

ADVANCED ILLUMINATION AND VIEW- SELECTION TECHNIQUES FOR VOLUME RENDERING AND ITS APPLICATION TO MEDICAL IMAGING

Marc Ruiz Altisent

Per citar o enllaçar aquest document:
Para citar o enlazar este documento:
Use this url to cite or link to this publication:
<http://hdl.handle.net/10803/371154>



<http://creativecommons.org/licenses/by/4.0/deed.ca>

Aquesta obra està subjecta a una llicència Creative Commons Reconeixement

Esta obra está bajo una licencia Creative Commons Reconocimiento

This work is licensed under a Creative Commons Attribution licence



PHD THESIS

**Advanced illumination and
view-selection techniques for
volume rendering and its
application to medical imaging**

Author:

Marc RUIZ ALTISENT

2012

Programa de Doctorat en Tecnologia

Advisors:

Dra. Imma BOADA OLIVERAS

Dr. Miquel FEIXAS FEIXAS

Dr. Mateu SBERT CASASAYAS

Memòria presentada per optar al títol de doctor per la Universitat de Girona

Advanced illumination and view-selection techniques for volume rendering and its application to medical imaging

Abstract: Volume visualization is a method of extracting meaningful information from volumetric data using interactive graphics and imaging. Volume visualization aims to assist visual interpretation of (medical) data by creating three-dimensional models that reproduce real three-dimensional objects with sufficient detail and speed to support interactive manipulation. Illustrative rendering is used to enhance the perception of structure, shape, orientation, and depth relationships in a volume model.

There are two major goals in volume rendering. The first one is to obtain high quality images with a low computational cost, allowing real time exploration. The second goal is to determine how to explore volume datasets in an effective way.

In this thesis we want to advance in these two research lines. We propose several methods to approximate global illumination with low computational cost, permitting realistic and illustrative rendering. We also present a method to automatize the transfer function definition given a target distribution. In addition, we introduce a method to automatize the creation of exploded views. Finally, we explore a few techniques to select the best viewpoints for a volume. An important part of our contributions are based on information theory.

Tècniques avançades d'il·luminació i selecció de vistes per a la visualització de volums i la seva aplicació a imatge mèdica

Resum: La visualització de volums és un mètode per extreure informació útil de dades volumètriques mitjançant imatges interactives. L'objectiu de la visualització de volums és assistir la interpretació visual de dades (mèdiques) creant models tridimensionals que reproduïxen objectes tridimensionals reals amb suficient detall i velocitat per permetre'n la manipulació interactivament. La visualització il·lustrativa s'utilitza per millorar la percepció de l'estructura, la forma, l'orientació i la profunditat en un model de volum.

Hi ha dos objectius principals en la visualització de volums. El primer és obtenir imatges d'alta qualitat amb un cost computacional baix, permetent l'exploració en temps real. El segon objectiu és determinar com explorar volums d'una manera efectiva.

En aquesta tesi ens proposem avançar en aquestes dues línies de recerca. En aquest sentit proposem diversos mètodes per aproximar la il·luminació global amb un cost computacional baix, permetent visualització realista i il·lustrativa. També presentem un mètode per automatitzar la definició de funcions de transferència donada una distribució objectiu. A més a més, presentem un mètode per automatitzar la creació d'*exploded views*. Finalment, explorem algunes tècniques per seleccionar els millors punts de vista d'un volum. Una part important de les nostres contribucions estan basades en la teoria de la informació.

Agraïments

Per començar vull agrair molt especialment als meus directors, Imma Boada, Miquel Feixas i Mateu Sbert, les idees, ajuda, treball i suport abans i durant el doctorat, que han fet possible aquesta tesi. També vull agrair especialment a l'Anton Bardera i l'Ivan Viola les seves imprescindibles contribucions i consells en molts dels articles, aixó com també la col·laboració de l'Stefan Bruckner, en László Szirmay-Kalos i en Tamás Umenhoffer. Agraïxo també el suport moral i l'ajuda dels companys de despatx i de doctorat: Anton, Ester, Ferran, Fran, Isma, Mario, Miquel, Nacho, Narcís, Nico, Olga, Roger, Tere. . . Per la gran feina desenvolupant la plataforma que ha servit com a base per implementar tots els mètodes d'aquesta tesi, vull agrair el seu treball a l'equip Starviewer: Anton, Chus, Ester, Marc, Marc, Quim, Roger, Rubén. . .

Vull agrair també la contribució indirecta però imprescindible de moltíssima gent que m'ha ajudat a distreure'm del doctorat i divertir-me. Entre ells, molts dels ja esmentats, a més d'altres companys del GILab i el departament d'IMA. També els amics amb qui comparteixo tot sovint dinars, sopars, viatges i altres activitats: Cargol, Carles, Carre, Fran, Marcel, Patri, Salvà, Soms, Xivi, Xuc. . . La família: pares, germà, cosins, oncles, àvies. . . I també voldria agrair la contribució directa o indirecta a la realització de la tesi de diverses persones que segur que m'he oblidat d'esmentar.

I per acabar, m'agradaria agrair a en Pep que hagi fet del Barça el millor equip de la història i a l'equip de l'APM? que hagi descobert tantes frases que ja s'han incorporat a la cultura popular. Ambdós han contribuït en certa manera a aquest treball ;)

Acknowledgements

The work in this thesis has been supported by the FPU grant, TIN2007-68066-C04-01, TIN2007-67982-C02, and TIN2010-21089-C03-01 of the Ministry of Education and Science (Spanish Government), by the 2009 SGR 643 from the Generalitat de Catalunya, by the MedViz Initiative in Bergen (medviz.uib.no), and by the Austrian Science Fund (FWF) grant no. P18322.

The stag beetle data set has been provided by Georg Glaeser, Vienna University of Applied Arts, Austria and Johannes Kastner, Wels College of Engineering, Austria. The salmon CT data set is the courtesy of Prof. Ole Brix, Christian Michelsen Research, Norway. The CT-head data set is courtesy of North Carolina Memorial Hospital, US (www9.informatik.uni-erlangen.de/External/vollib/). The CT-body data set is courtesy of the Visible Human Project, National Library of Medicine, US (www.nlm.nih.gov/research/visible/).

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Thesis outline	3
2	State of the art	7
2.1	Introduction	7
2.2	The volume visualization pipeline	8
2.2.1	Data acquisition	9
2.2.2	Data processing	10
2.2.3	Data rendering	11
2.3	Direct volume rendering	12
2.3.1	Direct volume rendering steps	12
2.3.2	Direct volume rendering algorithms	15
2.4	Transfer function definition	17
2.5	Illumination models and obscurances	19
2.5.1	Volume shadowing	20
2.5.2	Obscurances and ambient occlusion	21
2.6	Volume illustration	22
2.7	Information theory tools	24
2.7.1	Entropy	24
2.7.2	Kullback-Leibler divergence and mutual information	26
2.7.3	Inequalities	28
2.7.4	Entropy rate	30
2.7.5	Entropy and coding	31
2.7.6	Information bottleneck method	32
2.8	Information theory in computer graphics	33
2.9	Viewpoint selection and visibility channel in volume visualization	34
3	Visibility channel	37
3.1	Introduction	37
3.2	Background	38
3.2.1	Volume shadowing and illustrative techniques	38
3.2.2	Visibility channel	38
3.3	Voxel information	39
3.4	Illustrative visualization using voxel information	42
3.4.1	Ambient occlusion	42
3.4.2	Color ambient occlusion	47
3.4.3	Focus+context	47
3.5	Viewpoint selection using voxel information	50

3.6	Conclusions	51
4	Similarity-based exploded views	53
4.1	Introduction	53
4.2	Background	54
4.3	Similarity-steered visualization	54
4.4	Selection of structured views	56
4.5	Evaluating similarity	59
4.5.1	Model partitioning	59
4.5.2	Slice grouping	60
4.5.3	Results	61
4.6	Conclusions	63
5	Obscurance-based volume rendering framework	65
5.1	Introduction	65
5.2	Background	66
5.2.1	Obscurances and ambient occlusion	66
5.2.2	Volume shadowing versus obscurances	67
5.3	Obscurances for volume rendering	68
5.3.1	Algorithm	68
5.3.2	Analysis of ρ function	70
5.4	Volume saliency	73
5.5	Applications	75
5.5.1	Realistic and illustrative rendering	77
5.5.2	Saliency	77
5.6	Conclusions	78
6	Volumetric ambient occlusion for volumetric models	81
6.1	Introduction	81
6.2	Background	82
6.2.1	Obscurances and ambient occlusion	82
6.2.2	Volume shadowing	83
6.3	New approaches for volumetric ambient occlusion	83
6.3.1	Ambient occlusion as CDF	84
6.3.2	Reconstruction of the cumulative distribution	85
6.3.3	Gaussian cumulative distribution	86
6.4	Results	86
6.4.1	Ambient occlusion maps	87
6.4.2	Opacity modulation	90
6.5	Conclusions	93
7	Interactive volume illustration using intensity filtering	95
7.1	Introduction	95
7.2	Background	97
7.3	Proposed approach	97

7.3.1	Spatial importance map	97
7.3.2	Implementation	99
7.3.3	Modulation and illustration	103
7.4	Applications	103
7.4.1	Color modulation	103
7.4.2	Opacity modulation	105
7.4.3	Stippling	107
7.5	Conclusions	109
8	Automatic transfer functions based on informational divergence	111
8.1	Introduction	112
8.2	Background	113
8.2.1	Transfer function design	113
8.2.2	Information theory in computer graphics	113
8.3	Framework and motivation	113
8.3.1	Visibility channel	113
8.3.2	Motivation	115
8.4	Method	117
8.4.1	Target distribution	118
8.4.2	Objective function	120
8.4.3	Optimizer	121
8.5	Results	126
8.5.1	Limitations	133
8.6	Conclusions	133
9	Conclusions	135
9.1	Contributions	135
9.2	Future work	138
	Bibliography	139

List of figures

1.1	Graphical representations of the skull from Da Vinci's anatomical notebooks and generated from computed tomography data using advanced volume rendering techniques	2
2.1	Volume rendering pipeline	9
2.2	Comparison of low and high sampling rates	13
2.3	Comparison of pre-classification and post-classification	14
2.4	Comparison of local illumination and global illumination	19
2.5	Illustration of the main parameters involved in the obscurances computation	21
2.6	Example of stipple rendering	23
2.7	Plot of binary entropy	25
2.8	Venn diagram of Shannon's information measures	28
2.9	Visibility channel	35
3.1	Reversed visibility channel	39
3.2	VOMI maps generated for different models and transfer functions	40
3.3	Overview of the VOMI pipeline	41
3.4	AO maps obtained from VOMI with different numbers of viewpoints	43
3.5	Comparison of AO maps obtained with different methods	43
3.6	Ambient lighting on the CT-body with different AO maps	45
3.7	Local lighting on the CT-body with different AO maps applied as an additive term	46
3.8	CT-body with different transfer functions and illustrative effects	48
3.9	Opacity modulation of the CT-body	49
3.10	Opacity modulation of the CT-body	50
3.11	Selected viewpoints according to viewpoint entropy, viewpoint mutual information, and informativeness	52
4.1	Main steps of the selection of the most structured view	55
4.2	Top-down and bottom-up partitioning schemes	56
4.3	Most structured views for different volumes	58
4.4	Volume partitioning and slice grouping applied to a CT scan of the human body	60
4.5	Volume partitioning and slice grouping applied to a CT scan of a patient with a brain hemorrhage and to an histological data model	62
5.1	CT-body rendered with the obscurance-based volume rendering framework	66
5.2	Illustration of local volumetric shadowing and obscurance computation	69

5.3	Different $\rho(d)$ functions	71
5.4	Obscurances, obscurances with color bleeding, and obscurance-based illumination model, considering different ρ functions	72
5.5	Obscurance volumes of the aneurism and CT-body with different parameters	74
5.6	Saliency maps and illustrative visualizations obtained with a saliency-based opacity modulation	76
5.7	Saliency-based opacity modulation on the CT-body	78
6.1	Ambient occlusion as the cumulative distribution function	83
6.2	Ambient occlusion is approximated as the probability of a voxel in the bounding box of the tangent sphere having higher density than the processed voxel	84
6.3	Approximation of the cumulative distribution function with t^β	85
6.4	The Gaussian CDF with different parameters	87
6.5	AO maps for CT-body using different methods	88
6.6	AO maps for CT-head using different methods and radii	89
6.7	Ambient lighting for CT-body using different methods.	91
6.8	Ambient lighting for CT-head using different methods.	92
6.9	Opacity modulation of CT-body using different methods	93
6.10	Opacity modulation of CT-head using different methods	94
7.1	Main steps of volume illustration using intensity filtering	96
7.2	Spatial importance maps obtained for different data sets	98
7.3	Color modulation driven by spatial importance map of the CT-body with ambient lighting	100
7.4	Color modulation driven by absolute spatial importance map of the CT-body with local lighting	102
7.5	Different opacity modulation effects with different parameters applied to the CT-head model	104
7.6	Different opacity modulation effects using a default transfer function applied to the CT-head model	106
7.7	Stipple rendering of the CT-head modulated by the spatial importance map with different parameters	107
7.8	Stipple rendering of the CT-body modulated by the spatial importance map with different parameters	108
7.9	CT-body rendered combining stippling and opacity modulation effects	109
8.1	Volume renderings of the tooth data set using transfer functions obtained with different target distributions	111
8.2	Process of finding the optimal transfer function	117
8.3	Graph depicting the target distribution and the initial and final visibility distributions in a specific run	122

8.4	Evolution of the objective function $F(A)$ with respect to the number of iterations in a specific run	123
8.5	Global strategy applied to the CT-head with different target distributions	127
8.6	Importance-based strategy applied to the CT-head with different target distributions	128
8.7	Importance-based strategy applied to two MRI brain data sets with different masks	130
8.8	CT-body considering different values of β	131
8.9	CT-tooth considering 1 viewpoint and 42 viewpoints	131
8.10	CT-carp considering different numbers of viewpoints and $\beta = 0.5$	132

List of tables

3.1	Times to compute VOMI	42
4.1	Times to compute the most structured view and to perform volume partitioning and slice grouping	62
5.1	Times to compute obscurances	73
6.1	Times to compute the AO maps	90
7.1	Times to compute the spatial importance map	101
8.1	Number of iterations and time cost to find the optimal transfer function	133

Introduction

Contents

1.1 Motivation	1
1.2 Objectives	2
1.3 Thesis outline	3

1.1 Motivation

Graphical representations are a key component in the description of scientific processes and the comprehension of scientific data sets. In medical applications, for instance, data acquired by medical devices can be interpreted once medical images have been generated. Another example are oil explorations where the visualization of geoseismic data is crucial for determining the right perforation place. As these, many different examples can be found in many scientific fields and application areas, such as material sciences, fluid dynamics, or environmental sciences, just to name a few. Obtaining graphical representations has always been of human interest and the applied strategies have evolved from the hand made representations, such as the Da Vinci's ones, to current computer-based images. Obviously, the basis of these approaches are very different. While in the former drawing skills are required, in the latter specialized algorithms and data structures capable to transform data into visually comprehensible images are needed (see Figure 1.1). The development of such algorithms has become an important focus of research and it has led to one of the major lines of investigation in computer science: *visualization*.

The aim of visualization is to communicate information clearly and effectively through graphical means. To reach this objective several approaches have been proposed. They can be classified according to different features, such as the input data they deal with (one-, two-, three-, or four-dimensional), the rendering primitive used to generate the image (point-, line-, triangle-, or voxel-based), the method applied to generate the final rendering (backward or forward projection), the rendering style (photorealistic or non-photorealistic), or the degree of user interaction required to perform the processes involved in the rendering. In this thesis, we deal with 3D volume data sets with special emphasis on medical data. An important feature of this data sets is its regular spacial distribution which allows them to be represented in a regular

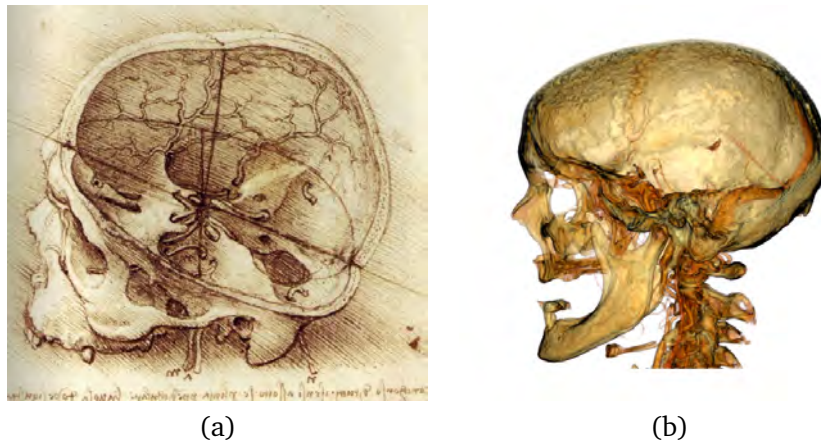


Figure 1.1: Graphical representations of the skull (a) from Da Vinci's anatomical notebooks and (b) generated from computed tomography data using advanced volume rendering techniques.

grid known as voxel model. Due to the importance of medical visualization, this has established itself as a research area.

In this thesis we will focus our interest on how to obtain high quality realistic and illustrative volume renderings. Our research topics will also include the automation of the procedures involved in volume rendering, as well as volume exploration strategies. We also want to go one step further towards overcoming some of the limitations related to these topics. Most of our contributions will be based on information theory.

1.2 Objectives

The main goal of this thesis is to develop new strategies to enhance the volume rendering process and the way to explore data, with special interest on medical environments. To reach this objective we aim to:

- Achieve high quality realistic and illustrative renderings.
 - Approximate global illumination for volume rendering.

Global illumination techniques allow to obtain high quality renderings by simulating physical light interactions. However, their application to volume rendering is difficult because of their high computational cost. In polygonal rendering there are several techniques, such as ambient occlusion and obscurances, that approximate global illumination at low cost. We aim to adapt these techniques to volume rendering.
 - Develop a filtering-based technique to produce illustrative renderings.

Illustrative visualization enhances the expressiveness of volume rendering by applying hand-crafted illustrative techniques. These techniques enhance the perception of structure, shape, orientation, and depth relationships in

a volume model. We want to develop a technique based on image filtering that allows to obtain such kind of renderings.

- Use separable filtering to compute volumetric ambient occlusion.

A novel interpretation of ambient occlusion measures the portion of the tangent sphere of a surface that belongs to the set of occluded points. We aim to extend this approximation to volumetric models and implement it using separable filtering to achieve a real time interaction.

- Use information theory to produce realistic and illustrative renderings.

Information theory deals with the transmission, storage, and processing of information and is used in fields such as image processing and computer graphics. We want to develop a volume rendering framework based on the information channel constructed between the volumetric dataset and a set of viewpoints. Using information theory measures obtained from this channel, we want to produce both realistic and non-photorealistic visualizations.

- Develop a semi-automatic method to create opacity transfer functions.

The transfer function definition is a main step in volume rendering. The transfer function maps scalar values in volume datasets to optical properties such as color and opacity. Manually creating a transfer function is a tedious and time-consuming process. We want to automate the process of defining the opacities. Our approach will be based on an information channel between the a set of viewpoints and the intensities of the volume dataset.

