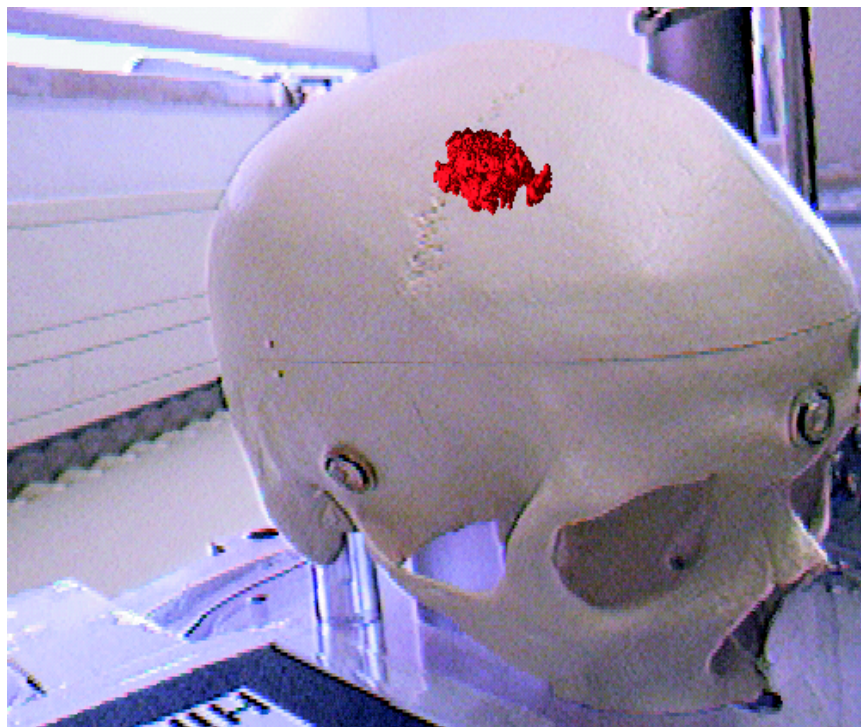


Figure 12.3: Image generated by our augmented reality system. The virtual tumor model is located at the center of the upper half of the image.

12.4 Static Occlusion Handling

In this paper, we present an approach for correctly handling the occlusion of virtual graphical objects by the patient. By solving this problem, we enable the user to easily determine whether a graphical object - like the rendering of a surgical tool or an instrument's trajectory - is supposed to be in front of or behind the patient. A volumetric dataset containing the relevant part of the patient's anatomy is acquired before the actual intervention using a medical scanning procedure like computed tomography (CT) or magnetic resonance imaging (MRI). After the initial registration, the IGS system requires that the patient's position and orientation remain stationary with respect to the infrared camera's coordinate system. Our method thus is a special case of static occlusion handling as described in [BWRT96].

The basic idea of static occluding handling is to define a polygonal representation of real objects in the user's surroundings. These so-called phantom models are then utilized in the AR rendering pipeline to achieve the effect of virtual objects appearing occluded. In most applications, relatively simple phantom models are used. Unlike such simple polygonal descriptions, models derived from medical volumetric datasets can consist of several million triangles. This problem is aggravated by the steadily increasing resolution provided by modern medical scanners. In order to prevent occlusion handling from having an exceedingly negative impact on the overall frame rate, we have thus devised a method for reducing the number of triangles.

12.4.1 Volume Preprocessing

We describe a volume preprocessing pipeline for reducing the polygonal complexity of models generated from medical datasets. Unlike general simplification schemes, our method tries to preserve a highly detailed model of the anatomy's outer surface, while completely eliminating inner structures invisible from the outside. This is achieved by extracting the visual hull volume, a binary volume in which all parts of the volume dataset than cannot be seen from the outside, including anatomical cavities, have full intensity. In order to compute this visual hull volume, we perform a number of first-hit raycasting tests from six orthogonal directions. This can be considered a simple approximative simulation of possible outside views of the volume dataset. Figure 12.4 illustrates the concept of the visual hull volume.

First-Hit Raycasting

The elementary operation of our algorithm is the casting of a single ray parallel to the respective coordinate axes into the volume dataset. The ray is iteratively traversed until a voxel intensity above a user-defined threshold is encountered. Until that threshold condition is met or the



Figure 12.4: Comparison of an image slice in the input and visual hull volumes for a skull dataset. (a) Original volume, (b) visual hull volume.

ray leaves the volume boundaries, zero values are written into a second volume dataset at the same coordinates. This resulting volume, the visual hull volume, is initialized with full voxel intensities (255 in case of 8-bit volumes) beforehand. This raycasting process is an approximative simulation of the user viewing the outer surface of the volume dataset from one direction.