- Develop new exploration techniques.

- Exploded views of volume models.

Exploded views are often used in illustration to overcome the problem of occlusion when depicting complex structures. To create exploded views, first the parts to be separated have to be found. We want to automate the definition of these parts or slabs with the help of information theory.

- Use information theory to select the best viewpoints.

One of the problems in volume visualization is to find the best views of the volume, the ones that are more helpful to the user. On the one hand, the information channel between the volume dataset and the viewpoints that we have proposed previously should also provide information that allows to find the best views. On the other hand, information-theoretic tools could also help us to find the most structured viewpoints.

1.3 Thesis outline

This dissertation is organized in nine chapters. Following this introduction, the following eight chapters remain:

- Chapter 2: **State of the art**

In this chapter, the background on volume rendering required for the comprehension of the main issues that are going to be analyzed in this thesis is introduced. The main concepts of information theory are also reviewed since it is the basis of most of our contributions.

- Chapter 3: **Visibility channel**

This chapter introduces a volume rendering framework based on the information channel constructed between the volumetric data set and a set of viewpoints. Information obtained from this channel is used to produce realistic and illustrative renderings and also to select the most informative views.

- Chapter 4: **Similarity-based exploded views**

Exploded views are often used in illustration to overcome the problem of occlusion when depicting complex structures. This chapter presents an information-theoretic technique that automatically decides how to partition the volume to create exploded views. The thickness of slabs is driven by the similarity between partitions.

- Chapter 5: **Obscurance-based volume rendering framework**

Obscurances is a technology that produces natural-looking lighting effects in a faster way than global illumination. An obscurance-based framework that allows to obtain realistic and illustrative renderings interactively is presented in this chapter. The concept of saliency as the gradient of obscurances is also introduced and used to enhance volume visualization and to select the most salient views.

- Chapter 6: **Volumetric ambient occlusion for volumetric models**

In this chapter, volumetric ambient occlusion is defined as the proportion of the tangent sphere of a surface that is occluded, and statistically robust estimates for the ambient occlusion value are proposed. Separable filters are used to compute the data needed to estimate this proportion at interactive rates.

- Chapter 7: **Interactive volume illustration using intensity filtering**

This chapter presents a simple and interactive technique for volume illustration that uses the difference between the original intensity values and a low-pass filtered copy. This difference provides a spatial importance map that captures salient and separability information about regions in the volume. This map is used to produce different illustrative effects.

- Chapter 8: **Automatic transfer functions based on informational divergence**

In this chapter, a framework to define transfer functions from a target distribution provided by the user is presented. The basis of the framework is a communication channel between a set of viewpoints and a set of bins of a volume data set, and it supports 1D as well as 2D transfer functions including the gradient information.

- Chapter 9: **Conclusions**

In this last chapter, conclusions of the thesis and future work will be presented, along with a summary of the publications related with this thesis.

State of the art

Contents

2.1	Introduction	7
2.2	The volume visualization pipeline	8
2.2.1	Data acquisition	9
2.2.2	Data processing	10
2.2.3	Data rendering	11
2.3	Direct volume rendering	12
2.3.1	Direct volume rendering steps	12
2.3.2	Direct volume rendering algorithms	15
2.4	Transfer function definition	17
2.5	Illumination models and obscurances	19
2.5.1	Volume shadowing	20
2.5.2	Obscurances and ambient occlusion	21
2.6	Volume illustration	22
2.7	Information theory tools	24
2.7.1	Entropy	24
2.7.2	Kullback-Leibler divergence and mutual information	26
2.7.3	Inequalities	28
2.7.4	Entropy rate	30
2.7.5	Entropy and coding	31
2.7.6	Information bottleneck method	32
2.8	Information theory in computer graphics	33
2.9	Viewpoint selection and visibility channel in volume visualization	34

2.1 Introduction

Volume visualization is a method of extracting meaningful information from volumetric data using interactive graphics and imaging, and it is concerned with volume data representation, modeling, manipulation and rendering [Kaufman 1990]. In the last decades, different techniques have been proposed to render volumetric data. At broad level these can be classified, according to the used graphic primitive, into direct or

indirect volume rendering techniques. In this thesis we will focus on direct volume rendering techniques, which attempt to capture all the volume data in a single 2D image without considering any intermediate representation, i.e. operating on samples from the actual data.

Volume visualization aims to assist visual interpretation of (medical) data by creating three-dimensional models that reproduce real three-dimensional objects with sufficient detail and speed to support interactive manipulation. Although many different strategies have been proposed with this objective, it is still a challenge to obtain renderings that adapt the appearance of the data to user needs in an interactive manner. The difficulty is to obtain high quality images fast enough to ensure efficient exploration. Moreover, one has to take into account user needs. For instance, while in a medical environment realistic renderings are suitable for diagnosis and pre-operative planning, non-photorealistic techniques are preferred in educational environments since they are able to simplify data, producing clearer images than traditional photorealistic methods. One handicap in this area is that computing global illumination is an expensive operation, thus it has to be approximated using techniques such as ambient occlusion and obscurances, which will be explained in Section 2.5.2. Providing strategies to explore volume data in an efficient manner is also of great interest. To ensure interactivity when exploring volume data, different strategies to accelerate the rendering process can be considered. Among them, focus+context and viewpoint-based strategies, with GPU-based implementations that exploit hardware capabilities.

This chapter is structured as follows. Section 2.2 provides an overview of the volume visualization pipeline common to most rendering methods. Section 2.3 describes more detailedly the steps that are common among direct volume rendering methods, and also presents the main algorithms in this category. Section 2.4 introduces the main problems in transfer function definition and cites several currently proposed solutions. In Section 2.5 local and global illumination models are compared, and various methods to approximate global illumination effects at a low cost are introduced, giving emphasis on obscurances. Section 2.6 gives an overview on illustrative volume rendering, citing several approaches by different authors. Section 2.7 explains the concepts on information theory that will be used in this thesis. Section 2.8 briefly reviews several applications of information theory to computer graphics. Section 2.9 presents an information channel between a set of viewpoints and the set of voxels of the volume model. The reversion of this channel is the basis of some of the methods proposed in this thesis.

2.2 The volume visualization pipeline

The so-called volume visualization pipeline defines the different steps that are required to generate an image from abstract volume data. The main steps of this pipeline are represented in Fig. 2.1 and described below.

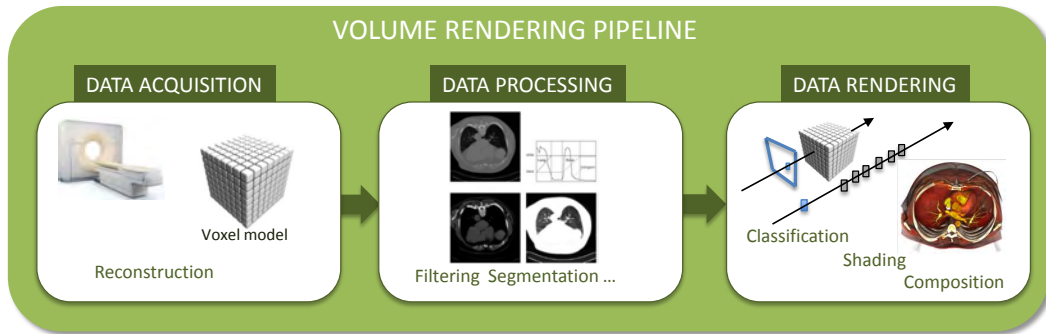


Figure 2.1: Volume rendering pipeline.

2.2.1 Data acquisition

The first step of the volume rendering pipeline is data acquisition. There are three main data sources: simulation, modeling, and scanning. This last one is the most common in medical environments and requires specialized medical scanners to sample data from real models. Scanners represent information as a specific type of image such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), single photon emission computed tomography (SPECT), and ultrasound (US). The physical principles of the scanner determine the type of data that will be acquired and represented in the images. For instance, CT represents density and is suitable for bone exploration, MRI represents oscillation and is suitable for soft tissue exploration, and US represents echoes and is suitable for vessel exploration.

Scanned data is provided at a specific spatial intensity resolution and scale with a given range of capture error. Most devices capture information following a regular planar distribution, which allows to represent data in a regular grid known as a *voxel model*. Although other representation schemes are possible, the voxel model is the most common in our context.

The voxel model is based on the decomposition of the 3D space into a regular set of identical cuboids, known as *voxels*, whose edges are parallel to the coordinate axis. This spatial structure allows for voxels to be directly represented by a 3D point (i, j, k) , where $1 \leq i, j, k \leq n_i, n_j, n_k$, being n_i, n_j, n_k the number of voxels per axis from which the whole geometrical and topological information may be retrieved.

The acquired data is usually represented as a set of tuples (x, y, z, v) , representing the value v of some property at a certain 3D location (x, y, z) . Samples may be taken along time, and then we have one more component: (x, y, z, t, v) . The value v can be a scalar, a vector, or a tensor. In this thesis we will deal with 3D volumes with scalar values.

While it is straightforward to consider each voxel as a cube with an homogeneous value in its whole volume (also known as the *voxel approach*), it is more useful for rendering purposes to interpret each value in the dataset as a sample obtained from a

continuous three-dimensional function (also known as the *cell approach*):

$$f(p) \in \mathbb{R} \quad \text{with} \quad p \in \mathbb{R}^3. \quad (2.1)$$

With this interpretation we can reconstruct the information at any location of the model. Two of the most common reconstruction filters are nearest neighbour and trilinear interpolation. Nearest neighbour or zero order interpolation takes the value of the closest cell vertex; it is very fast but models look blocky. On the other hand, trilinear or first order interpolation computes linear interpolation from the closest values in each axis; the result is smoother than with nearest neighbour interpolation, but it can produce blurry images or 3D diamond-like structures. More sophisticated interpolation methods exist, like tricubic or spline-based interpolations, but their cost is high and thus they are not frequently used. When rendering with the GPU, nearest neighbour and linear interpolation are already provided by the hardware, but higher order methods can be programmed [Ruijters 2008, Hadwiger 2009]. The main requirements of reconstruction filters are high quality reconstruction but small performance overhead.

2.2.2 Data processing

Data processing is the second step of the visualization pipeline. It includes the different techniques that are applied to extract and communicate information more effectively according to user and application requirements. Below, we review some of the most representative techniques.

- **Filtering**

Acquired images may be filtered to minimize undesirable effects such as noise or artifacts, commonly produced in the acquisition process. Some of the most popular filters are Gaussian filters, which have a blurring effect, and anisotropic diffusion filters, which preserve the edges of the image. [Gerig 1992, Castaño Moraga 2007]

- **Registration**

Image registration is a process that determines the spatial transformation that will bring two different images into alignment. This operation is of special interest in clinical practice since it allows us to combine in a single model the information acquired with different devices, at different times or from different patients. Image registration allows us, for instance, to complement different data in order to complete the anatomical representation or combine functional information with anatomical representation. This capability is of special interest if we are interested in performing studies between different patients and carrying out studies along time or inpatient studies. For a review of registration techniques, see [Pluim 2003].

- **Segmentation**

Segmentation is defined as the division of an image or volume into coherent regions using some local image characteristic criteria. After segmentation each region is labeled with a distinct value. Segmentation is useful for applications that need to differentiate regions that represent different tissues but have overlapping ranges of scalar values, in the identification of pathological areas, for measuring the volume of a lesion, etc. For a detailed review of segmentation strategies, see [Wirjadi 2007, Withey 2007, Sharma 2010].

2.2.3 Data rendering

The last phase of the volume rendering pipeline is rendering. Rendering methods can be divided among those that require a previous mapping to geometric primitives and those that render volume data directly considering the voxel model as the primitive.

The first set are known as *indirect volume rendering*¹ techniques. These strategies are classified according to the dimension of the graphical primitives into the following:

- **Point or particle mapping (0D)**

0D mapping uses points or particles as the primitive to render. Points and particles both encapsulate a 3D position and related information, although particles are more general because they can have any visual representation. Particles are created at the locations of features that have to be visualized. Then, they can be redistributed to cover the entire feature, and filtered to remove particles from hidden areas. Finally, geometry is created for the particles according to a desired visual style, and this geometry is sent to the rendering pipeline to be projected onto the final image. [Busking 2008, van Pelt 2010]

- **Contour mapping (1D)**

1D mapping extracts contour lines from the model. Object contours are rendered visualizing areas with locally high gradient magnitude, independently of the voxel value. This way, 3D structures become visible without being obstructed by visual representations of continuous regions in the volume. [Csébfalvi 2001, Isenberg 2006]

- **Surface mapping (2D)**

This mapping technique generates a polygonal approximation of an *isosurface*, i.e. the surface that approximates the volume points with a given value, known as *isovalue*. This approach is well suited to objects with sharply determined borders, like bones in CT, but it is inappropriate for amorphous objects that are difficult to represent by thin surfaces. The most popular isosurface extraction method is the marching cubes algorithm, proposed by Lorensen and Cline [Lorensen 1987, Newman 2006].

¹Indirect volume rendering is normally used to refer only to surface mapping, but here, for the sake of simplicity, we also include point and contour mapping into it.

The algorithms corresponding to each mapping technique are, respectively, point or particle rendering [Rusinkiewicz 2000, Csébfalvi 2003], contour rendering [Csébfalvi 2001], and surface rendering [Tiede 1990]. The most popular among these are surface rendering algorithms, that represent surfaces as triangle meshes. There are also other indirect volume rendering techniques that do not fit in any of the above categories, such as methods that operate on the frequency domain instead of the spatial one [Malzbender 1993].

The second set of visualization methods are known as *direct volume rendering* techniques. These were developed to capture the entire 3D data in a 2D image, contrary to the previous approaches that extract just a subset of the information. Since there is no mapping, the whole dataset is passed directly to the renderer. This is the approach that has been used in this thesis, so in the next section we will explain it in detail.

2.3 Direct volume rendering

Direct volume rendering (DVR) techniques consider volume data as a transparent gel and simulate the effect of light travelling through this gel. To carry out this process three main steps are required: shading, classification, and compositing. In this section, first we review these steps and then the major algorithms for direct volume rendering are explained.

2.3.1 Direct volume rendering steps

Direct volume rendering techniques perform the three steps illustrated in the last box of Figure 2.1 and described below.

2.3.1.1 Shading

Shading or illumination aims to enhance the appearance of a rendered object by modeling effects like shadows, light scattering, and absorption. The subset of these effects that is actually used depends on the chosen optical model [Max 1995]. There are several different models: from only absorption or emission, to both, to models that include scattering and advanced shadowing. Usually, a simple absorption and emission model is used because it is relatively cheap to evaluate and gives good enough results. In this model, it is considered that each sample of the data absorbs and emits light. The absorption is simulated by an opacity value assigned to the sample, while emission is determined by a color. Both color and opacity are defined in a classification process. The sampling rate determines the number of samples we are going to consider for each ray (see Figure 2.2).

Given the optical model, we still can distinguish between local illumination and global illumination. Local lighting shades each object—each sample in this case— independently, ignoring the rest of the scene, so it cannot produce shadows or scattering, but it has the advantage of being cheap to compute. Global lighting can simulate any effect and so gives better results, but it is expensive and seldom used in real time. Some

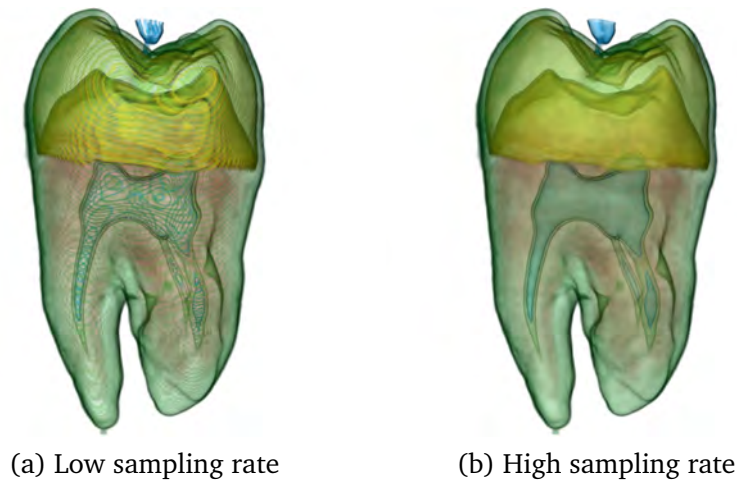


Figure 2.2: Comparison of the results obtained by rendering the same volume with (a) low and (b) high sampling rates.

shadowing effects of global illumination can be approximated with techniques such as ambient occlusion or obscurances, and then used to enhance local illumination. The application of obscurances to volume rendering is one of our main research foci. For this reason, a more thorough explanation will be given in Section 2.5.

2.3.1.2 Classification

As we have said, the absorption and emission optical model requires that each sample is given a color and an opacity. This is done in the classification step. Classification allows to extract important parts of the data by mapping primitives to graphical attributes such as color and opacity. This mapping is done with a *transfer function*, which in the simplest case takes as input the sample value and returns its corresponding color and opacity. It can be more complex, though: for example, the transfer function may take the gradient magnitude as an additional input to return an opacity, or it can have an additional output such as the shininess. The definition of the transfer function is a challenging task. Several methods have been developed to aid the user in this process, as it will be seen in Section 2.4.

Classification can be done before or after reconstruction, with different results. The former approach is known as pre-classification and consists of classifying voxel values and then interpolating the colours and the opacities. On the other hand, post-classification involves classifying sampled values gotten through interpolation. Choosing one alternative or the other will result in different image quality and artifacts, as it can be seen in Figure 2.3. Post-classification (Fig. 2.3(a)) produces images with defined edges, but is prone to aliasing because high frequency details may be missed. On the contrary, pre-classification (Fig. 2.3(b)) produces blurred images when the resolution of the image is higher than that of the volume and it is prone to color bleeding if colors and opacities are interpolated independently. However, this artifact is eliminated

if color interpolation is done with opacity-weighted colors, i.e., colors that have been multiplied by their corresponding opacity.

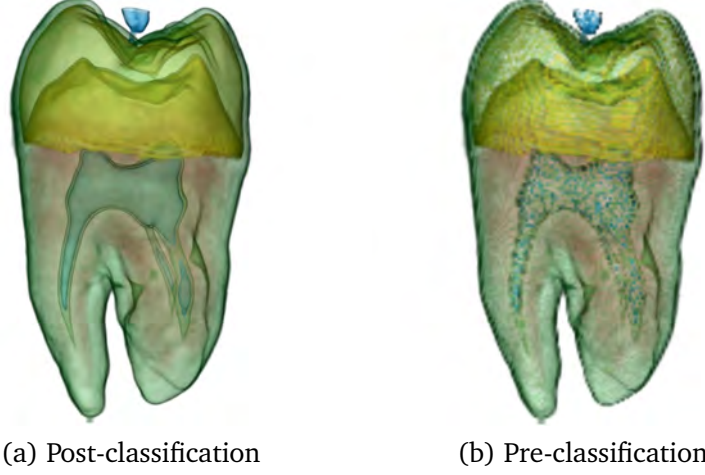


Figure 2.3: Comparison of the results obtained by rendering the same volume with (a) post-classification and (b) pre-classification.