Parallel rays are generated over the entire area of each of the volume boundary's faces. The visual hull volume is initialized with full intensities only at the very beginning of the process. Each of the raycasting iterations then works on the already modified volume. Thus the empty parts surrounding the actual anatomy are "carved out" of the volume consecutively. The principle of the consecutive first-hit raycasting iterations is shown in the following piece of pseudocode:

```
originalVolume: VolumeData;
visualHull:      VolumeData;

rayCastingDirs: Vector3d[6] :=
  {{ 0, 0, 1}, { 0, 0,-1}, { 0, 1, 0},
   { 0,-1, 0}, { 1, 0, 0}, {-1, 0, 0}
  };

procedure preprocessVolume(threshold: double)
  startPos, pos: Vector3d;

begin
  initializeWithFullIntensity(visualHull);

  for directionLoop := 1 to 6 do
    for startPos ∈ all positions on corresponding
      face of volume boundary do
      pos := startPos;
      while (pos within volume boundaries) do
        if (originalVolume[pos] > threshold)
```

```
        break;

        visualHull[pos] := 0;
        pos := pos + rayCastingDirs[directionLoop];

    done
done
done
end
```

While the first-hit raycasting method works well for most volume datasets, ray iterations can be terminated erratically in cases with an above-average level of noise. We have thus embedded the raycasting process into a volume processing pipeline comprising additional filtering steps.

Volume Processing Pipeline

Before the actual raycasting operation is applied to the volume dataset, two filtering steps are performed. The first is a standard low-pass Gaussian volume filter. It reduces noise in the volume, but also tends to soften the edges of anatomical structures. Like for most steps in our volume processing pipeline, the user can opt against using the Gaussian filter and directly use the original volume as input for the next stage. If Gaussian smoothing is applied to the dataset, the standard deviation used by the filter can be modified by the user. After this low-pass filtering step, morphological operations are applied to the volume. These operations can remove isolated islands of low or high intensity (e.g. small holes) from the volume. In order to achieve this improvement, at first a 3-d morphological opening and then a closing operation are performed. A more detailed description of morphological operators can be found in [RW92]. Here again, the user can skip this preprocessing step and directly continue with the next stage. The kernel size used for the morphological opening and closing steps can be adjusted manually, if required. Figure 12.5 shows the effect of the volume preprocessing stage on a slice of a volume with a relatively high level of noise.

The computation of the visual hull volume from the preprocessed volume dataset, as described earlier in this section, only yields voxels with either full intensity or zero intensity. Since such a quasi-binary volume dataset leads to visible artifacts in the subsequent polygonal iso-surface extraction, another volume filtering step is applied beforehand. In order to remove possible remaining noise and smooth the edges, first a median filter [RW92] and then another Gaussian smoothing are used. Unlike for previous steps of the pipeline, we have found fixed kernel sizes to work well for the postprocessing of visual hull volumes. The median filter uses a kernel size of 3x3x3 voxels, while the Gaussian postprocessing filter has a standard deviation of 1.5 voxels. Since the median filter step is not necessary for all volumes, it can be

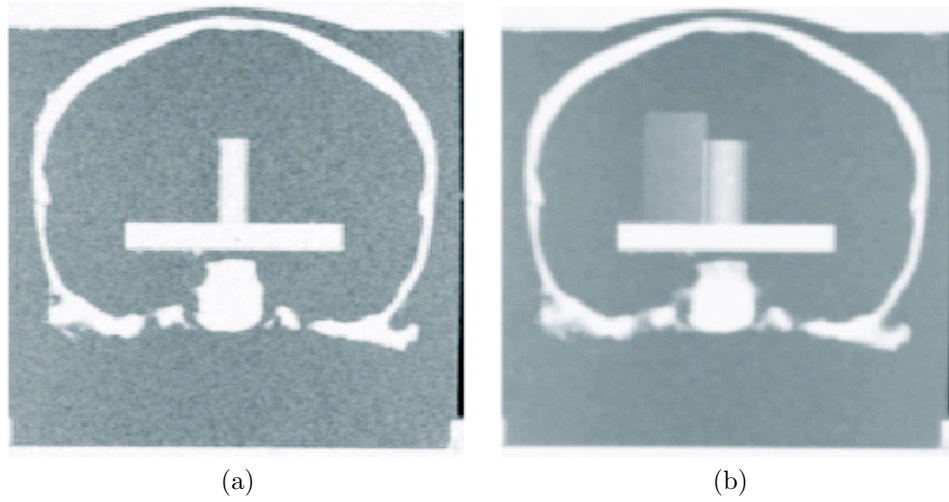


Figure 12.5: One slice of a volume dataset of the plastic skull before and after application of volume preprocessing steps (Gaussian filter and morphological operators). (a) Input volume, (b) preprocessed volume.

skipped. The result of postprocessing with a Gaussian filter operation is illustrated in Figure 12.6.

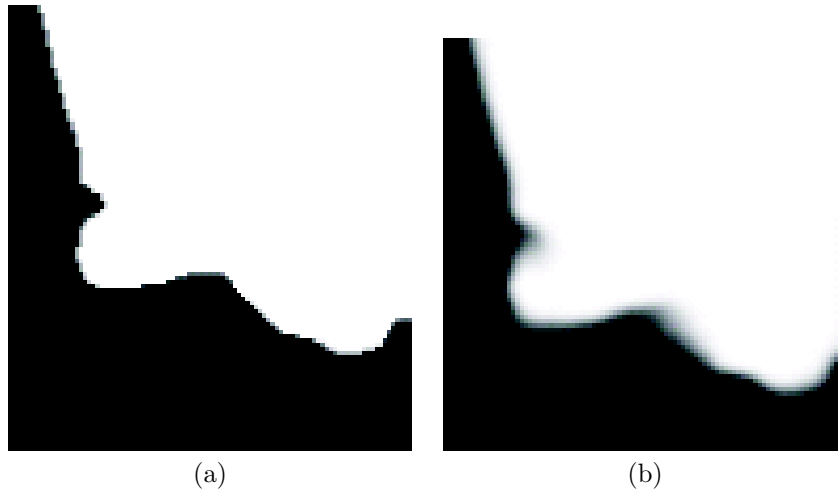


Figure 12.6: After the raycasting process, the visual hull volume is smoothed using a Gaussian filter. A magnified detail of one slice of a visual hull volume is shown. (a) Visual hull volume, (b) smoothed version.

If areas of full voxel intensity are located at the boundary of the volume dataset, the subsequent iso-surface extraction is unable to generate a suitable polygonal representation. In such cases, the computed iso-surface will have holes at the respective places of the visual hull volume. Such holes make the polygonal surface unsuitable for the task of occlusion handling. The user can thus select to apply a final postprocessing step to the volume, in which it is padded with a surrounding layer of zero intensity voxels. This additional layer causes the iso-surface ex-

traction algorithm to generate a closed outside surface for the whole volume. Figure 12.7 gives an overview of the entire volume processing pipeline used for computing the visual hull volume. Several of the pipeline's steps were implemented using respective functionality from the standard Visualization Toolkit (VTK) library [SML98].

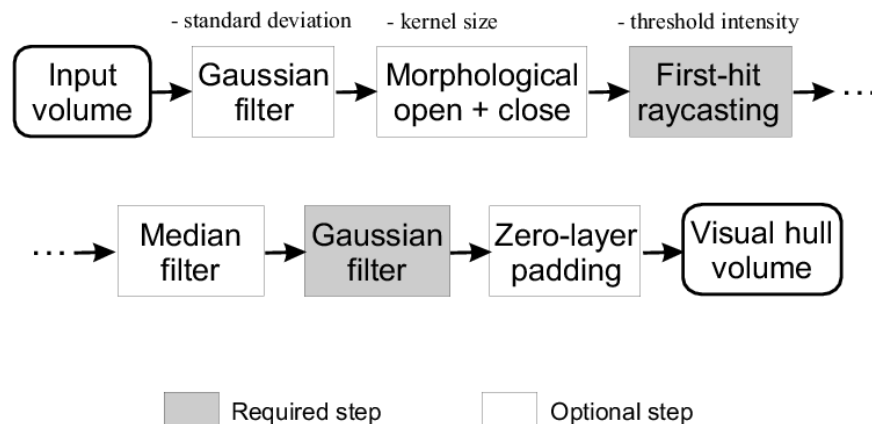


Figure 12.7: Overview of the volume processing pipeline involved in computing the visual hull volume. The steps in boxes with white backgrounds are optional and do not need to be applied in all cases. The items above some of the boxes list user-definable parameters for the respective processing steps.

12.4.2 Integration into Rendering Process

A polygonal iso-surface is extracted from the visual hull volume using the standard marching cubes algorithm [LC87]. Since the visual hull volume contains a representation of the outer surface as the interface between full and zero intensity, the half of the full intensity is selected as iso-surface threshold (128 in case of 8-bit volumes).