2.3.1.3 Compositing

Compositing is the process by which samples with the optical properties given in classification and shading are integrated along viewing rays. This integration follows the volume rendering integral from light transport theory [Meißner 2000]. The theoretical form of the integral is

$$I_\lambda(x, r) = \int_0^L C_\lambda(s) \mu(s) e^{-\int_0^s \mu(t) dt} ds, \quad (2.2)$$

where $I_\lambda(x, r)$ is the amount of light of wavelength λ coming from ray direction r that is received at location x on the image plane, L is the length of ray r , $C_\lambda(s)$ is the light of wavelength λ reflected and/or emitted at location s in the direction of r , $\mu(s)$ is the density at s , and the exponential is an attenuation function.

Since the volume rendering integral cannot be computed analytically in general, it is usually approximated by a Riemann sum of the emitted and absorbed light in intervals i of width Δs :

$$I_\lambda(x, r) \approx \sum_{i=0}^{\frac{L}{\Delta s}} C_\lambda(s_i) \alpha(s_i) \cdot \prod_{j=0}^{i-1} (1 - \alpha(s_j)). \quad (2.3)$$

Here the density has been substituted by the opacity α and the exponential has been approximated by a Taylor series. The opacity has to be normalized for $\Delta s \neq 1$.

From this approximation we get the familiar compositing equations. In front-to-

back order these are:

$$C'_i = C'_{i-1} + (1 - A'_{i-1})C_i, \quad (2.4)$$

$$A'_i = A'_{i-1} + (1 - A'_{i-1})A_i. \quad (2.5)$$

The advantage with this approach is an optimization called early ray termination, which cuts a ray once the accumulated opacity (or remaining transparency) reaches a threshold where further contributions are negligible, i.e., 0.99 (or 0.01). In back-to-front order it is not necessary to keep track of accumulated opacity:

$$C'_i = C_i + (1 - A_i)C'_{i+1}. \quad (2.6)$$

This way the algorithm is simpler but we cannot do early ray termination.

There are other combining functions that can be used instead of compositing. One of the most popular is maximum intensity projection (MIP), which keeps the sample with the highest intensity value.

2.3.2 Direct volume rendering algorithms

Direct volume rendering can be achieved using different algorithms. The most popular are ray casting, splatting, shear-warp, and 3D texture mapping. These algorithms have been developed and refined over the years, beginning with simple software implementations (except for 3D texture mapping), and in recent years exploiting graphics hardware capabilities. All of them have a similar structure with the common steps of classification, shading, and compositing, but they differ in how each step is performed. In this section we give an overview of each of these algorithms.

2.3.2.1 Ray casting

Ray casting [Levoy 1988] is an image-order direct volume rendering algorithm, and it is the most popular. It is conceptually very simple. A ray is cast for each pixel into the volume. Samples of the volume are taken along the ray at equi-spaced intervals and mapped to optical properties according to the transfer function. Then samples are shaded and finally composited to obtain the pixel color.

Both pre- and post-classification are possible with ray casting. The most used reconstruction filters are nearest neighbour for fast visualization and trilinear interpolation for high quality. Both front-to-back and back-to-front compositing are possible, although the former is the most usual to take advantage of early ray termination.

Refinements include jittering in the sample positions to avoid patterned sampling artifacts, and space-leaping to quickly traverse empty regions.

Using current graphics cards it is possible to achieve very fast implementations of the ray casting algorithm. The basic idea is to store the volume in a single 3D texture and use the fragment shader to cast rays into the volume. Each pixel corresponds to a single ray. The direction of each ray can be computed from the camera position and

the screen space coordinates of the pixel, or be obtained via rasterization, by computing the distance between pixels of the rasterized front and back faces of the volume bounding box. There is extensive literature on GPU-based volume ray casting [Krüger 2003, Hadwiger 2009, Pavlík 2009].

2.3.2.2 Splatting

Splatting is an object-order direct volume rendering algorithm first proposed by Westover [Westover 1992]. It represents the volume as an array of overlapping basis functions, commonly Gaussian kernels with amplitudes scaled by the voxel values. These basis functions generate footprints that spread the energy of voxels over multiple pixels. The basis functions are projected to the screen and composited to generate the image.

Both pre- and post-classification are possible with splatting, but compositing can only be performed in back-to-front order.

To improve the performance of this technique, footprints can be precomputed and stored in a lookup table. In addition, only voxels relevant to the image must be projected. Splatting can use a concept similar to early ray termination: early splat elimination, based on a dynamically computed screen occlusion map, that conservatively culls invisible splats early from the rendering pipeline [Mueller 1999].

Splatting can also be GPU-accelerated using several techniques [Neophytou 2005, Neophytou 2006].

2.3.2.3 Shear-warp

Shear-warp was proposed by Lacroute and Levoy [Lacroute 1994]. This algorithm transforms the view and the volume in order to perform axis-aligned renderings so it can take advantage of memory alignment and fixed scaling and blending factors. The method starts by transforming the viewing transformation so that slices of the volume become axis-aligned with an off-screen image buffer, where rendering is performed. To compensate for this transformation, the volume is sheared so that the same parts of it are viewed. Then, rendering is performed using a ray casting-like scheme, where rays are perpendicular to the slices and sample values are reconstructed from the values in the slice. After compositing, a warping step transforms the off-screen image so it has the desired orientation and perspective.

Both pre- and post-classification are possible, but the former allows an optimization. Sampled values are usually reconstructed using bilinear interpolation. The sampling rate is fixed given the view.

Improvements of the basic method include a pre-processing step where voxel runs are run-length-encoded based on pre-classified opacities. This requires the construction of a separate encoded volume for each of the three major viewing directions, incurring in a higher memory overhead than other methods, but allows to skip opaque image regions and transparent voxels.

2.3.2.4 3D texture mapping

The use of 3D texture mapping was popularized by Cabral et al. [Cabral 1994] for non-shaded volume rendering. The volume is usually pre-classified and then uploaded as a 3D texture to the graphics card, and then the GPU rasterizes polygonal slices through the volume parallel to the viewpoint. The slices are blended in back-to-front order to form the final image.

3D texture mapping usually employs pre-classification, although post-classification is also possible with multi-pass variants. Both nearest neighbour and trilinear interpolation are possible.

With this algorithm all the voxels are always rendered, it is not possible to skip unseen voxels. Another drawback is that if the volume is too large it has to be cut in subvolumes that have to be swapped in and out of texture memory in order to render them all.

In sections 2.2 and 2.3 we have made a broad introduction to volume rendering. In the following sections we will focus on the specific problems that we will take on.

2.4 Transfer function definition

A crucial step in volume rendering is the transfer function definition. This function assigns optical properties, such as color and opacity, to the data being visualized, determining which structures of the volume will be visible and how they will be rendered. A good transfer function reveals the important structures in the data without obscuring them with unimportant regions. The simplest approach to find a good transfer function is by trial and error, i.e., colors and opacities are manually entered and then modified if the obtained visualization is not the desired one. Obviously, this strategy is tedious and time-consuming. Therefore, several techniques have been proposed to aid the user in transfer function design. A significant part of these works involve special user interfaces to assist the user in editing the transfer functions or to define the parameters for the algorithms that generate these transfer functions. In this section, we review some of these strategies.

Pfister et al. [Pfister 2001] classified transfer functions into two different categories: data-centric or image-centric.

Data-centric transfer functions define visual properties based on volume data values and their derived attributes, such as the gradient magnitude [Levoy 1988], first and second order gradient-aligned derivatives [Kindlmann 1998], or curvature measures [Hladůvka 2000, Kindlmann 2003]. Levoy [Levoy 1988] proposed the use of the gradient magnitude to identify surfaces in volume data. Kindlmann and Durkin [Kindlmann 1998] used the first and second derivatives along the gradient direction to calculate a boundary emphasis to be included in the opacity transfer function. In addition to the design of the opacity transfer function, general multi-dimensional transfer functions were studied to better convey the boundaries and features in volume

data [Kindlmann 2003, Kniss 2002a, Kniss 2003, Lum 2004]. These methods create two-dimensional histograms where each entry represents the number of voxels at a given feature space pair at which the user in a trial-and-error manner assigns color and opacity until the desired visualization is obtained. To avoid this trial-and-error process, Maciejewski et al. [Maciejewski 2009] proposed the addition of non-parametric clustering within the transfer function feature space in order to extract patterns and guide transfer function generation. A special class of multidimensional transfer functions, called distance-based, consider distance as a second data dimension [Kanda 2002]. Roettger et al. [Roettger 2005] introduced spatialized transfer functions, a special variant of local transfer functions where connected components are identified and the positional information is mapped to color. In this way, different objects with the same values can be isolated. Lundström et al. [Lundström 2006] introduced local histograms to detect and identify materials with similar intensities. Šereda et al. [Šereda 2006b] proposed an extension of the local histograms capable of detecting the materials that form the boundaries of the objects.

As an alternative, image-centric transfer functions are designed considering parameters that can be derived from the rendered images. He et al. [He 1996] treated the transfer function specification as a parameter optimization problem and addressed it with stochastic search techniques. Marks et al. [Marks 1997] introduced design galleries as a general approach for selecting visualization parameters in a multidimensional space. Transfer function specification with this approach is accomplished by selecting previews from a randomized selection to guide the search process. König and Gröller [König 2001] introduced a user interface paradigm with a set of specification tools assisted with real time rendering to aid the user in the selection of the transfer function. Wu and Qu [Wu 2007] proposed a method that uses editing operations and stochastic search of transfer function parameters to maximize the similarity between rendered images given by the user. In general, image-centric methods are more goal-oriented and require less user interaction.

Correa and Ma [Correa 2009a] presented a method for classifying volume data based on the ambient occlusion of voxels. They detected occlusion patterns that reveal the spatial structure of materials or features of a volume and represented them in an occlusion spectrum. This occlusion spectrum leads to better two-dimensional transfer functions that can help classify complex data sets in terms of the spatial relationships among features. Correa and Ma [Correa 2009b] also proposed to use the visibility to guide the transfer function design. They introduced the notion of visibility histogram, which represents the contribution of each sample in the final resulting image, as an interactive aid for generating effective transfer functions. Later, Correa and Ma [Correa 2011] also generalized the notion of visibility histogram along a number of dimensions and proposed a semiautomated method for generating transfer functions, which progressively explores the transfer function space towards the goal of maximizing the visibility of important structures. A main limitation of reported techniques is that they require user interaction. Automatic transfer function specification is still a challenge and few methods support it. Salama et al. [Salama 2006] introduced a high level semantic model with a simple user interface that allows visualization experts to design

transfer function models for specific application areas, which can then be used intuitively by non-expert users. Šereda et al. [Šereda 2006a] proposed hierarchical clustering of material boundaries for automating the transfer function design. Zhou and Takatsuka [Zhou 2009] presented an approach for automating transfer function generation by utilizing topological attributes derived from the contour tree of a volume that acts as a visual index to volume segments. Wang et al. [Wang 2010] presented an interactive transfer function design tool based on ellipsoidal Gaussian transfer functions. These techniques generally require a previous segmentation or classification of the volume data set to automate the process.

In Chapter 8 we propose a new technique to deal with the problem of the transfer function definition. Our approach is semiautomatic and based on information theory.

2.5 Illumination models and obscurances

The absorption and emission optical model has become the most used in volume rendering. Two main strategies can be considered depending on the data used to approximate the model: local illumination and global illumination (see Figure 2.4).

Local illumination (Fig. 2.4(a)) computes the light arriving at each object independently, without taking into account the other objects. This kind of illumination gives acceptable results and is cheap to compute, so it is suitable for real time applications.

Global illumination (Fig. 2.4(b)), on the other hand, computes light interactions between all the objects in the scene, being able to simulate any lighting effect. In the absorption and emission model it takes into account occlusions to produce more realistic shadings. However, having to compute interactions between all the objects incurs a heavy penalty in computation time. So, although global illumination gives very good results and is a well-known technique for producing realistic scenes, its high computational cost makes it inappropriate for real time rendering.

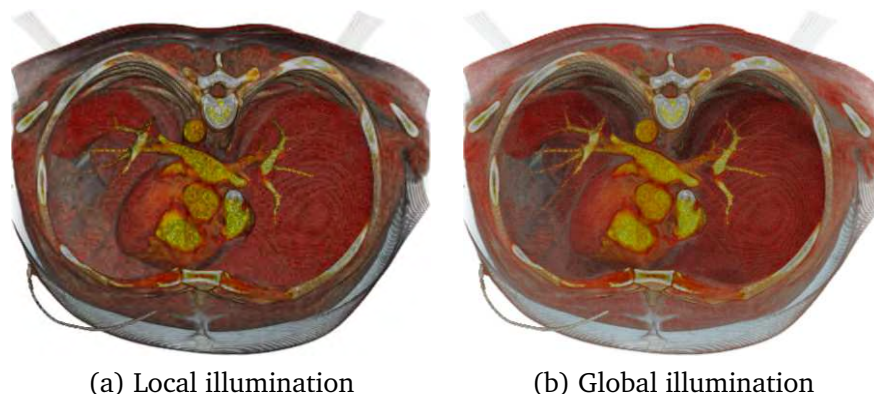


Figure 2.4: Comparison of the results obtained by rendering the same volume with (a) local illumination and (b) global illumination.

2.5.1 Volume shadowing

Different strategies have been proposed to simulate global illumination effects while preserving interactive frame rates. Behrens and Ratering [Behrens 1998] integrated in a texture-based volume renderer shadow maps that store the intensities computed according to light and transfer function conditions. Shadows can also be computed with half-angle slicing, either by using simultaneous slicing for rendering and shadow computation [Kniss 2002b], or combined with splatting [Zhang 2003]. Mavridis et al. [Mavridis 2010] used voxelization of polygonal scenes to compute illumination at discrete locations and apply it back to the polygons. The illumination is computed by iteratively propagating the radiance of each voxel to its neighbours. Kronander et al. [Kronander 2011] developed a method to achieve real time dynamic lighting in direct volume rendering by encoding local and global volumetric visibility using spherical harmonic basis functions stored in a multiresolution grid. Their technique allows directional lights, point lights, and environment maps.

Other strategies are based on the ambient occlusion technique introduced by Landis [Landis 2002], a simplified version of the obscurances illumination model [Zhukov 1998]. A volumetric version of the technique, called vicinity shading, was proposed by Stewart [Stewart 2003]. Vicinity shading simulates illumination of isosurfaces by taking into account neighboring voxels. An occlusion volume is computed and stored in a shading texture that is accessed during rendering. This volume has to be re-computed each time that the rendering parameters are modified and the method does not support color bleeding. Tarini et al. [Tarini 2006] refined this model to increase the performance. Wyman et al. [Wyman 2006] presented a method that supports the simulation of direct lighting, shadows and interreflections by storing pre-computed global illumination in an additional volume to allow viewpoint, lighting and isovalue changes. Despite the improvements achieved with these methods they still have a main limitation, they only allow to represent one of the surfaces of the volume.

This limitation is overcome by Ropinski et al. [Ropinski 2008] and Hernell et al. [Hernell 2007] using a local volumetric shadowing effect. Ropinski et al. compute a local histogram for each voxel from the voxel's neighbourhood, by accumulating intensities weighted by inverse squared distances. These local histograms can be combined interactively with the user-defined transfer function to give an effect similar to local ambient lighting. Hernell et al. [Hernell 2007] obtain the incident light intensity, arriving at a voxel, by integrating for each voxel and within a sphere surrounding it the attenuated transfer function density. This comes to compute, in the usual way, the visibility arriving at a voxel, using the opacities, averaged for all directions.

Desgranges and Engel [Desgranges 2007] proposed a method which combines ambient occlusion volumes terms from a plurality of different filtered volumes. Recomputation is also required whenever the transfer function is changed.

Jainek et al. [Jainek 2008] combined ambient occlusion calculations with illustrative display styles to enhance the clarity of the visual output in brain studies.

Šoltészová et al. [Šoltészová 2010] proposed a technique that simulates directional light scattering, based on the directional occlusion shading model. Their method does

not need any pre-computation and allows interactive modification of all illumination and rendering parameters.

In this thesis we have focused our interest on obscurances and ambient occlusion, therefore we describe them in detail below.

2.5.2 Obscurances and ambient occlusion

Zhukov et al. introduced ambient occlusion with the term obscurances [Zhukov 1998, Iones 2003] as an efficient technique that gives perceptually similar results to global illumination with a small fraction of the computational cost. Roughly speaking, obscurance measures the part of the hemisphere obscured by the neighboring surfaces. For instance, a corner of a room is more obscured than the center. From the physics of light transport, obscurance expresses the lack of secondary (reflected) light rays coming to the specific parts of the scene, thus making them darker. Computation was done as a preprocess and the obscurance values were used as an ambient term during rendering. Since the obscurance computation was a property of the geometry and not of the lighting conditions, results could be combined with an arbitrary direct illumination. The method was also useful for interactive applications because the results were independent from the viewpoint. Landis detailed how ambient occlusion could be used to add realism to models [Landis 2002] and Méndez et al. extended the use of obscurances to ray tracing [Méndez 2003b]. For a survey see [Méndez 2009].

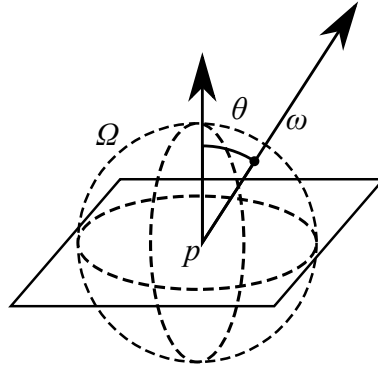


Figure 2.5: Illustration of the main parameters involved in the obscurances computation.

The obscurance O of a point p is defined as the integral

$$O(p) = \frac{1}{\pi} \int_{\Omega} \rho(d(p, \omega)) \cos \theta \, d\omega, \quad (2.7)$$

where ρ is a function of the distance $d(p, \omega)$ of the first intersection of a ray shot from point p with direction ω , p is a surface point, θ is the angle between the normal vector at p and direction ω , and the integration is over the hemisphere oriented according to the surface normal. Some of the variables are depicted in Figure 2.5. We only consider a neighborhood of p , i.e. function ρ is set to 1 for distances greater than a maximum

distance d_{\max} . Therefore, the integral function $O(p)$ captures occlusion (or openness) information of the environment of point p . Considering the extreme cases, an obscurance value of 1 means that the point is completely open, i.e. not occluded, and a value of 0 means that it is completely occluded.

Ambient occlusion [Landis 2002] is a simplified version of the obscurances illumination model. Ambient occlusion

$$A(p) = \frac{1}{\pi} \int_{\Omega} V(p, \omega) \cos \theta \, d\omega, \quad (2.8)$$

substitutes the ρ function in the obscurances equation (2.7) by the visibility function $V(p, \omega)$ that has value 0 when no geometry is visible in direction ω and 1 otherwise.

Extensive background on obscurances and ambient occlusion will be provided in Chapter 5.

2.6 Volume illustration

The concept of volume illustration was introduced by Rheingans and Ebert [Rheingans 2001], combining the familiarity of a physics-approximated illumination model with the ability to enhance important features using non-photorealistic rendering techniques. They propose a powerful unified framework for producing a wide range of illustration styles using local and global properties of the volume model to control opacity accumulation and illumination enhancing the perception of structure, shape, orientation, and depth relationships in a volume model.

The most popular styles, such as stippling (Fig. 2.6), hatching, and silhouettes, are from the pen-and-ink family. Csébfalvi et al. [Csébfalvi 2001] present a technique to visualize object contours, characterized by locally high gradient values. Lu et al. [Lu 2002] developed an interactive direct volume illustration system that simulates traditional stipple drawing.

Hauser et al. [Hauser 2001] proposed the two-level volume rendering concepts which allows focus+context visualization of volume data. Different rendering styles, such as direct volume rendering and maximum intensity projection, are used to emphasize objects of interest while still displaying the remaining data as context. Viola et al. [Viola 2004] introduced importance-driven volume rendering, where features within the volumetric data are classified according to object importance. Bruckner et al. [Bruckner 2006] presented context-preserving volume rendering, where the opacity of a sample is modulated by a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity.

Kindlmann et al. [Kindlmann 2003] utilized curvature-based transfer functions to incorporate illustrative effects in a volume renderer.

Cut-aways, exploded views, and high-level abstraction strategies, are also illustrative techniques used to reveal insights and represent essential structures of the volume in a clear way while less important details are subjugated [Correa 2006]. Clipping away or removing away parts of the data to eliminate occlusion is a well-known and exten-

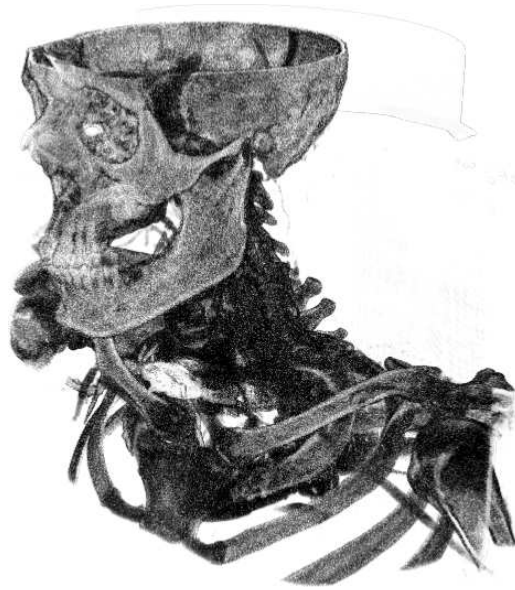


Figure 2.6: Example image of a volume rendered with a stipple effect.

sively used approach. The loss of context due to removed parts is the main limiting factor of such a technique. To overcome this limitation, strategies with more complex clipping geometry have been proposed. Wang and Kaufman [Wang 1995] introduced volume sculpting as a flexible approach to explore data. Weiskopf et al. [Weiskopf 2003] proposed several interactive clipping techniques that are capable of using complex clip geometries. Konrad-Verse et al. [Konrad-Verse 2004] described a method which is based on a deformable cutting plane for virtual resection. Viola et al. [Viola 2005] presented an importance-driven approach capable of enhancing important features while preserving the necessary context by generating cut-away views and ghosted images from volumetric data. Bruckner et al. [Bruckner 2006] proposed an alternative to conventional clipping techniques in order to avoid loss of context. Their context-preserving volume rendering model uses a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity to selectively reduce the opacity in less important data regions.

Exploded views and deformations are a common strategy for communicating the structure of complex 3D objects that are composed of many subparts. Deformation metaphors for browsing structures in volumetric data were introduced in volume visualization by McGuffin et al. [McGuffin 2003]. They presented an approach for volume exploration based on deformations that allows the users to cut into and open up, spread apart, or peel-away layers of the volume while still retaining the surrounding context. The explosion of the parts is set manually. Bruckner et al. [Bruckner 2006] went one step further by automating the explosion. Their method uses a continuous degree-of-interest function to distinguish between focus and context and is capable of re-arranging the parts dynamically based on the viewpoint. In these techniques, a priori knowledge of the volume data to define the layers or to set the focus of interest is

assumed — the data has been explicitly partitioned by the user.

In this thesis we apply illustrative effects in many occasions. In Chapter 3 we use voxel mutual information to produce several illustrative effects, including cool-and-warm and opacity modulation, and combine them with other effects not dependent on voxel mutual information, such as contours and color quantization. In Chapter 4 we explain an automatic method to automatically partition the volume based on characteristics of the data, in order to create exploded views. In Chapter 5 we define saliency as the gradient of obscurances, and then use this saliency to modulate the opacity of the volume. In Chapter 7 we use a spatial importance map obtained with a filtering method to modulate both the opacity of the volume and the density of points in stipple rendering.

2.7 Information theory tools

In 1948, Claude Shannon published a paper entitled “A mathematical theory of communication” [Shannon 1948] which marks the beginning of information theory. In this paper, Shannon defined measures such as entropy and mutual information², and introduced the fundamental laws of data compression and transmission. Information theory deals with the transmission, storage, and processing of information and is used in fields such as physics, computer science, mathematics, statistics, economics, biology, linguistics, neurology, learning, image processing, and computer graphics.

In information theory, *information* is simply the outcome of a selection among a finite number of possibilities and an information source is modelled as a random variable or a random process. The classical measure of information, Shannon entropy, expresses the information content or the uncertainty of a single random variable. It is also a measure of the dispersion or diversity of a probability distribution of observed events. For two random variables, their mutual information is a measure of the dependence between them. Mutual information plays an important role in the study of a *communication channel*, a system in which the output depends probabilistically on its input [Cover 1991, Verdú 1998, Yeung 2008].

This section presents Shannon’s information measures (entropy, conditional entropy, and mutual information) and their most basic properties. The information bottleneck method is also introduced. Good references of information theory are the books by Cover and Thomas [Cover 1991], and Yeung [Yeung 2008].

2.7.1 Entropy

Let X be a discrete random variable with alphabet \mathcal{X} and probability distribution $\{p(x)\}$, where $p(x) = \Pr[X = x]$ and $x \in \mathcal{X}$. In this thesis, $\{p(x)\}$ will be also denoted by $p(X)$ or simply p . This notation will be extended to two or more random variables.

²In Shannon’s paper, the mutual information is called rate of transmission.

The entropy $H(X)$ of a discrete random variable X is defined by

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x), \quad (2.9)$$

where the summation is over the corresponding alphabet and the convention $0 \log 0 = 0$ is taken.

In this thesis, logarithms are taken in base 2 and, as a consequence, entropy is expressed in bits. The convention $0 \log 0 = 0$ is justified by continuity since $x \log x \rightarrow 0$ as $x \rightarrow 0$. The term $-\log p(x)$ represents the information content (or uncertainty) associated with the result x . Thus, the entropy gives us the average amount of information (or uncertainty) of a random variable. Note that the entropy depends only on the probabilities. We can use interchangeably the notation $H(X)$ or $H(p)$ for the entropy, where p stands for the probability distribution $p(X)$.

Some relevant properties [Shannon 1948] of the entropy are:

- $0 \leq H(X) \leq \log |\mathcal{X}|$.
 - $H(X) = 0$ if and only if all the probabilities except one are zero, this one having the unit value, i.e., when we are certain of the outcome.
 - $H(X) = \log |\mathcal{X}|$ when all the probabilities are equal, i.e., we have maximum uncertainty.
- If the probabilities are equalized, entropy increases.

The binary entropy (Fig. 2.7) of a random variable X with alphabet $\{x_1, x_2\}$ and probability distribution $\{p, 1 - p\}$ is given by

$$H(X) = -p \log p - (1 - p) \log(1 - p). \quad (2.10)$$

Note that the maximum entropy is $H(X) = 1$ bit when $p = \frac{1}{2}$.

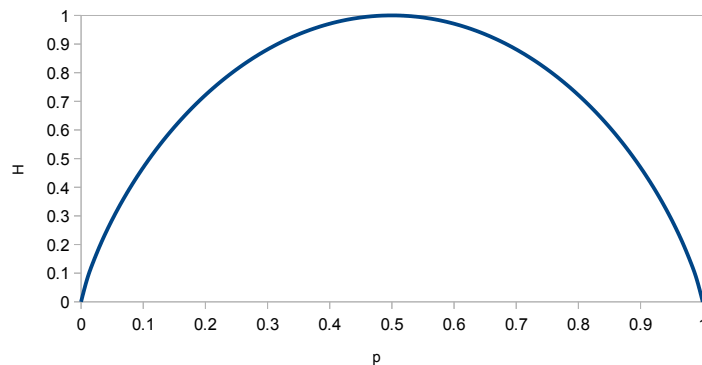


Figure 2.7: Plot of binary entropy.

The definition of entropy is now extended to a pair of random variables. The joint entropy $H(X, Y)$ of a pair of discrete random variables X and Y with a joint probability

distribution $p(X, Y) = \{p(x, y)\}$ is defined by

$$H(X, Y) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y), \quad (2.11)$$

where $p(x, y) = \Pr[X = x, Y = y]$ is the joint probability of x and y .

The conditional entropy $H(Y|X)$ of a random variable Y given a random variable X is defined as the expected value of the entropies of the conditional distributions:

$$\begin{aligned} H(Y|X) &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) = \sum_{x \in \mathcal{X}} p(x) \left(- \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x) \right) \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x), \end{aligned} \quad (2.12)$$

where $p(y|x) = \Pr[Y = y|X = x]$ is the conditional probability of y given x .

The Bayes theorem relates marginal probabilities $p(x)$ and $p(y)$, conditional probabilities $p(y|x)$ and $p(x|y)$, and joint probabilities $p(x, y)$:

$$p(x, y) = p(x)p(y|x) = p(y)p(x|y). \quad (2.13)$$

If X and Y are independent, then $p(x, y) = p(x)p(y)$. Marginal probabilities can be obtained from $p(x, y)$ by summation: $p(x) = \sum_{y \in \mathcal{Y}} p(x, y)$ and $p(y) = \sum_{x \in \mathcal{X}} p(x, y)$.

The conditional entropy can be thought of in terms of a communication or *information channel* $X \rightarrow Y$ whose output Y depends probabilistically on its input X . This information channel is characterized by a transition probability matrix which determines the conditional distribution of the output given the input [Cover 1991]. Hence, $H(Y|X)$ corresponds to the uncertainty in the channel output from the sender's point of view, and vice versa for $H(X|Y)$. Note that in general $H(Y|X) \neq H(X|Y)$. In this thesis, the conditional probability distribution of Y given x will be denoted by $p(Y|x)$ and the transition probability matrix (i.e., the matrix whose rows are given by $p(Y|x)$) will be denoted by $p(Y|X)$.

The following properties hold:

- $H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$.
- $H(X, Y) \leq H(X) + H(Y)$.
- $H(X) \geq H(X|Y) \geq 0$.
- If X and Y are independent, then $H(Y|X) = H(Y)$ since $p(y|x) = p(y)$ and, consequently, $H(X, Y) = H(X) + H(Y)$ (i.e., entropy is additive for independent random variables).

2.7.2 Kullback-Leibler divergence and mutual information

We now introduce two new measures, Kullback-Leibler divergence and mutual information, which quantify the distance between two probability distributions and the

shared information between two random variables, respectively.

The relative entropy or Kullback-Leibler divergence [Kullback 1951] $D_{\text{KL}}(p||q)$ between two probability distributions p and q , that are defined over the alphabet \mathcal{X} , is defined by

$$D_{\text{KL}}(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \quad (2.14)$$

The conventions that $0 \log \frac{0}{0} = 0$ and $a \log \frac{a}{0} = \infty$ if $a > 0$ are adopted. The Kullback-Leibler divergence satisfies the information inequality

$$D_{\text{KL}}(p||q) \geq 0, \quad (2.15)$$

with equality if and only if $p = q$. The Kullback-Leibler divergence is also called information divergence [Csiszár 2004] or informational divergence [Yeung 2008], and it is not strictly a metric³ since it is not symmetric and does not satisfy the triangle inequality. The Kullback-Leibler divergence is “a measure of the inefficiency of assuming that the distribution is q when the true distribution is p ” [Cover 1991].

The mutual information $I(X; Y)$ between two random variables X and Y is defined by

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log \frac{p(y|x)}{p(y)}. \end{aligned} \quad (2.16)$$

Mutual information represents the amount of information that one random variable, the input of the channel, contains about a second random variable, the output of the channel, and vice versa. That is, mutual information expresses how much the knowledge of Y decreases the uncertainty of X , and vice versa. $I(X; Y)$ is a measure of the shared information or dependence between X and Y . Thus, if X and Y are independent, then $I(X; Y) = 0$. Note that the mutual information can be expressed as the relative entropy between the joint distribution and the product of marginal distributions:

$$I(X; Y) = D_{\text{KL}}(p(X, Y)||p(X)p(Y)). \quad (2.17)$$

Mutual information $I(X; Y)$ fulfills the following properties:

- $I(X; Y) \geq 0$ with equality if and only if X and Y are independent
- $I(X; Y) = I(Y; X)$
- $I(X; Y) = H(X) + H(Y) - H(X, Y)$
- $I(X; Y) \leq \min\{H(X), H(Y)\}$

³A metric between x and y is defined as a function $d(x, y)$ that fulfills the following properties: (1) non-negativity: $d(x, y) \geq 0$, (2) identity: $d(x, y) = 0$ if and only if $x = y$, (3) symmetry: $d(x, y) = d(y, x)$, and (4) triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$.

- $I(X;X) = H(X)$

The relationship between Shannon's information measures can be expressed by a Venn diagram, as shown in Fig. 2.8⁴. The correspondence between Shannon's information measures and set theory is discussed in [Yeung 2008].

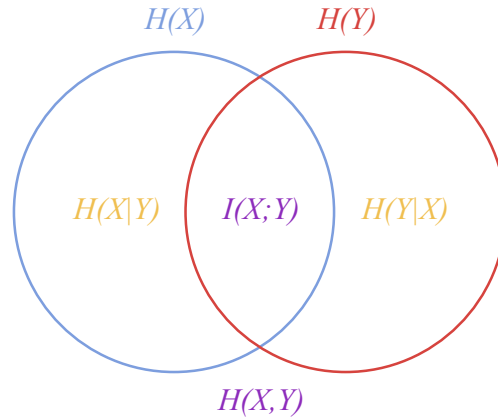


Figure 2.8: The information diagram represents the relationship between Shannon's information measures. Observe that $I(X;Y)$ and $H(X,Y)$ are represented, respectively, by the intersection and the union of the information in X (represented by $H(X)$) with the information in Y (represented by $H(Y)$). $H(X|Y)$ is represented by the difference between the information in X and the information in Y , and vice versa for $H(Y|X)$.

2.7.3 Inequalities

In this section, we introduce a group of inequalities that are essential in the study of information theory.

2.7.3.1 Jensen's inequality

In this section, we introduce the concepts of convexity and concavity. Many important inequalities and results in information theory are obtained from the concavity of the logarithmic function.

A function $f(x)$ is convex over an interval $[a, b]$ (the graph of the function lies below any chord) if for every $x_1, x_2 \in [a, b]$ and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2). \quad (2.18)$$

A function is strictly convex if equality holds only if $\lambda = 0$ or $\lambda = 1$.

A function $f(x)$ is concave (the graph of the function lies above any chord) if $-f(x)$ is convex.

For instance, x^2 and $x \log x$ (for $x > 0$) are strictly convex functions, and $\log x$ (for $x > 0$) is a strictly concave function.

⁴The information diagram does not include the universal set as in a usual Venn diagram.

Jensen's inequality can be expressed as follows. If f is a convex function on the interval $[a, b]$, then

$$\sum_{i=1}^n \lambda_i f(x_i) - f\left(\sum_{i=1}^n \lambda_i x_i\right) \geq 0, \quad (2.19)$$

where $0 \leq \lambda_i \leq 1$, $\sum_{i=1}^n \lambda_i = 1$, and $x_i \in [a, b]$. If f is a concave function, the inequality is reversed. A special case of this inequality is when $\lambda_i = \frac{1}{n}$ because then

$$\frac{1}{n} \sum_{i=1}^n f(x_i) - f\left(\frac{1}{n} \sum_{i=1}^n x_i\right) \geq 0, \quad (2.20)$$

that is, the value of the function at the mean of the x_i is less or equal than the mean of the values of the function at each x_i .

Jensen's inequality can also be expressed in the following way: if f is convex on the range of a random variable X , then

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)], \quad (2.21)$$

where \mathbb{E} denotes expectation (i.e., $\mathbb{E}[f(X)] = \sum_{x \in \mathcal{X}} p(x)f(x)$). Observe that if $f(x) = x^2$ (convex function), then $\mathbb{E}[X^2] - (\mathbb{E}[X])^2 \geq 0$. Thus, the variance is always positive.

2.7.3.2 Log-sum inequality

The log-sum inequality can be obtained from Jensen's inequality (Eq. (2.19)). For non-negative numbers a_1, a_2, \dots, a_n and b_1, b_2, \dots, b_n , the log-sum inequality is expressed as

$$\sum_{i=1}^n a_i \log \frac{a_i}{b_i} - \left(\sum_{i=1}^n a_i\right) \log \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \geq 0, \quad (2.22)$$

with equality if and only if $\frac{a_i}{b_i}$ is constant for all i . The conventions that $0 \log 0 = 0$, $0 \log \frac{0}{0} = 0$, and $a \log \frac{a}{0} = \infty$ if $a > 0$ are again adopted.

From this inequality, it can be proved that $H(X)$ is a concave function of p [Cover 1991].

From this inequality, the following properties can be proved [Cover 1991]:

- $D_{\text{KL}}(p||q)$ is convex in the pair (p, q) .
- $H(X)$ is a concave function of p .
- If X and Y have the joint distribution $p(x, y) = p(x)p(y|x)$, then $I(X; Y)$ is a concave function of $p(x)$ for fixed $p(y|x)$ and a convex function of $p(y|x)$ for fixed $p(x)$.

2.7.3.3 Jensen-Shannon inequality

The Jensen-Shannon divergence, derived from the concavity of entropy, is used to measure the dissimilarity between two probability distributions and has the impor-

tant feature that a different weight can be assigned to each probability distribution. The Jensen-Shannon (JS) divergence is defined by

$$JS(\pi_1, \pi_2, \dots, \pi_n; p_1, p_2, \dots, p_n) = H\left(\sum_{i=1}^n \pi_i p_i\right) - \sum_{i=1}^n \pi_i H(p_i), \quad (2.23)$$

where p_1, p_2, \dots, p_n are a set of probability distributions defined over the same alphabet with prior probabilities or weights $\pi_1, \pi_2, \dots, \pi_n$, fulfilling $\sum_{i=1}^n \pi_i = 1$, and $\sum_{i=1}^n \pi_i p_i$ is the probability distribution obtained from the weighted sum of the probability distributions p_1, p_2, \dots, p_n .

From the concavity of entropy (Section 2.7.3.2), the Jensen-Shannon inequality [Burbea 1982] is obtained:

$$JS(\pi_1, \pi_2, \dots, \pi_n; p_1, p_2, \dots, p_n) \geq 0. \quad (2.24)$$

The JS-divergence measures how far the probabilities p_i are from their mixing distribution $\sum_{i=1}^n \pi_i p_i$, and equals zero if and only if all the p_i are equal. It is important to note that the JS-divergence is identical to the mutual information $I(X; Y)$ when $\pi_i = p(x_i)$ (i.e., $\{\pi_i\}$ corresponds to the marginal distribution $p(X)$), $p_i = p(Y|x_i)$ for all $x_i \in \mathcal{X}$ (i.e., p_i corresponds to the conditional distribution of Y given x_i), and $n = |\mathcal{X}|$ [Burbea 1982, Slonim 2000].