This visual hull iso-surface is then used for suppressing the display of occluded graphical objects during the augmented reality image composition process. In order to achieve this effect, the iso-surface is rendered in a special way before the actual virtual objects. The color buffers are disabled, while the visual hull surface is rendered, so that only the z-buffer (depth buffer) is altered. Afterwards, the color buffer access is enabled again. When the graphical objects are rendered, their color values overwrite the original camera image, but pixels lying behind the (invisible) visual hull are suppressed. Since the graphical model contained in the volumetric dataset corresponds to actual patient anatomy, the impression of virtual objects being occluded by the patient is created. This modified rendering process is illustrated in the following pseudocode, using OpenGL-like terminology:

```

glClear(GL_DEPTH_BUFFER_BIT);
drawCameraImage();
computeAndLoadTransformationMatrix();

```

Table 12.1: Dimensions and bit depth per voxel of test datasets

Dataset	Size X	Size Y	Size Z	Depth
Plastic skull	512	512	160	8 bits
Patient A	512	512	147	8 bits
Patient B	256	256	40	8 bits
CTHead	256	256	113	8 bits
MRBrain	256	256	109	8 bits

```
// Disable color buffer access
glDrawBuffer(GL_NONE);
renderOcclusionVisualHullSurface();

// Enable color buffer access
glDrawBuffer(GL_BACK);
renderVirtualObjects();
```

12.5 Results

We have tested the visual hull volume extraction with several datasets. Figure 12.8 gives an overview of the four medical volumes used for the evaluation, and their dimensions are listed in Table 12.5. "Plastic skull" (Figure 12.8a) is an MRI dataset of the plastic skull mock-up used in the actual augmented reality setup. Figures 12.8b and 12.8c show CT and MRI scans of the heads of patients who were treated at the department of maxillofacial surgery of the University Hospital Tübingen.

The last test datasets (Figure 12.8d and the MRBrain in Table 12.5) have been taken from the Stanford volume data archive. Each of the columns shows one slice from the input volume dataset and the corresponding slice in the computed visual hull volume. As illustrated in Figure 12.8, the visual hull volumes generated by our pipeline are an accurate representation of the outer surface of the patient's anatomy. Assuming that sufficiently good parameters for thresholds and filter kernel sizes are selected by the user, almost all computed volume slices are free of errors. The visual hull volume slice in Figure 12.8d shows one rare exception, where artifacts at the lower end of the skull were not removed by our algorithm. The defect in this case was caused by a physical contact between the skull of the patient and the CT tube has was placed in, which deteriorates the CT scan.

Figure 12.9 shows iso-surfaces generated from the MRBrain dataset. The first image (Figure 12.9a) is a rendering of the iso-surface extracted directly from the the original input volume. In Figure 12.9b, the iso-surface of the corresponding visual hull volume can be seen. As illustrated in the images, most of the detail of the anatomy's surface is preserved by our volume processing pipeline. A slight smoothing of the surface occurs due to the Gaussian filters used in the visual hull extraction process. The quality of the computed visual hull iso-surface is more than sufficient for the occlusion handling task, in particular