2.7.3.4 Data processing inequality

The data processing inequality is expressed as follows. If $X \rightarrow Y \rightarrow Z$ is a Markov chain⁵, then

$$I(X; Y) \geq I(X; Z). \quad (2.25)$$

This result proves that no processing of Y , deterministic or random, can increase the information that Y contains about X . In particular, if $Z = f(Y)$, then $X \rightarrow Y \rightarrow f(Y)$ and, consequently, $I(X; Y) \geq I(X; f(Y))$ [Cover 1991].

2.7.4 Entropy rate

Using the property $H(X_1, X_2) = H(X_1) + H(X_2|X_1)$ (Sec. 2.7.1) and the induction on n [Yeung 2008], it can be proved that the joint entropy of a collection of n random variables X_1, \dots, X_n is given by

$$H(X_1, \dots, X_n) = \sum_{i=1}^n H(X_i | X_1, \dots, X_{i-1}). \quad (2.26)$$

We now introduce the entropy rate that quantifies how the entropy of a sequence of n random variables increases with n . The entropy rate or entropy density H_X of a

⁵For random variables X, Y , and $Z, X \rightarrow Y \rightarrow Z$ forms a Markov chain if $p(x, y, z) = p(x)p(y|x)p(z|y)$. That is, the probability of the future state depends on the current state only and is independent of what happened before the current state.

stochastic process⁶ $\{X_i\}$ is defined by

$$H_X = \lim_{n \rightarrow \infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \quad (2.27)$$

when the limit exists.

Entropy rate represents the average information content per symbol in a stochastic process. For a stationary stochastic process⁷, the entropy rate exists and is equal to

$$H_X = \lim_{n \rightarrow \infty} H_X(n), \quad (2.28)$$

where $H_X(n) = H(X_1, \dots, X_n) - H(X_1, \dots, X_{n-1}) = H(X_n | X_{n-1}, \dots, X_1)$. Entropy rate can be seen as the uncertainty associated with a given symbol if all the preceding symbols are known. It can also be interpreted as the irreducible randomness in sequences produced by an information source [Feldman 1998].

2.7.5 Entropy and coding

In this section, we present different interpretations of the Shannon entropy:

- As we have seen in Sec. 2.7.1, $-\log p(x)$ represents the information associated with the result x . The value $-\log p(x)$ can also be interpreted as the surprise associated with the outcome x . If $p(x)$ is small, the surprise is large; if $p(x)$ is large, the surprise is small. Thus, entropy (Eq. (2.9)) can be seen as the expectation value of the surprise [Feldman 2002].
- A fundamental result of information theory is the Shannon source coding theorem, which deals with the encoding of information in order to store or transmit it efficiently. This theorem can be formulated in the following ways [Cover 1991, Feldman 2002]:
 - Given a random variable X , $H(X)$ fulfills

$$H(X) \leq \bar{\ell} < H(X) + 1, \quad (2.29)$$

where $\bar{\ell}$ is the expected length of an optimal binary code for X . An example of an optimal binary code is the Huffman instantaneous coding⁸.

⁶A stochastic process or a discrete-time information source $\{X_i\}$ is an indexed sequence of random variables characterized by the joint probability distribution $p(x_1, x_2, \dots, x_n) = \Pr[(X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)]$ with $(x_1, x_2, \dots, x_n) \in \mathcal{X}^n$ for $n \geq 1$ [Cover 1991, Yeung 2008].

⁷A stochastic process $\{X_i\}$ is stationary if two subsets of the sequence, $\{X_1, X_2, \dots, X_n\}$ and $\{X_{1+l}, X_{2+l}, \dots, X_{n+l}\}$, have the same joint probability distribution for any $n, l \geq 1$: $\Pr[(X_1, \dots, X_n) = (x_1, x_2, \dots, x_n)] = \Pr[(X_{1+l}, X_{2+l}, \dots, X_{n+l}) = (x_1, x_2, \dots, x_n)]$. That is, the statistical properties of the process are invariant to a shift in time. At least, H_X exists for all stationary stochastic processes.

⁸A code is called a prefix or instantaneous code if no codeword is a prefix of any other codeword. Huffman coding uses a specific algorithm to obtain the representation for each symbol. The main characteristic of this code is that the most common symbols use shorter strings of bits than the ones used by the less common symbols.

- If we optimally encode n identically distributed random variables X with a binary code, the Shannon source coding theorem can be enunciated in the following way:

$$H(X) \leq \bar{\ell}_n < H(X) + \frac{1}{n}, \quad (2.30)$$

where $\bar{\ell}_n$ is the expected codeword length per unit symbol. Thus, by using large block lengths, we can achieve an expected codelength per symbol arbitrarily close to the entropy [Cover 1991].

- For a stationary stochastic process, we have

$$\frac{H(X_1, X_2, \dots, X_n)}{n} \leq \bar{\ell}_n < \frac{H(X_1, X_2, \dots, X_n)}{n} + 1 \quad (2.31)$$

and, from the definition of entropy rate H_X (Eq. (2.27)),

$$\lim_{n \rightarrow \infty} \bar{\ell}_n \rightarrow H_X. \quad (2.32)$$

Thus, the entropy rate is the expected number of bits per symbol required to describe the stochastic process.

2.7.6 Information bottleneck method

The information bottleneck method, introduced by Tishby et al. [Tishby 1999], is a technique that extracts a compact representation of the variable X , denoted by \hat{X} , with minimal loss of mutual information with respect to another variable Y (i.e., \hat{X} preserves as much information as possible about the control variable Y). Thus, given an information channel between X and Y , the information bottleneck method tries to find the optimal tradeoff between accuracy and compression of X when the bins of this variable are clustered.

Soft [Tishby 1999] and hard [Slonim 1999] partitions of X can be adopted. In the first case, every $x \in \mathcal{X}$ can be assigned to a cluster $\hat{x} \in \hat{\mathcal{X}}$ with some conditional probability $p(\hat{x}|x)$ (soft clustering). In the second case, every $x \in \mathcal{X}$ is assigned to only one cluster $\hat{x} \in \hat{\mathcal{X}}$ (hard clustering).

In this thesis, we consider hard partitions and we focus our attention on the agglomerative information bottleneck method [Slonim 1999]. Given a cluster \hat{x} defined by $\hat{x} = \{x_1, \dots, x_l\}$, where $x_k \in \mathcal{X}$ for all $k \in \{1, \dots, l\}$, and the probabilities $p(\hat{x})$ and $p(y|\hat{x})$ defined by

$$p(\hat{x}) = \sum_{k=1}^l p(x_k), \quad (2.33)$$

$$p(y|\hat{x}) = \frac{1}{p(\hat{x})} \sum_{k=1}^l p(x_k, y) \quad \forall y \in \mathcal{Y}, \quad (2.34)$$

the following properties are fulfilled:

- The decrease in the mutual information $I(X; Y)$ due to the merge of x_1, \dots, x_l is given by

$$\delta I_{\hat{x}} = p(\hat{x}) JS(\pi_1, \dots, \pi_l; p_1, \dots, p_l) \geq 0, \quad (2.35)$$

where the weights and probability distributions of the JS-divergence are given by $\pi_k = \frac{p(x_k)}{p(\hat{x})}$ and $p_k = p(Y|x_k)$ for all $k \in \{1, \dots, l\}$, respectively. An optimal clustering algorithm should minimize $\delta I_{\hat{x}}$.

- An optimal merge of l components can be obtained by $l - 1$ consecutive optimal merges of pairs of components.

2.8 Information theory in computer graphics

Two excellent surveys of the application of information theory to computer graphics are by Chen and Jänicke [Chen 2010], and by Wang and Shen [Wang 2011]. A summary of information theory tools for computer graphics is presented in [Sbert 2009].

In computer graphics, the most basic information-theoretic measures have been used in scene complexity [Feixas 1999], global illumination [Rigau 2003], light positioning [Gumhold 2002], and viewpoint selection for polygonal scenes [Vázquez 2001, Sbert 2005, Feixas 2009]. In the latter field, entropy [Vázquez 2001], Kullback-Leibler distance [Sbert 2005], and mutual information [Feixas 2009] have been applied to quantify the quality of a viewpoint. From an information channel between the set of viewpoints and the polygons of an object, all these measures can be presented in a unified framework, enabling to compute other aspects such as the similarity of two viewpoints, both the stability and the saliency of a viewpoint, and both the information and the saliency associated with a polygon [Feixas 2009, González 2008].

In visualization, information theory has been applied to fields such as view selection, flow visualization, time-varying volume visualization, and transfer function definition. Viewpoint entropy has been introduced by Bordoloi and Shen [Bordoloi 2005] and Takahashi et al. [Takahashi 2005] to select the best views in volume rendering. Bordoloi and Shen [Bordoloi 2005] also used the Jensen-Shannon divergence to compute the stability of a viewpoint and the conditional entropy for time varying volume data. Viola et al. [Viola 2006] introduced the mutual information between a set of viewpoints and a set of objects to calculate the representativeness of a viewpoint. Xu et al. [Xu 2010] used entropy to measure the information content in the local regions across a vector field and conditional entropy to evaluate the effectiveness of streamlines to represent the input vector field. Lee et al. [Lee 2011] used entropy for viewpoint selection and streamline filtering for flow visualization. Wang and Shen [Wang 2006] used entropy to validate the quality of each individual data block in a LOD and the relationships among them. In time-varying volume visualization, Ji and Shen [Ji 2006] applied entropy to dynamic view selection, and Wang et al. [Wang 2008] introduced the conditional entropy to quantify the information a data block contains with respect to other blocks in the time sequence. Haidacher et al. [Haidacher 2008] introduced the decomposition of mutual information for transfer function design in multimodal

volume visualization. They proposed a new 2D space for manually defining transfer functions. Bruckner and Möller [Bruckner 2010] introduced isosurface similarity maps to present structural information of a volume data set by depicting similarities between individual isosurfaces quantified by mutual information. The maps are used to guide the transfer function design and the visualization parameter specification.

2.9 Viewpoint selection and visibility channel in volume visualization

Automatic selection of the most informative viewpoints is a very useful focusing mechanism in visualization of scientific data, guiding the viewer to the most interesting information of the data set. Best view selection algorithms have been applied to computer graphics domains, such as scene understanding and virtual exploration [Vázquez 2003, Sokolov 2006], and volume visualization [Bordoloi 2005, Takahashi 2005, Chan 2006, Viola 2006].

As we have mentioned in the previous section, Shannon’s information measures, such as entropy and mutual information, have been used in computer graphics and scientific visualization to measure the quality of a viewpoint from which a given scene is rendered. Viewpoint entropy, first introduced in [Vázquez 2001] for polygonal models, has been applied to volume visualization in [Bordoloi 2005, Takahashi 2005]. In particular, Bordoloi and Shen [Bordoloi 2005] obtained the goodness of a viewpoint from the entropy of the visibility of the volume voxels. Viola et al. [Viola 2006] proposed a visibility channel and used the viewpoint mutual information to automatically determine the most expressive view on a selected focus. A unified information-theoretic framework for viewpoint selection, ambient occlusion, and mesh saliency for polygonal models has been presented in [González 2008, Feixas 2009]. Next we review the definitions of visibility channel, viewpoint mutual information, and viewpoint entropy in volume visualization.

To select the most representative or relevant views of a volume data set, a viewpoint quality measure, the *viewpoint mutual information*, was defined [Viola 2006] from an *information channel* $V \rightarrow Z$ between the random variables V (input) and Z (output), which represent, respectively, a set of viewpoints \mathcal{V} and the set of objects (or voxels) \mathcal{Z} of a volume data set (see Figure 2.9(a)). Viewpoints are indexed by v and voxels by z . The capital letters V and Z as arguments of $p()$ are used to denote probability distributions. For instance, while $p(v)$ denotes the probability of a single viewpoint v , $p(V)$ denotes the input probability distribution of the set of viewpoints.

The information channel $V \rightarrow Z$ is characterized by a probability transition matrix (or conditional probability distribution) which determines, given the input, the output probability distribution (see Figure 2.9(b)). The main elements of this channel are the following:

- The transition probability matrix $p(Z|V)$, where each conditional probability $p(z|v)$ is given by the quotient $\frac{vis(z|v)}{vis(v)}$, where $vis(z|v)$ is the visibility of voxel z

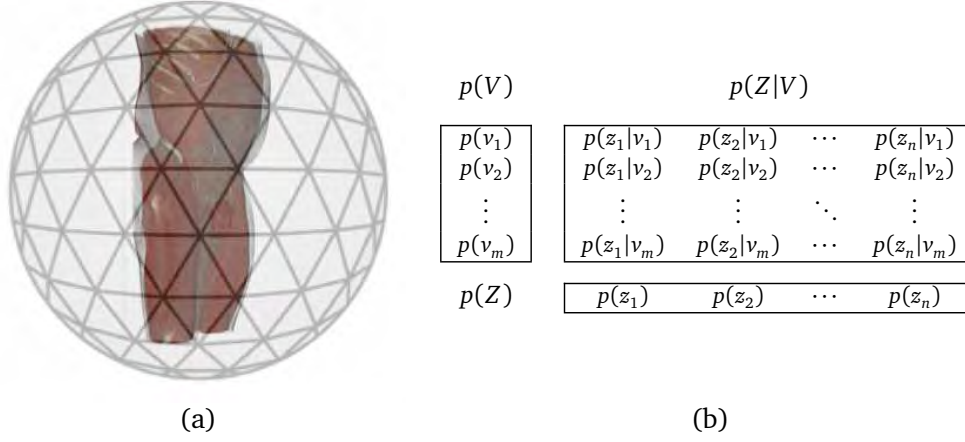


Figure 2.9: Visibility channel. (a) Sphere of viewpoints of a voxel model. (b) Probability distributions of channel $V \rightarrow Z$.

from viewpoint v and $vis(v) = \sum_{z \in \mathcal{Z}} vis(z|v)$ is the captured visibility of all voxels over the sphere of directions. The visibility $vis(z|v)$ of a voxel z is considered as the contribution of this voxel to the final image as rendered from viewpoint v . Conditional probabilities fulfill that $\sum_{z \in \mathcal{Z}} p(z|v) = 1$.

- The input probability distribution $p(V)$ is given by the probabilities of selecting each viewpoint, where an element $p(v)$ of this probability distribution can be interpreted as the importance of viewpoint v . In this paper, $p(V)$ is obtained from the normalization of the captured visibility of the data set over each viewpoint. Thus, $p(v) = \frac{vis(v)}{\sum_{v \in \mathcal{V}} vis(v)}$ expresses how much volume is visible from viewpoint v . In [Viola 2006], uniform importance was assigned to each viewpoint.
- From $p(V)$ and $p(Z|V)$, the output probability distribution $p(Z)$ is given by

$$p(z) = \sum_{v \in \mathcal{V}} p(v)p(z|v), \quad (2.36)$$

which expresses the average visibility of each voxel.

The degree of *dependence* or *correlation* between a set of viewpoints \mathcal{V} and the volume data set \mathcal{Z} , expressed by the *mutual information* (MI) between V and Z , is given by

$$I(V; Z) = \sum_{v \in \mathcal{V}} p(v) \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} = \sum_{v \in \mathcal{V}} p(v) I(v; Z), \quad (2.37)$$

where

$$I(v; Z) = \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} \quad (2.38)$$

is defined as the *viewpoint mutual information* (VMI), which measures the degree of dependence between the viewpoint v and the set of voxels. Note that $I(v; Z)$ is not

properly a mutual information, but the contribution of viewpoint v to mutual information $I(V; Z)$. In this framework, the most *representative* viewpoint is defined as the one that has minimum VMI. Low values correspond to more independent views, showing the maximum possible number of voxels in a balanced way. The term ‘balance’ is used here to express that the visibility distribution $p(Z|v)$ of v is similar to $p(Z)$. This similarity is expressed by the Kullback-Leibler distance between $p(Z|v)$ and $p(Z)$ (see [Feixas 2009, Viola 2006]). This distance is zero when $p(Z|v) = p(Z)$. On the other hand, high values of $I(v; Z)$ mean a high dependence between viewpoint v and the object, indicating a highly coupled view (for instance, between the viewpoint and a small number of voxels with low average visibility). In [Viola 2006], it has been shown that one of the main properties of VMI is its robustness to deal with any type of discretisation or resolution of the volume data set. The same behavior can be observed for polygonal data [Feixas 2009].

From the visibility channel, the *viewpoint entropy* (VE) of viewpoint v is defined by

$$H(Z|v) = - \sum_{z \in \mathcal{Z}} p(z|v) \log p(z|v). \quad (2.39)$$

VE measures the degree of uniformity of the visibility distribution $p(Z|v)$ at viewpoint v . The best viewpoint is defined as the one that has maximum VE, that would be obtained when a certain viewpoint can see all the voxels with the same projected visibility. On the other hand, minimum VE would be obtained when most of the visibility is captured from few voxels.

Visibility channel

Contents

3.1 Introduction	37
3.2 Background	38
3.2.1 Volume shadowing and illustrative techniques	38
3.2.2 Visibility channel	38
3.3 Voxel information	39
3.4 Illustrative visualization using voxel information	42
3.4.1 Ambient occlusion	42
3.4.2 Color ambient occlusion	47
3.4.3 Focus+context	47
3.5 Viewpoint selection using voxel information	50
3.6 Conclusions	51

3.1 Introduction

In the last decades many different strategies have been proposed to visualize and explore volume data sets efficiently. One of the main challenges is to obtain realistic or illustrative renderings that adapt the appearance of the data to the specific task satisfying user requirements. Moreover, the demand of interactivity when exploring volume data has led to the development of new strategies to accelerate the rendering process. In this context, focus+context and viewpoint-based strategies improve the exploration efficiency by directing the users to the most informative parts of the data. GPU-based implementations which exploit hardware capabilities have been also proposed.

In this chapter, we propose a volume visualization system based on the information channel defined between the voxels of a volume data set and a set of viewpoints. This channel is obtained from the reversion of the visibility channel defined in Viola et al. [Viola 2006] and explained in Section 2.9. Thus, instead of analyzing how a viewpoint sees the volume data set, we focus on how a voxel “sees” the viewpoints. The shared information of each voxel with the set of visible viewpoints is interpreted as a visibility quality descriptor of a voxel that provides a natural ambient occlusion value [Landis 2002, Zhukov 1998].

The proposed framework results in a flexible system for producing realistic and non-photorealistic renderings in an automatic way. The use of the voxel information

combined with the assignation of color to each viewpoint and non-photorealistic effects produces an enhanced visualization of the volume data set. Voxel information is also applied to modulate the transfer function in order to focus on or highlight the most informative parts of the data set. Finally, a new viewpoint selection measure based on voxel information is introduced and compared with other information-theoretic viewpoint measures. The proposed framework has been partially implemented using Compute Unified Device Architecture (CUDA)¹, allowing to exploit the capabilities of modern GPUs.

The chapter is organized as follows. Section 3.2 reminds of previous work related to volumetric shadowing, volume illustration, and viewpoint selection, that has been reviewed in detail in Chapter 2. Section 3.3 introduces an information channel which enables us to calculate the information associated to each voxel. Section 3.4 presents different visualization applications that can be derived from the voxel information. Section 3.5 defines a viewpoint quality measure based on voxel information. Finally, Section 3.6 presents the conclusions and future work.

3.2 Background

In this section, we give a brief reminder of the concepts from Chapter 2 that are relevant to this chapter.

3.2.1 Volume shadowing and illustrative techniques

Although the integration of global illumination effects in direct volume rendering enhances volume data interpretation, its high computational cost overcomes its application. In Section 2.5.1 we have reviewed several strategies that have been proposed to simulate these effects preserving interactive frame rates. The approaches of ambient occlusion by Landis [Landis 2002] and vicinity shading by Stewart [Stewart 2003] are especially relevant for comparison purposes in this chapter. We will also compare to our own obscurances method explained in Chapter 5.

Illustrative techniques are suitable for emphasizing certain features or properties while omitting or greatly simplifying less important details. Several techniques in this field are reviewed in Section 2.6. The most relevant to this chapter is the opacity modulation by Bruckner et al. [Bruckner 2006].

3.2.2 Visibility channel

Viola et al. [Viola 2006] defined an information channel $V \rightarrow Z$ between the random variables V (input) and Z (output), which represent, respectively, a set of viewpoints \mathcal{V} and the set of objects (or voxels) \mathcal{X} of a volume data set (see Section 2.9). From this channel, several information measures can be computed. In particular, mutual

¹www.nvidia.com/cuda

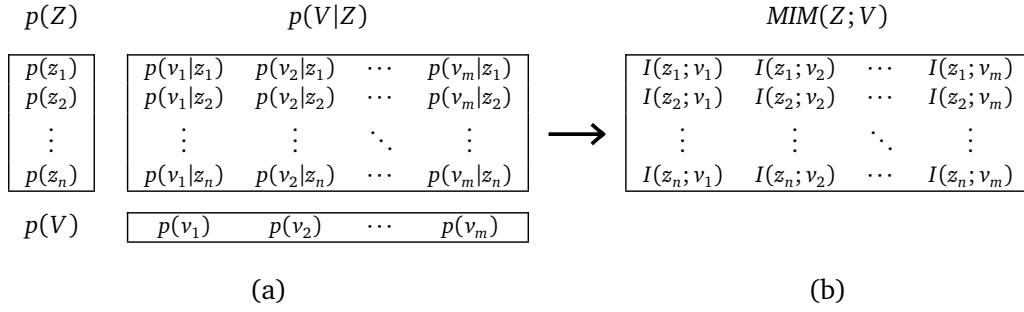


Figure 3.1: (a) Probability distributions of channel $Z \rightarrow V$, used to compute the voxel mutual information. (b) The elements of matrix $MIM(Z; V)$ are given by $I(z_i; v_j) = p(v_j|z_i) \log \frac{p(v_j|z_i)}{p(v_j)}$ and used to calculate the color ambient occlusion in Section 3.4.1.

information (MI) between V and Z ,

$$I(V; Z) = \sum_{v \in \mathcal{V}} p(v) \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} = \sum_{v \in \mathcal{V}} p(v) I(v; Z), \quad (3.1)$$

which expresses the degree of dependence or correlation between the set of viewpoints and the volume data set, viewpoint mutual information (VMI),

$$I(v; Z) = \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)}, \quad (3.2)$$

which measures the degree of dependence between the viewpoint v and the set of voxels, and viewpoint entropy (VE),

$$H(Z|v) = - \sum_{z \in \mathcal{Z}} p(z|v) \log p(z|v), \quad (3.3)$$

which measures the degree of uniformity of the visibility distribution $p(Z|v)$ at viewpoint v .

A complete definition of this visibility channel can be found in Section 2.9.

3.3 Voxel information

As we have seen in Section 2.9, the information associated with each viewpoint (VMI) is obtained from the definition of the channel between the sphere of viewpoints and the voxels (or objects) of the volume data set. In this section, the voxel information is defined from the reversed channel $Z \rightarrow V$, so that Z is now the input and V the output. The probability distributions of this channel are shown in Fig. 3.1(a). Note that MI is invariant to the reversion of the channel: $I(V; Z) = I(Z; V)$. The idea of reversing the channel was introduced in [González 2008] for polygonal models, together with the computation of the information associated to a polygon.

From the Bayes theorem $p(v, z) = p(v)p(z|v) = p(z)p(v|z)$, MI (see Eq. (3.1)) can be rewritten as

$$I(Z; V) = \sum_{z \in \mathcal{Z}} p(z) \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)} = \sum_{z \in \mathcal{Z}} p(z) I(z; V), \quad (3.4)$$

where

$$I(z; V) = \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)} \quad (3.5)$$

is the contribution of voxel z to $I(Z; V)$ and is defined as the *voxel mutual information* (VOMI). This represents the degree of correlation between the voxel z and the set of viewpoints, and can be interpreted as the information associated with voxel z . Analogous to VMI, low values of VOMI can correspond to voxels seen by a large number of viewpoints in a balanced way. That is, the lowest values of VOMI correspond to the voxels with conditional probability distribution $p(V|z)$ similar to $p(V)$. The opposite happens for high values.

Fig. 3.2 shows for different data sets the VOMI maps computed using 42 viewpoints and colored using the thermal scale represented in Fig. 3.2(e). Warm colors correspond to high VOMI values and cool colors to low ones.

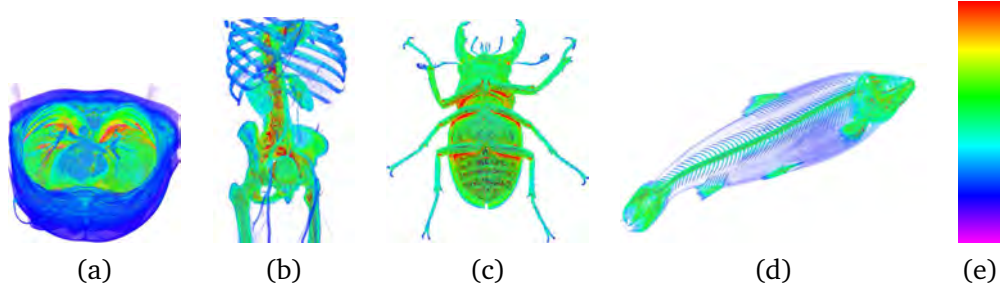


Figure 3.2: VOMI maps generated using 42 viewpoints for different models and transfer functions: (a) CT-body, (b) CT-body (skeleton), (c) CT-beetle, and (d) CT-salmon. Maps are colored using the thermal scale in (e).

The great potential of VOMI is that it allows varied interpretations that can be used in different visualization applications, such as volume illustration and viewpoint selection. Both will be discussed in detail in Sections 3.4 and 3.5, respectively.

In Fig. 3.3 we represent the different steps required for computing the VOMI of a voxel model. The process starts classifying the volume data by defining a transfer function. Then, a ray casting is performed considering the volume data set centered in a sphere of viewpoints and the camera looking at the center of this sphere. For each viewpoint a histogram of visibilities is created and then used to estimate $p(Z|v)$. Using Eq. (2.36) and the Bayes theorem, $p(Z)$ and $p(V|Z)$ can be obtained from both $p(V)$ and $p(Z|V)$. Finally, the VOMI map is obtained.

Since $p(Z|V)$ is a very huge matrix it cannot be stored in memory and we have to compute its rows, $p(Z|v)$, every time we need them. Thus, the VOMI computation

requires executing the ray casting stage three times, to compute $p(V)$, then $p(Z)$, and then the VOMI map.

We have implemented the most costly computations of our framework (ray casting, computation of $p(Z)$, and VOMI) using CUDA in order to speed up the process compared to a pure CPU implementation.

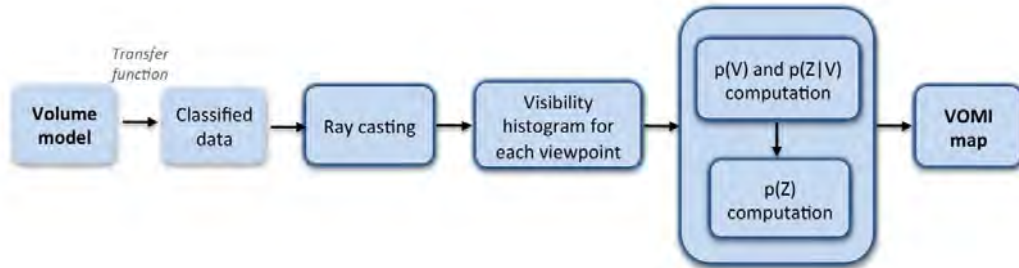


Figure 3.3: Overview of the VOMI pipeline.

The first and most complex part done in CUDA is the ray casting needed to compute $p(Z|v)$ for a viewpoint v . In this ray casting we need to fill a table with the visibility of each voxel from a given viewpoint, from which $p(Z|v)$ can be obtained. This visibility is a real value equal to the contribution of the voxel to the final image according to its opacity and also to the opacity of the preceding voxels in each ray that visits it [Levoy 1988]. For example, a voxel that is seen from one ray, that is fully opaque, and that is not occluded at all by any other voxel in this ray, has a visibility of 1. To update the visibility table, we need atomic operations in order to avoid race conditions (the same voxel may be visited by two or more neighboring rays), but unfortunately CUDA doesn't support atomic operations with floating point values. To overcome this limitation we multiply the visibility by a big constant (we have used 1 million) and truncate the result to obtain an integer that is atomically added to the visibility table; this is equivalent to working with fixed point precision. At the end of the process, we divide each value by the same constant to get real values. So, we have two kernels for this whole task: the first one does the ray casting and fills the integer table, and the second converts this table to a table of floating point numbers.

The other two parts that we have implemented in CUDA are fairly simple. One of them is the computation of the voxel probabilities $p(Z)$ (see Eq. (2.36)), with a kernel that just accumulates $p(v)p(Z|v)$ for a given viewpoint v . The other part is the computation of VOMI, where the kernel accumulates the term $p(v|z) \log \frac{p(v|z)}{p(v)}$. Remember that $p(v|z) = p(v) \frac{p(z|v)}{p(z)}$, for each voxel z and viewpoint v .

Table 3.1 shows the times to compute the VOMI map for different data sets and transfer functions with 42 viewpoints. We also report the individual times to do the ray casting and fill the visibility histogram, accumulate $p(Z)$ and accumulate VOMI, each for 1 viewpoint, because these are the processing bottlenecks that we have implemented in CUDA. The total time is the result of adding the ray casting 3×42 times, plus $p(Z)$ 42 times, plus VOMI 42 times, plus additional costs (CPU, synchronizations,

memory transfers, etc.).

	Ray casting (1 v)	$p(Z)$ (1 v)	VOMI (1 v)	Total (42 v)
Body	141.99 ms	13.09 ms	14.02 ms	29.716 s
Skeleton	108.09 ms	13.09 ms	13.41 ms	26.262 s
Beetle	158.93 ms	21.54 ms	19.92 ms	40.184 s
Salmon	114.65 ms	14.96 ms	14.78 ms	28.416 s

Table 3.1: Times to compute the VOMI maps for the data sets shown in Fig. 3.2. First and second rows: body and skeleton ($256 \times 256 \times 415$); third row: beetle ($416 \times 416 \times 247$); fourth row: salmon ($336 \times 173 \times 511$). First column: mean time to do a ray casting and compute the visibility histogram for 1 viewpoint; second column: time to accumulate $p(Z)$ for 1 viewpoint; third column: time to accumulate VOMI for 1 viewpoint; fourth column: total time to compute the VOMI map for 42 viewpoints.

3.4 Illustrative visualization using voxel information

In this section we describe how to achieve interactive visualization of realistic and non-photorealistic styles based on VOMI.

3.4.1 Ambient occlusion

A first application of the VOMI is by interpreting it as an ambient occlusion (AO) term (see Sec. 2.5.2). AO is a measure of the visibility around a voxel, but while classical AO takes into account only local visibility, VOMI considers the whole volumetric data visibility around a voxel, from viewpoints surrounding the volume. VOMI measures how this visibility is distributed between viewpoints. Thus, the more uniform the visibility the less important is which viewpoint we consider, meaning that the voxel is less interesting or informative. In that case the VOMI value is low. On the other hand, the less uniform the visibility, the more important is which viewpoint we consider, meaning that the voxel is more interesting or informative. In that case the VOMI value is high.

To obtain the AO of each voxel, the VOMI of all voxels has been normalized between 0 and 1 and subtracted from 1, because low values of VOMI, represented in the grey map by values near 1, correspond to non-occluded or visible (from many viewpoints) voxels, while high values of VOMI, represented in the grey map by values near 0, correspond to highly occluded voxels.

In Fig. 3.4 we show the results of computing AO using different numbers of viewpoints. Note that a sphere of 42 viewpoints provides enough quality for the AO maps, although 162 viewpoints is preferable for better quality.

In Fig. 3.5, we compare the AO maps corresponding to the models of column (a) generated using different strategies. From column (b) to (e), respectively, we present the approaches by Landis [Landis 2002], Stewart [Stewart 2003], our obscurances (Chap. 5), and finally our current proposed technique. Landis' approach is obtained by

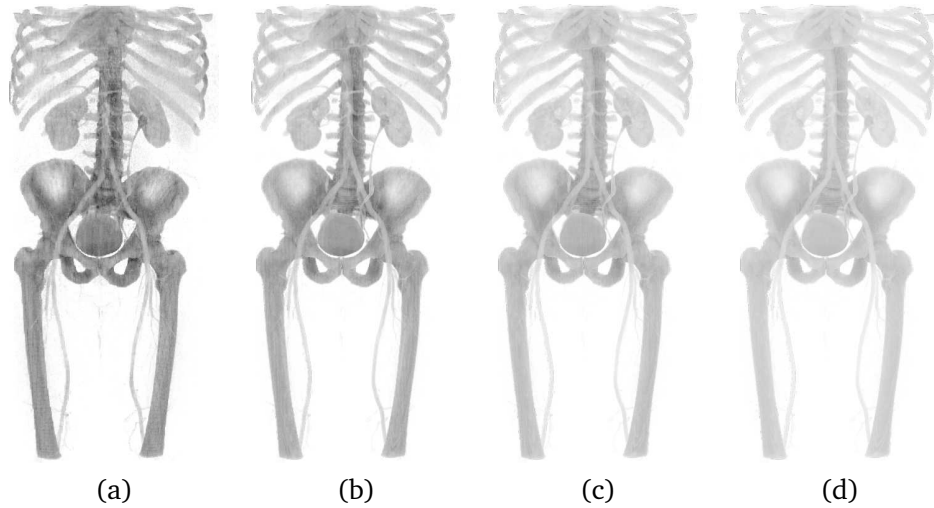


Figure 3.4: Ambient occlusion obtained from VOMI of the CT-body model for 12, 20, 42, and 162 viewpoints.

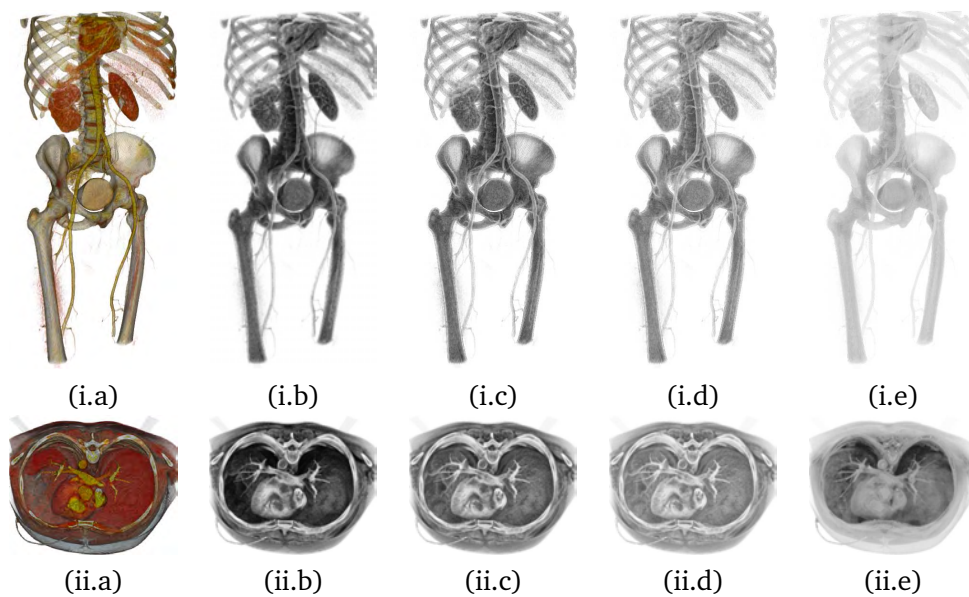


Figure 3.5: AO maps generated using (b) Landis', (c) Stewart's, (d) obscurances, and (e) VOMI approaches for the CT-body model with (a) two different transfer functions.

the application of the ambient occlusion technique [Landis 2002] to volume rendering. Observe that, as expected, the all-or-nothing technique by Landis produces a too contrasted effect, due to too sharp transitions within the discrete set of occlusion values. On the other hand, Stewart's and obscurances methods generate smoother maps because of the continuous range of these values. These AO techniques take only in consideration the local occlusion of the voxel. The VOMI technique works in a different way since it considers the whole visibility, and thus occlusions, from the voxel to all viewpoints. This way, it integrates information of the whole volume with respect to the given voxel. This information will result in an AO map which will capture different effects from the volumetric model, as will be seen below. In the examples, we will consider Stewart's method as representative of local occlusion methods.

The simplest effect is obtained considering the AO value as an ambient lighting AL term. In this case the color of a voxel z is obtained as

$$C(z) = AL(z) = k_i AO(z) C_{TF}(z), \quad (3.6)$$

where k_i is a constant factor that modulates the intensity of $AO(z)$ and $C_{TF}(z)$ is the pure color of the voxel as defined in the transfer function.

In Fig. 3.6 we illustrate the applications of the AO maps as an ambient lighting term, comparing the result of applying a local, classic ambient occlusion method [Stewart 2003] (Fig. 3.6(b)) with our approach (Fig. 3.6(c)). As we have commented above, VOMI takes into account the whole volume visibility, offering a more shaded result than local ambient occlusion. This is clearly visible in the skeleton. The overall information given by VOMI (Fig. 3.6(c)) produces better results than local ambient occlusion (Fig. 3.6(b)) with respect to the raw color information (Fig. 3.6(a)).

A different effect is obtained by adding the AO term to the local lighting equation, as in the global illumination case where ambient occlusion fakes indirect illumination [Iones 2003]. Then, the final color of a voxel is obtained as

$$C(z) = (1 - w_{AO})((k_d N(z) \cdot L) C_{TF}(z) + k_s (N(z) \cdot H)^n) + w_{AO} AL(z) \quad (3.7)$$

where, k_d and k_s are the diffuse and specular lighting coefficients, $N(z)$ is the normal of the voxel, L is the light vector, H is the half-angle vector between L and the direction to the viewer, $AL(z)$ is the ambient lighting, and w_{AO} is the weight of the ambient occlusion in the final color.