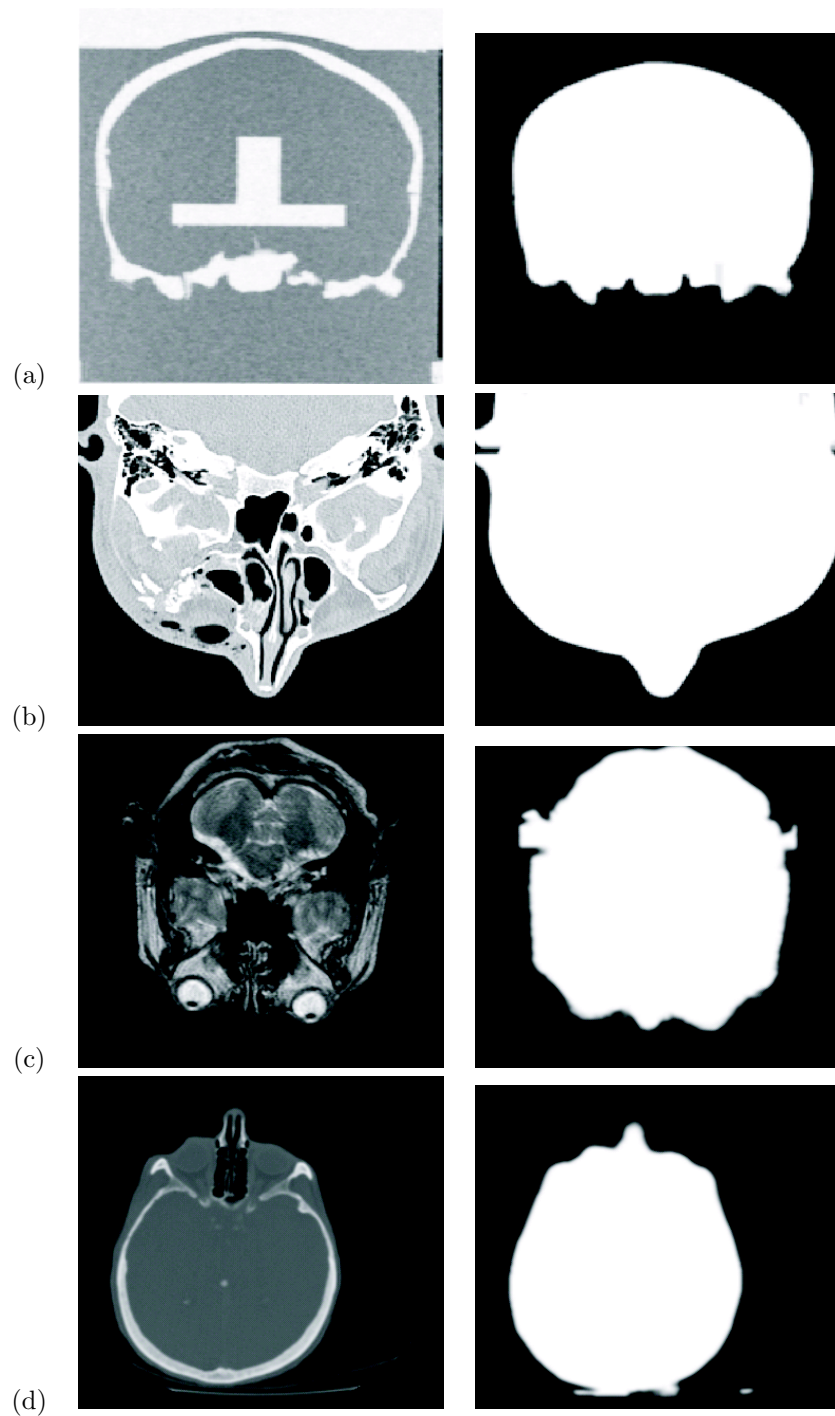


Figure 12.8: Visual hull volumes computed for four volume datasets. The first column shows one slice from each input volume, while the second column shows the corresponding slices from the visual hull volumes. The dataset in column (a) contains the plastic skull used in the actual AR system setup, columns (b) and (c) are from clinical cases at the University Hospital Tübingen, and (d) has been taken from the Stanford volume data archive.

Table 12.2: Triangle counts in iso-surfaces generated from unprocessed input and visual hull volumes

Dataset	Input	Visual hull	Reduction
Plastic skull	2,219K	1,279K	42.4%
Patient A	3,312K	1,627K	50.9%
Patient B	767K	230K	70.0%
CTHead	339K	283K	16.5%
MRBrain	433K	282K	34.9%
Average			42.9%

since it mainly depends on the accuracy of the surface silhouette.

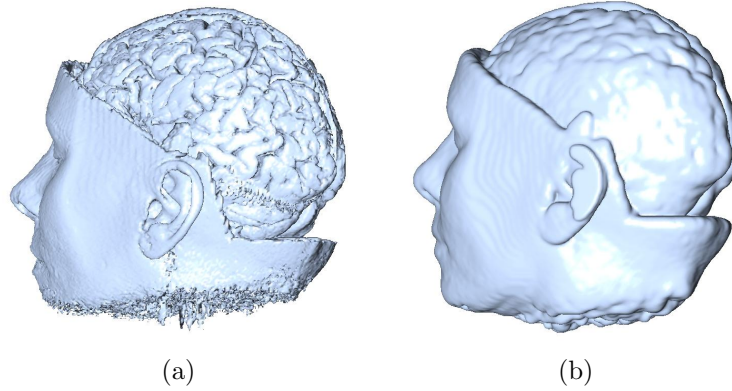


Figure 12.9: Comparison of the iso-surfaces generated from the original volume and visual hull volumes of the "MRBrain" dataset. The same viewpoint was used for rendering both images.

We have extracted iso-surfaces for both the original input and the visual hull volumes of all five test datasets. A comparison of the triangle counts of the generated surfaces is shown in Table 12.5. Threshold and filtering parameters in both cases were selected, so that an outer surface suitable for occlusion handling was obtained. As shown in the table, a triangle count reduction of at least one third is achieved in all but one cases. While reduction rates of close to or over 50% frequently occur, only dataset "CTHead" is not significantly simplified by our volume processing pipeline, due to the already low complexity of the input volume. The measured average triangle count reduction rate of 42.9% indicates our algorithm's simplification performance.

The effect of our occlusion handling method on augmented images is shown in Figure 12.10. In the first image (Figure 12.10a), a conventional augmented reality view can be seen. The long cylindrical shape is a virtual graphical object representing a surgical tool. This graphical model is positioned corresponding to an actual surgical tool mock-up held by the user. Although the surgical tool actually is behind the skull, the graphical model appears to be in front of it. The reason for this erratic display is that the geometry is always rendered over the camera

image regardless of the spatial relationships. Figure 12.10b illustrates that this problem is solved by our occlusion handling method. Although the situation is practically identical to the one shown in Figure 12.10a, the surgical tool model is now correctly occluded. Most of it cannot be seen because the rendering of the respective pixels is suppressed by the depth values of the skull's visual hull surface in the depth buffer. This creates the visual impression that the graphical model is actually occluded by the real skull itself. The simultaneous correct occlusion of multiple surgical tool models is shown in Figure 12.10c, in which the tips of two graphical tool models appear to be behind the skull. The fact that small details in the medical dataset are preserved by our volume preprocessing pipeline is illustrated in Figure 12.10d. In this image the surgical tool model is occluded by the visual hull iso-surface of the cheek-bone, while the geometry remains visible outside of it.

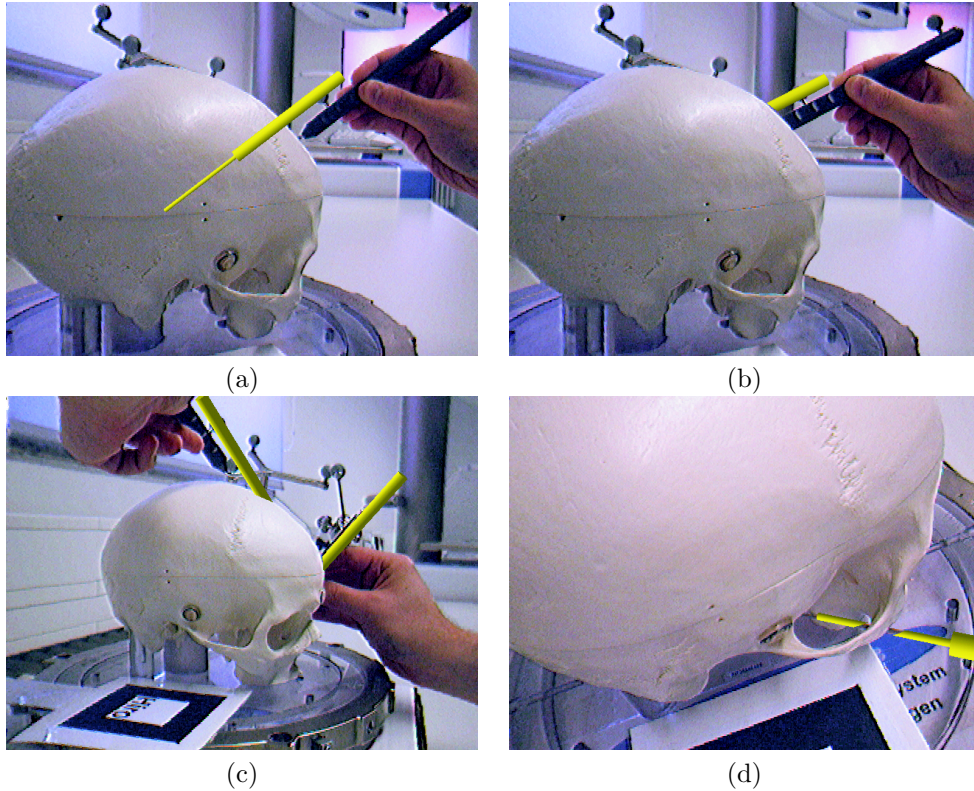


Figure 12.10: The effect of our occlusion handling method in the augmented reality setup. Graphical representations of surgical tools are used as virtual objects. While the virtual object in (a) is incorrectly rendered over the entire camera image (without application of our method), correct occlusion handling is shown in (b) to (d).