In Fig. 3.7 we illustrate the application of the AO maps as an additive term to the local lighting, comparing the result of applying Stewart's method (Fig. 3.7(b)) with our approach (Fig. 3.7(c)). Here again, as in Fig. 3.6, VOMI (Fig. 3.7(c)) produces better results with respect to the direct illumination image of column Fig. 3.7(a) than classic ambient occlusion (Fig. 3.7(b)). Observe that the overall features of the volume model are more distinguishable. Context information is better captured, giving an enhanced depth perception. This is clearly visible in the ribs: while in Fig. 3.7(ii.b) all ribs appear in the same intensity due to the fact that local information is the same for all of them, in Fig. 3.7(ii.c) the inner ribs are darker.

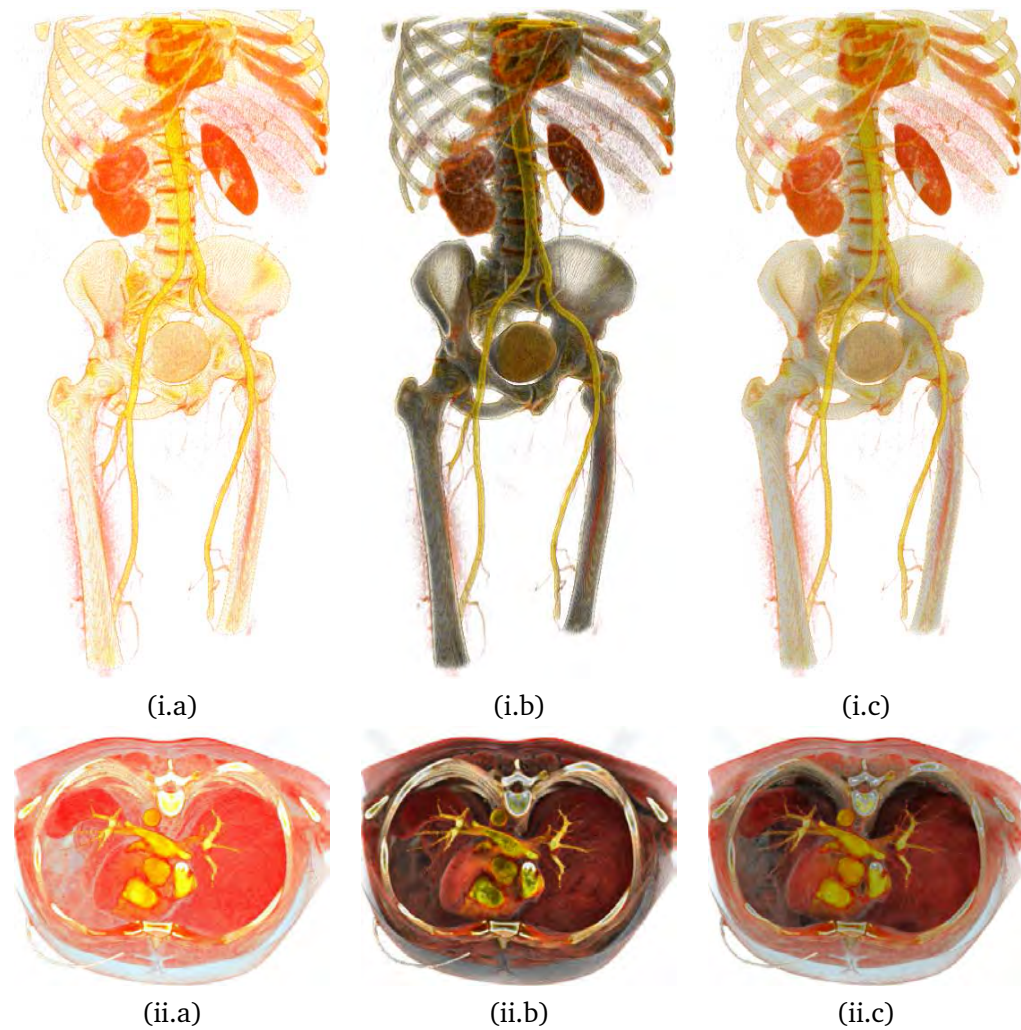


Figure 3.6: The original CT-body model is shown (a) without illumination effects, and illuminated using AO computed with (b) Stewart's method and (c) VOMI, both applied as an ambient lighting term.

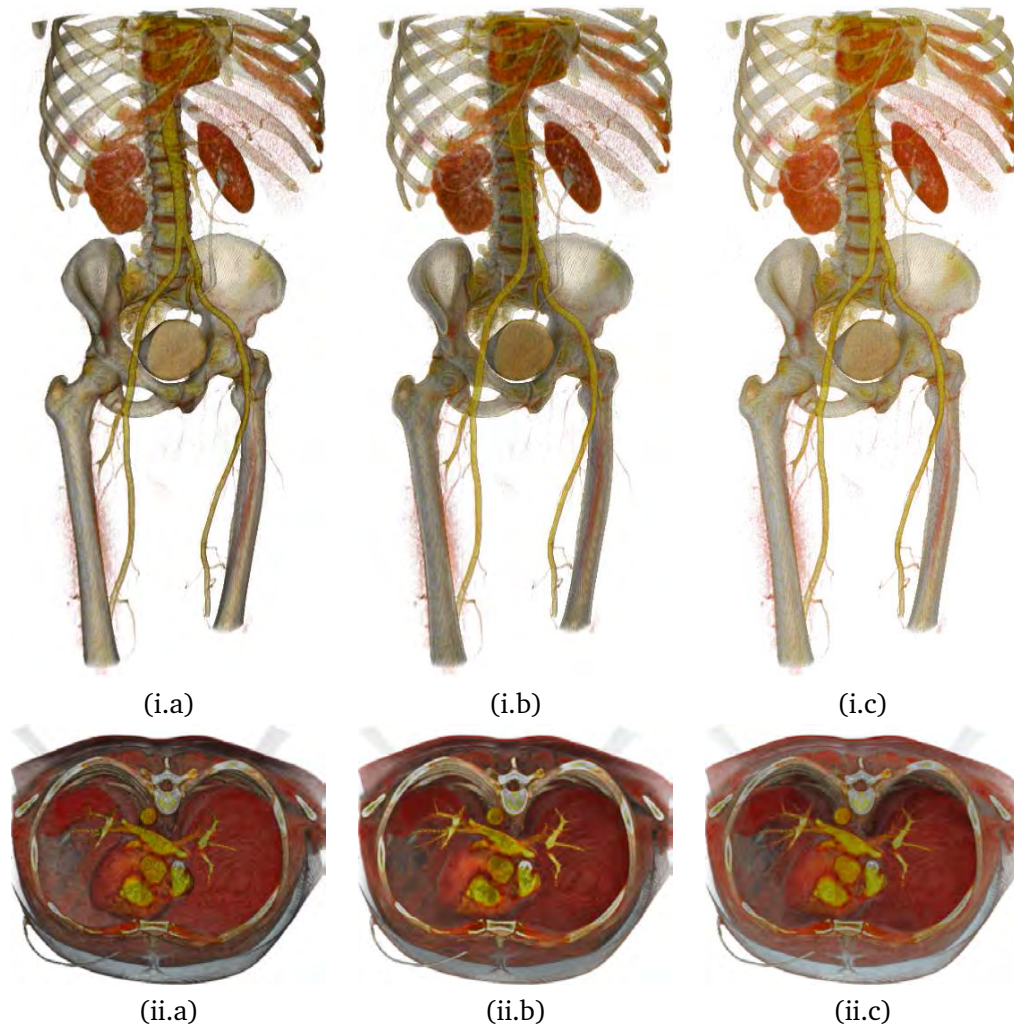


Figure 3.7: The original CT-body model is shown with (a) local lighting, and with AO computed with (b) Stewart's method and (c) VOMI, both applied as an additive term.

3.4.2 Color ambient occlusion

Another effect that can be derived from the voxel information is the *color ambient occlusion*, which simulates the use of colored light sources at the different viewpoints. When all sources have the same color we recover the original AO. This can be obtained from the mutual information matrix and the color associated with each viewpoint. From Eq. (3.4), we can consider that the mutual information matrix $MIM(Z;V)$ is constituted by the terms $I(z;v) = p(v|z) \log \frac{p(v|z)}{p(v)}$ (see Fig. 3.1). Each term represents the shared information between voxel z and viewpoint v . The color ambient occlusion $CAO_\alpha(z;V)$ associated with the voxel z is defined by the scalar product of row z of matrix $MIM(Z;V)$ and the complement of a color vector $C(V)$ assigned to the set of viewpoints:

$$CAO_\alpha(z;V) = \sum_{v \in \mathcal{V}} I(z;v)(1 - C_\alpha(v)), \quad (3.8)$$

where α stands for each color channel, $C_\alpha(v)$ is the normalized vector for channel α , and $I(z;v)$ is a matrix element of $MIM(Z;V)$. After computing VOMI for each channel, the final color ambient occlusion is given by the combination of the color channel values. We can get a color vector by assigning certain colors to specific viewpoints and then interpolating the colors for the rest. In this way, a color is assigned to each viewpoint.

These relighting effects can be easily combined with other illustrative effects, such as color quantization, contours, and cool-and-warm. Fig. 3.8 shows some of these effects applied to the CT-body model considering different transfer functions. Figs. 3.8(a–c) show, respectively, the AO map, the corresponding color ambient occlusion, and the AO map colored using a cool-and-warm technique [Gooch 1998]. Figs. 3.8(d–f) show the use of color ambient occlusion combined with contours and color quantization.

3.4.3 Focus+context

The last application of VOMI is as a *focus+context strategy*. In this case, VOMI is interpreted as a measure of importance and is used to modulate the opacity of a transfer function. The focus of interest is considered as the most informative part of the volume. Then, the opacity of the most informative voxels is increased (or preserved) while the opacity of the least informative is reduced. This opacity modulation effect is driven by the following equation:

$$A'(z) = \begin{cases} A(z)k_l \overline{I(z;V)}, & \text{if } \overline{I(z;V)} < t_l, \\ A(z)k_h \overline{I(z;V)}, & \text{if } \overline{I(z;V)} > t_h, \\ A(z), & \text{otherwise,} \end{cases} \quad (3.9)$$

where $A(z)$ is the opacity of the voxel z before modulation, t_l and t_h are the low and high thresholds respectively, k_l and k_h are factors to regulate the effect of the modulation, $\overline{I(z;V)}$ is the normalized VOMI, and $A'(z)$ is the opacity of the voxel after modulation.

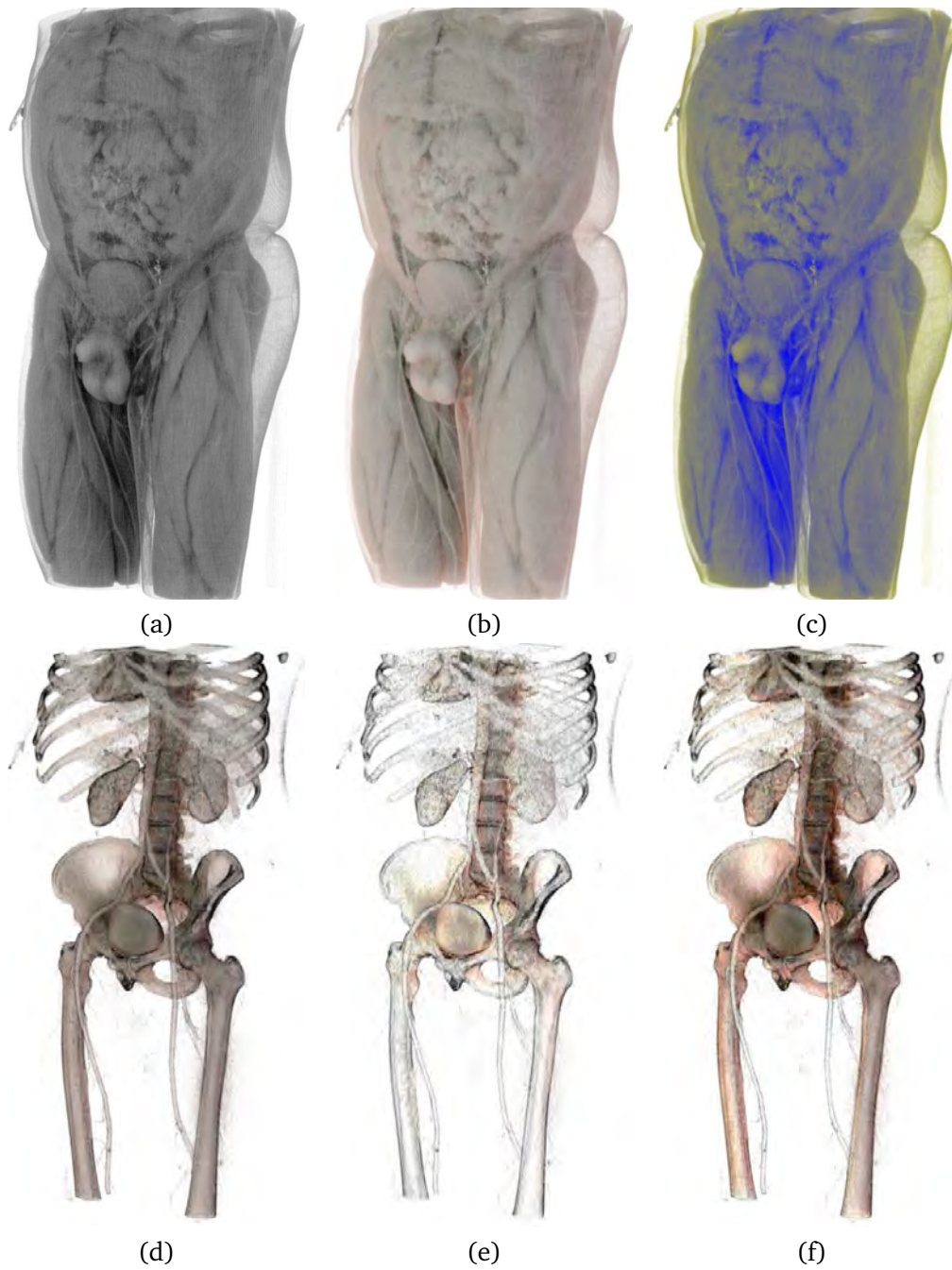


Figure 3.8: CT-body model with different transfer functions and illustrative effects: (a) grayscale AO map, (b) color AO map, (c) cool-and-warm AO map, (d) color AO map with contours, (e) and (f) different color AO maps with contours and color quantization.

In the next figures we present the different effects that can be obtained varying the thresholds and factors, the transfer function, or the viewpoints considered for the visualization. In all the cases we have computed VOMI using a set of 162 viewpoints. Fig. 3.9 has been obtained modifying the thresholds and factors in order to emphasize a selected part of the model while preserving the context. Figs. 3.9(a) and 3.9(d) correspond to the original CT-body with $t_l = 0$ and $t_h = 1$ viewed from different viewpoints and with different transfer functions. In Figs. 3.9(b) and 3.9(c) our target is the skeleton. As this is a highly occluded part, i.e., it has a high VOMI, to reach our objective, we have to decrease the opacity of less occluded parts, such as muscles, which have low VOMI. In Fig. 3.9(b) we obtain this effect by setting t_l to 0.5 and t_h , k_l and k_h to 1. In Fig. 3.9(c) we get a more extreme effect by changing k_l to 0.5, thus making less occluded parts even more transparent. In Fig. 3.9(e) we focus on the ribs, therefore, we want to make the muscles around them more transparent. We achieve this with $t_l = 0.3$, $t_h = 1$, $k_l = 0.1$, $k_h = 1$.

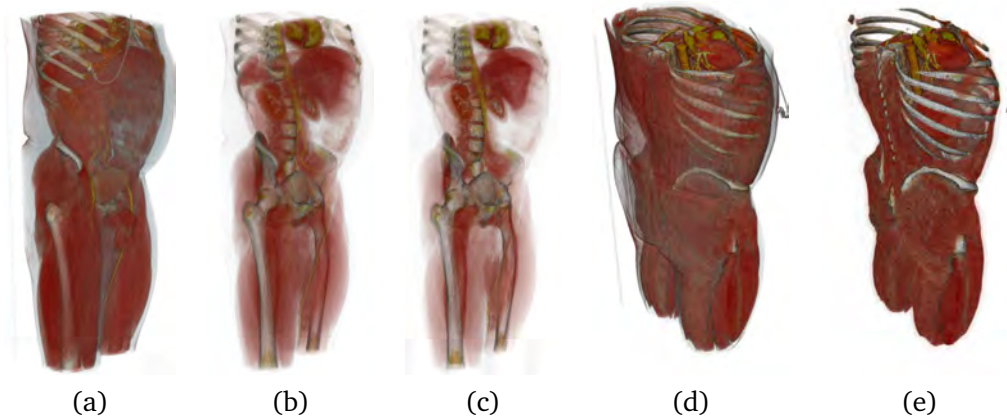


Figure 3.9: CT-body model visualized with different transfer functions: (a) and (d) in their original states, (b) and (c) modulated from (a) by VOMI to emphasize the skeleton, (e) modulated from (d) by VOMI to emphasize the ribs.

VOMI depends on the transfer function used to visualize the model. In the previous example, we modulate the transfer function of the volume with the VOMI computed with that same transfer function. However, it is also possible to compute the VOMI with one transfer function and use it to modulate another one over the same model. To show this effect, we use the VOMI computed with the transfer function used in Fig. 3.9(a), to modulate the one used in Fig. 3.10(a). Fig. 3.10(b) is obtained setting the parameters to the same values as in Fig. 3.9(b). Since muscle in Fig. 3.9(a) is more transparent than in Fig. 3.10(a), modulating the opacity of the latter with the VOMI map of the former makes the muscle more transparent than it would be with its own VOMI map.

Until now, we have always computed VOMI from a set of viewpoints uniformly distributed over the surface of a sphere, but it is also possible to compute it from a subset consisting of one of the viewpoints and its neighbours. This can be useful to emphasize a part of the volume seen from that viewpoint while preserving the rest. For instance, in order to emphasize the right hip and the femur we compute the VOMI map

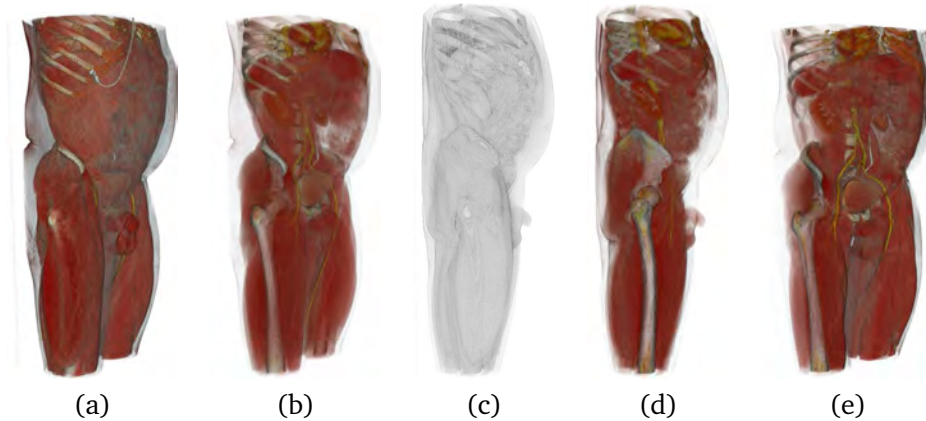


Figure 3.10: CT-body model visualized with the transfer function: (a) of the original model, (b) modulated by VOMI computed with the transfer function used in Fig. 3.9(a), (d) and (e) modulated by VOMI computed from the viewpoint in (c).

considering the viewpoint at the right side of the model and its neighbours. Fig. 3.10(c) shows the obtained VOMI map, and Fig. 3.10(d) and Fig. 3.10(e) show the modulation of Fig. 3.10(a) using this map. Note how our target is emphasized.