In order to examine the benefits of our visual hull volume extraction algorithm, we compare the times required for rendering iso-surfaces generated from unprocessed and processed volumes. Table 12.3 lists average rendering times measured during test sessions. Each test run was

Table 12.3: Average time required for rendering iso-surfaces into the depth buffer

Dataset	Input (msecs)	Visual hull (msecs)	Speedup
Plastic skull	35.7	17.4	105.2%
Patient A	51.7	23.3	121.9%
Patient B	14.4	6.4	125.0%
CTHead	4.6	3.7	24.3%
MRBrain	5.9	4.9	20.4%
Average			79.4%

at least 60 seconds long, during which similar typical camera movements were performed. The average time required for rendering the extracted iso-surface into the depth buffer was calculated over all frames generated during the test run (typically more than 1000 frames). The tests were performed on a high end computer system with an Intel Pentium 4 Xeon processor running at 2.66 GHz, 2 GBytes of main memory and a graphics card based on an NVidia GeForce FX 5900 chipset. While some of the absolute speedups listed in Table 12.3 may not seem very large, it has to be noted that rendering iso-surfaces comprising several 100K triangles takes considerably longer on most slower computer systems. Still, even in our tests the absolute reductions of ca. 20 milliseconds achieved for the datasets "Plastic skull" and "Patient A" are a significant improvement. The iso-surface rendering is embedded into a more complex augmented reality pipeline of video image acquisition, download of tracking information over the network and the final rendering stage. Thus rendering times in the range of 40 to 50 milliseconds for the occlusion handling can seriously hamper real-time image generation in the AR system. This is confirmed by a perceivable drop of the overall frame rate when iso-surface rendering is enabled for a larger dataset. As shown in Table 12.3, in terms of relative speedup we have generally measured considerable performance improvements through our algorithm. We assume this acceleration quality to scale well for cases in which larger datasets or several datasets are used.

12.6 Conclusion

We have presented a new method for handling occlusion in a medical augmented reality system. While the volume processing pipeline was designed for volumetric datasets containing patient anatomy, we believe that our approach can easily be employed for other applications in which volume datasets are available. These could include fields like engineering or production planning. Our tests have shown that our method is capable of generating easily understandable augmented images, in which virtual objects appear correctly occluded by the patient anatomy. This makes a better interpretation of the spatial relationships in the AR scene possible.

The visual hull volume extraction process aims at ensuring that the occlusion handling does not have a negative impact on the frame rate. In order to do so, an iso-surface is generated which contains a highly detailed representation of the patient anatomy's outer surface only. All inner structures are removed since they don't play a role for the occlusion handling algorithm.

It is conceivable that our first-hit raycasting strategy with rays parallel to coordinate axes might not work optimally for certain volumetric objects. However, we have found it to generate very good results for all our test datasets. Alternative approaches would include volume segmentation algorithms like region growing [RW92], which have a high computational complexity and often are not robust enough, when used with non-watertight objects. Our approach typically only takes a few seconds to compute even for large datasets, and it practically never leaks through the anatomy's outer surface. At the same time, the number of parameters to be selected by the user is kept at a minimum.

The typical approach to reducing the number of triangles of an iso-surface is to apply a mesh decimation algorithm (e.g. as described in [SZL92]). The major disadvantage of such methods in the context of occlusion handling is the fact that they do not distinguish between inner and outside surfaces. In order to achieve a similar reduction of the triangle count, a significant degradation of the quality of the outside surface geometry would have to be accepted. Still, we consider using a rather conservative mesh decimation as a postprocessing step on the visual hull surfaces generated by our pipeline. This could further reduce the number of triangles as long as sufficient visual quality of the outside hull is guaranteed.

The occlusion handling method described in this paper has been integrated into our medical augmented reality system *ARGUS*. *ARGUS* is designed to depend only on existing and commercially available medical equipment. We expect this quality to facilitate the transition of the AR system from an experimental state into clinical practice. The improved visual quality achieved through our occlusion handling method could further promote the acceptance of augmented reality in medicine.

Acknowledgements

We would like to thank the BrainLAB company (Heimstetten, Germany) for their continuing and very helpful support. In particular our research would not have been possible without the groundwork done by Markus Neff, on whose VectorVision Link software our augmented reality system is based.

The medical datasets used for developing and testing our algorithm were provided by our project partners at the University Hospital Tübingen. We would like to thank Frank Duffner and Dirk Freudenstein of the department of neurosurgery, as well as Jürgen Hoffmann and Dirk

Troitzsch of the department of maxillofacial surgery for the respective CT and MRI volume datasets.

This work has been supported by project VIRTUE in the focus program on "Medical Robotics and Navigation" of the German Research Foundation (DFG).

Bibliography

- [Azu97] R. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, 1997.
- [Ber97] M. Berger. Resolving Occlusion in Augmented Reality: A Contour Based Approach without 3d Reconstruction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 91–96, 1997.
- [Bra02] BrainLAB. Neurosurgery Product Brochure. BrainLAB AG, 2002.
- [BWRT96] D. Breen, R. Whitaker, E. Rose, and M. Tuceryan. Interactive Occlusion and Automatic Object Placement for Augmented Reality. *Computer Graphics Forum*, 15(3):11–22, 1996.
- [DIC03] DICOM Standards Committee. *Digital Imaging and Communications in Medicine (DICOM)*. National Electrical Manufacturers Association, 2003.
- [FHFG99] A. Fuhrmann, G. Hesina, F. Faure, and M. Gervautz. Occlusion in Collaborative Augmented Environments. *Computers & Graphics*, 23(6):809–819, 1999.
- [FNBF04] J. Fischer, M. Neff, D. Bartz, and D. Freudenstein. Medical Augmented Reality based on Commercial Image Guided Surgery. In *Proc. of Eurographics Symposium on Virtual Environments*, pages 83–86, 2004.
- [FRB03] J. Fischer, H. Regenbrecht, and G. Barattoff. Detecting Dynamic Occlusion in front of Static Backgrounds for AR Scenes. In *Proc. of Eurographics Symposium on Virtual Environments*, pages 153–161, 2003.
- [GBB⁺02] G. Gordon, M. Billinghamurst, M. Bell, J. Woodfill, B. Kowalik, A. Erendi, and J. Tilander. The Use of Dense Stereo Range Data in Augmented Reality. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2002.

- [Goe03] G. Goebbels. ARSyS-Tricorder - Entwicklung eines Augmented Reality Systems für die intraoperative Navigation in der MKG Chirurgie. In *Jahrestagung der Deutschen Gesellschaft für Computer-und Roboterassistierte Chirurgie e. V. (CURAC)*, 2003.
- [KB99] H. Kato and M. Billinghurst. Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In *Proc. of IEEE and ACM International Workshop on Augmented Reality*, pages 85–94, 1999.
- [Kli01] G. Klinker. Praktikum Augmented Reality - Occlusion Handling Between Real and Virtual Objects. <http://www.bruegge.in.tum.de/teaching/ss01/AR-Praktikum01/presentation/occlusions.pdf>, 2001.
- [KOTY00] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya. A Stereoscopic Video See-through Augmented Reality System Based on Real-time Vision-based Registration. In *Proc. of IEEE Virtual Reality*, pages 255–262, 2000.
- [LB00] V. Lepetit and M. Berger. A Semi-Automatic Method for Resolving Occlusion in Augmented Reality. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.
- [LC87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. of ACM SIGGRAPH*, pages 163–169, 1987.
- [MMR02] S. Malik, C. McDonald, and G. Roth. Hand Tracking for Interactive Pattern-based Augmented Reality. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2002.
- [ND03] M. Neff and K. Diepold. Design and Implementation of an Interface Facilitating Data Exchange between an IGS System and External Image Processing Software. Diplomarbeit, TU München, 2003.
- [RW92] Gonzalez R and R. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [SFW02] R. Splechtna, A. Fuhrmann, and R. Wegenkittl. ARAS - Augmented Reality Aided Surgery System Description. Technical report, VRVis Research Center for Virtual Reality and Visualization, 2002.
- [SLH⁺96] A. State, M. Livingston, G. Hirota, W. Garrett, M. Whitton, H. Fuchs, and E. Pisano. Technologies for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle

- Biopsies. In *Proc. of ACM SIGGRAPH*, pages 439–446, 1996.
- [SML98] W. Schroeder, K. Martin, and W. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 2nd edition, 1998.
- [SSW02] B. Schwald, H. Seibert, and T. Weller. A Flexible Tracking Concept Applied to Medical Scenarios Using an AR Window. In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 261–262, 2002.
- [SZL92] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of Triangle Meshes. In *Proc. of ACM SIGGRAPH*, pages 65–70, 1992.
- [WA95] M. Wloka and B. Anderson. Resolving Occlusion in Augmented Reality. In *Proc. of ACM Symposium on Interactive 3D Graphics*, pages 5–12, 1995.