3.5 Viewpoint selection using voxel information

In this section we introduce a new viewpoint selection measure based on voxel information. Then we analyze the behavior of viewpoint entropy and viewpoint mutual information compared with the new measure.

Once we have calculated the information associated to the voxels of a volume data set, this information can be “projected” on a viewpoint in order to obtain its informativeness. This method has been previously used to select the most informative views for polygonal models [Feixas 2009]. The information projection over a viewpoint v can be done weighting the VOMI of voxel z by the transition probability $p(v|z)$ and summing over all voxels. Thus, the *informativeness* (INF) of a viewpoint v is defined by

$$INF(v) = \sum_{z \in \mathcal{Z}} p(v|z) I(z; V). \quad (3.10)$$

This represents the total voxel information seen by each viewpoint. Thus, high values of INF will correspond to viewpoints which see a lot of voxel information, i.e., highly occluded parts of the model. In many cases, these parts with high voxel information values show relevant details of the model. On the other hand, low values of INF correspond to low voxel information that is usually associated with smooth changes in visibility and less detail. This will be seen in the examples.

As we have seen in Section 2.9, different information-theoretic viewpoint measures have been introduced to select the “best” views. But the “goodness” of a view cannot be separated of the pursued objective. Thus, if our objective is to see the maximum

number of voxels, viewpoint entropy (Eq. (3.3)) can be the most appropriate measure. This is due to the fact that the maximum entropy would be obtained when all the voxels were seen with the same projected visibility. Minimum entropy would be obtained when only one voxel was visible. On the other hand, viewpoint mutual information (Eq. (3.2)) can be used to detect the most representative views. That is, the views that are most similar to the virtual view of the object obtained from the projection of all viewpoints. The main difference between VE and VMI is that, while VE is very sensitive to the resolution of the volumetric data set, VMI is very robust to deal with any type of segmentation [Viola 2006]. Due to the regular discretization of the volume data set in voxels, the behavior of VE and VMI is not significantly different in the experiments shown below.

Fig. 3.11 shows the views which capture the maximum and minimum VE, VMI, and INF. For each model, the first row corresponds to the “best” views (maximum VE, minimum VMI, and maximum INF) and the second row to the “worst” views (minimum VE, maximum VMI, and minimum INF). Observe the different behaviour of the presented viewpoint measures. While maximum VE and minimum VMI present a relatively similar behaviour, showing respectively the maximum number of voxels in a uniform way and the most representative view, maximum INF is devoted to show the maximum number of highly occluded voxels.

3.6 Conclusions

In this chapter, we have presented a visibility channel to obtain illustrative renderings of volume data sets. The viewpoints and the voxels are the input and output distributions, respectively, for an information channel which is further defined by the visibility values of the voxels referred to each viewpoint. By reversing this channel we assign to each voxel an information value which can be shown to represent an ambient occlusion value. This quantity has been used for illustrative rendering purposes and combined with a lighting texture to further enhance the volume data. In addition, we have analyzed how the voxel information can be used to modulate transfer functions. Finally, we have used the voxel information to select the most informative viewpoints.

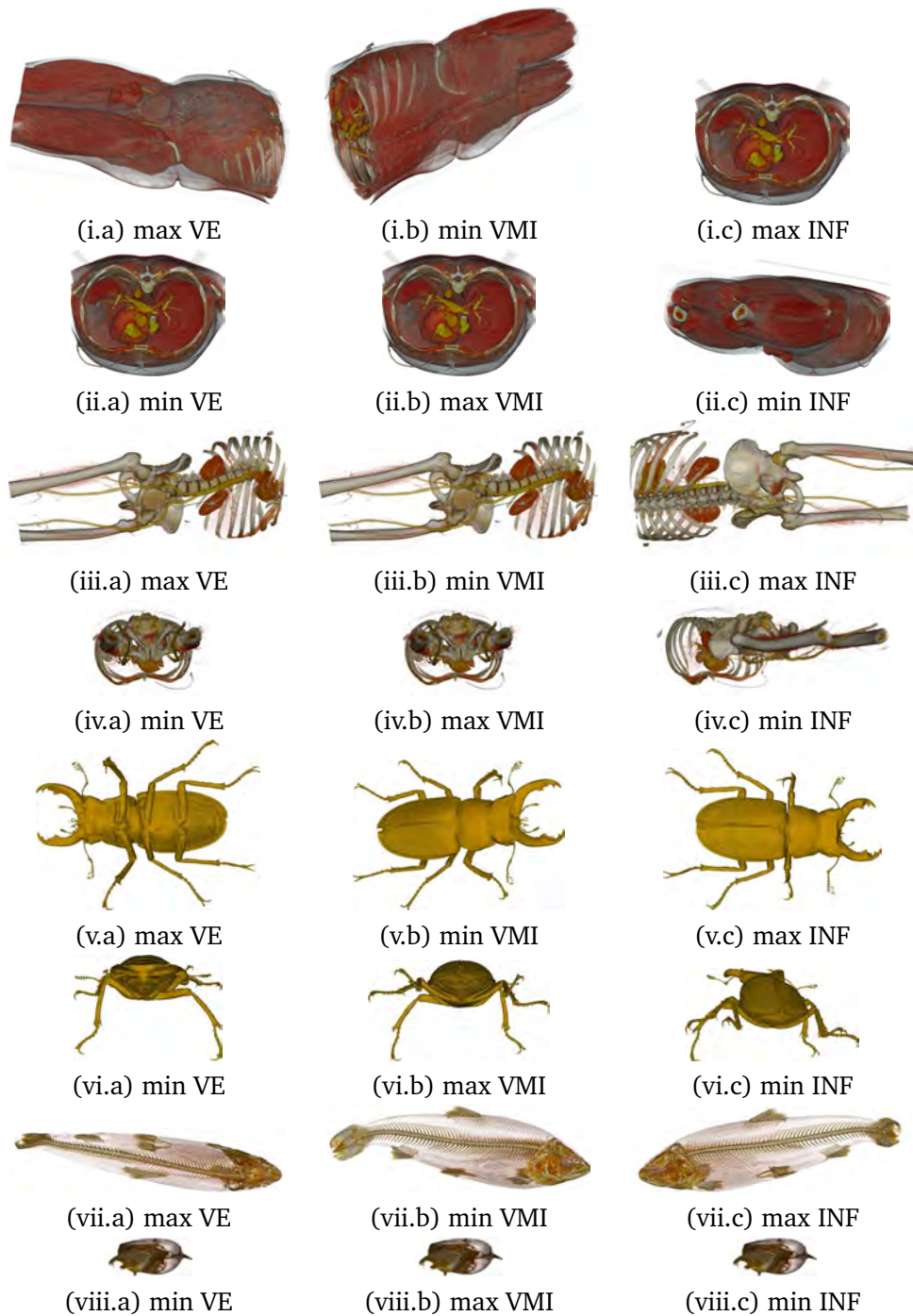


Figure 3.11: Selected viewpoints over a set of 162 with various models according to (a) viewpoint entropy, (b) viewpoint mutual information, and (c) informativeness.

Similarity-based exploded views

Contents

4.1	Introduction	53
4.2	Background	54
4.3	Similarity-steered visualization	54
4.4	Selection of structured views	56
4.5	Evaluating similarity	59
4.5.1	Model partitioning	59
4.5.2	Slice grouping	60
4.5.3	Results	61
4.6	Conclusions	63

4.1 Introduction

Volume visualization aims at gaining insight into volumetric data using interactive graphics and imaging techniques. Current volume data sets generated by scientific domains contain large amounts of data of complex structures. Effective visualization of such data sets that clearly shows all contained structures is challenging.

Illustrative visualization enhances the expressiveness of volume rendering by applying hand-crafted illustrative techniques. Cut-aways, exploded views or high-level abstraction strategies, amongst others, are used to reveal insights and represent essential structures of the volume in a clear way while less important details are subjugated. To employ these techniques, certain controlling mechanisms based on data or higher semantical levels (e.g. segmentation into objects from the domain perspective and the assigning of object importance based on the given domain scenario) are required. These mechanisms vary from fully interactive steered by user (e.g. voxel-by-voxel segmentation) to fully automatic techniques (e.g. shape analysis of the acquired data based on higher-order derivatives). To explore unclassified data sets, automatic controlling mechanisms for steering expressive visualization are useful, and possibly can be combined with interactive techniques that *fine-tune* the first automatic *educated guess*.

Our interest is now focused on exploded views, which partition the volume into different parts that are displaced away from each other as if there had been a small controlled explosion emanating from the focus of interest. Exploded views enable to

see details of otherwise overlapping structures, exploiting the observer's understanding of the original spatial arrangement. In this chapter, a new partitioning approach for automatic generation of exploded views is presented. This method divides the data set into a set of slabs defined by parallel planes, combining in this way the advantages of 2D and 3D views. While 3D visualization provides a global view of the entire model, the 2D cross sectional views reveal insights. To partition the volume, two alternative strategies are proposed. The first one starts with the entire volume and partitions it recursively guided by a *maximum dissimilarity* criterion. The second one considers initially all individual slices and groups them together according to a *similarity* criterion. In both cases, the controlling mechanism is the similarity value that is computed automatically using information-theoretic measures. The only necessary interaction of the user with the data is a single threshold parameter which determines when the partitioning (or grouping) has to stop. An important advantage of this approach is that no a priori information or pre-processing of the data is required. This is suitable, especially, for computer-guided exploration of histology volume data.

4.2 Background

The main limiting factor when exploring volume data is the occlusion between structures. For complex volumetric data sets it is difficult to achieve a visual representation that not only shows all the internal structures but also preserves the global representation of the model. To enhance volume data interpretation Rheingans and Ebert [Rheingans 2001] introduced the volume illustration approach, combining the familiarity of a physics-approximated illumination model with the ability to enhance important features using non-photorealistic rendering techniques. Several illustrative techniques are reviewed in Section 2.6. One of the limitations of the technique relevant to this paper, exploded views, is to decide the parts that have to be exploded. We aim to automatically partition the volume based on characteristics of the data.

Viewpoint selection background is also relevant to this chapter. Several methods on this topic have been reviewed in sections 2.8 and 2.9.

4.3 Similarity-steered visualization

To automatically obtain the partitioning planes for the exploded views, we propose a two-step process. First, we select the view of the model along which the organs or components will be better separated. This view is called the *most structured view* of the model. Second, we calculate the partitions of the model along the most structured view. Such partitions will be obtained using two complementary approaches: a top-down strategy that divides the model according to the maximum information gain and a bottom-up method that joins the slices according to a similarity criterion. Then, the explosion of the model is visualized in the interactive system VolumeShop [Bruckner 2005]. The two steps of the method are described below.

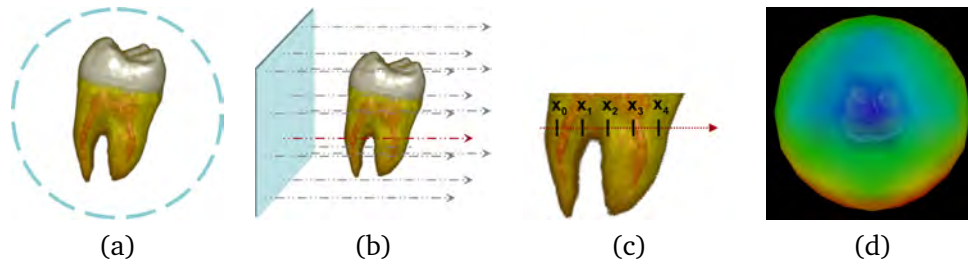


Figure 4.1: Main steps of the selection of the most structured view. (a) Sphere of viewpoints, (b) sampling process for one viewpoint, (c) samples considered for the entropy rate computation, and (d) colored viewpoint sphere (using a thermic scale, from blue to red) representing the values of the viewpoint entropy rate.

1. Selection of splitting axis

The goal of this step is to obtain the most structured view of the model. To reach this objective a viewpoint measure able to capture the structure of the volumetric dataset along any view axis is used. In information theory, *entropy rate* is defined as a measure of the irreducible randomness of an object or the degree of unpredictability of a sequence of values. Since a high randomness corresponds to a low structure and vice versa, we can use the entropy rate to quantify the degree of structure or predictability of a model. We proceed as illustrated in Figure 4.1. First of all, the model is centered in a viewpoint sphere built from the recursive discretisation of an icosahedron (Fig. 4.1(a)). Then, for each viewpoint the entropy rate is computed as described in Section 4.4 (Fig. 4.1(b) and Fig. 4.1(c)). Finally, we identify the lowest entropy rate value which corresponds to the most structured view of the model (Fig. 4.1(d)). This direction is used as axis to which similarity-based partitioning planes are perpendicular to.

2. Volume partitioning

This task consists of selecting the optimal partitions of the model from the most structured view. To carry out this process two different strategies are presented:

- (a) **Top-down approach.** Initially, the entire volume is considered as a single slab and partitioning planes are taken perpendicular to the most structured view (Fig. 4.2(i.a)). To divide the dataset into different parts, we use a greedy algorithm which successively selects the partition that provides us with the maximum gain of information. According to the information bottleneck method [Tishby 1999, Slonim 1999], the information gain can be calculated using the Jensen-Shannon divergence between two parts of the model (Fig. 4.2(i.b)). This measure can be interpreted as the degree of dissimilarity between the parts and attempts to divide the model into homogeneous regions (Fig. 4.2(i.c)). A more detailed description of this approach is given in Section 4.5.1.
- (b) **Bottom-up approach.** All the slices of the volume, perpendicular to the

most structured view, are considered as the initial slabs (Fig. 4.2(ii.a)). Neighboring slabs are iteratively grouped (Fig. 4.2(ii.b)) when mutual information between them is higher than a given threshold (Fig. 4.2(ii.c)). Dealing with similarity between slabs instead of individual slices, we avoid an incorrect grouping, for instance, due to smooth changes along many consecutive slices. The grouping process is further described in Section 4.5.2.

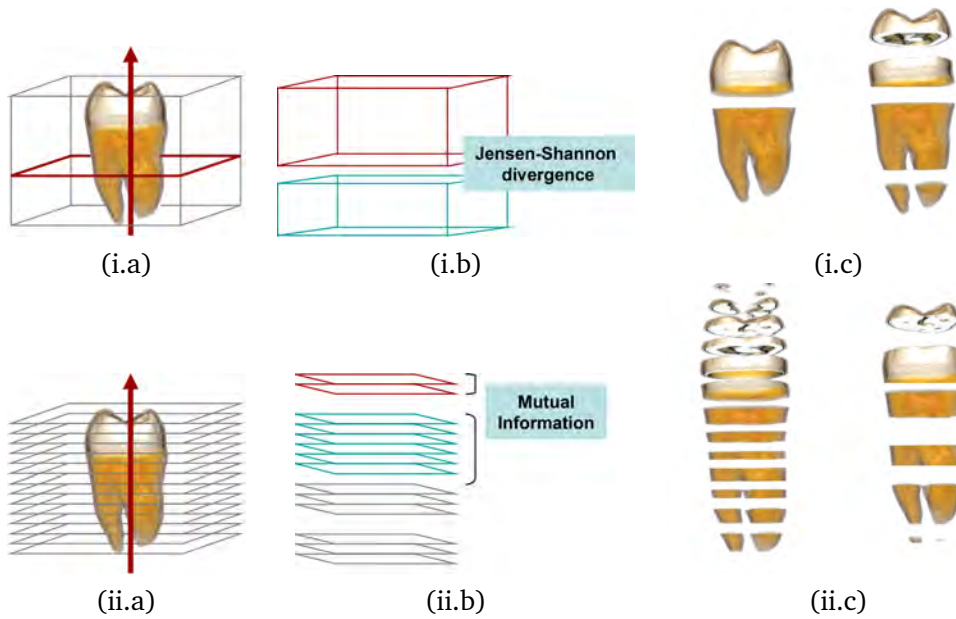


Figure 4.2: (i) Top-down volume partition: (i.a) partitioning planes are taken perpendicular to the most structured view, (i.b) dissimilarity between subvolumes is given by the Jensen-Shannon divergence, and (i.c) examples showing two different partitions. (ii) Bottom-up volume partition: (ii.a) slices are taken perpendicular to the most structured view direction, (ii.b) similarity between slices or slabs is computed using mutual information, and (ii.c) two examples resulting from the grouping process.

4.4 Selection of structured views

To quantify the degree of structure of a volumetric data set along a given viewing direction, we estimate the entropy rate of the sequence of values (intensities) obtained by casting a bundle of parallel lines along that direction. These lines act as probes to sample the intensity of the model. The view with the lowest entropy rate will correspond to the most structured view.

The definitions of both Shannon entropy and entropy rate [Cover 1991] are now reviewed (see Section 2.7). In this chapter, the notation used (inspired by the work of Feldman and Crutchfield [Feldman 2003]) is slightly different from the one presented in Section 2.7.

Let \mathcal{X} be a finite set and X a random variable taking values x in \mathcal{X} with probability distribution $p(x) = \Pr[X = x]$. The *Shannon entropy* $H(X)$ of a random variable X is defined by

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x). \quad (4.1)$$

The Shannon entropy measures the average uncertainty of random variable X . If the logarithms are taken in base 2, entropy is expressed in bits.

Given a sequence $X_1 X_2 \dots$ of random variables X_i taking values in \mathcal{X} , a block of L consecutive random variables is denoted by $X^L = X_1 \dots X_L$. The probability that the particular L -block x^L occurs is denoted by joint probability $p(x^L) = p(x_1, \dots, x_L)$. The *joint entropy* of a block of L consecutive symbols or *L -block entropy* is defined by

$$H(X^L) = - \sum_{x^L \in \mathcal{X}^L} p(x^L) \log p(x^L), \quad (4.2)$$

where the sum runs over all possible L -blocks.

The *entropy rate* or *entropy density* is defined by

$$h = \lim_{L \rightarrow \infty} \frac{H(X^L)}{L} \quad (4.3)$$

and measures the average amount of information per symbol x [Cover 1991].

It can also be rewritten as

$$h = \lim_{L \rightarrow \infty} (H(X^L) - H(X^{L-1})). \quad (4.4)$$

The entropy rate of a sequence of symbols is a measure of its uncertainty, randomness or unpredictability. The entropy rate is also a measure of the compressibility of a sequence: the higher the uncertainty, the lower the compressibility. For instance, in a text, if there are strong correlations between letters (or words), knowledge of all previous letters (or words) will significantly decrease our uncertainty about the next one [Feldman 2003].

How to compute the entropy rate for a given viewpoint is now shown. Consider the scheme in Figure 4.1. For each viewpoint, a sequence of samples (intensity values) to compute the measure is obtained performing a ray casting from a plane centered at the viewpoint. We proceed as follows:

- From the plane at each viewpoint, parallel rays with a regular horizontal and vertical spacing x are cast. Along the ray within the volume, equidistant samples at distance y are taken.
- To build the two joint histograms of X^L and X^{L-1} required for the entropy rate computation, we take into account all possible groups of consecutive samples of length L and $L - 1$, respectively. For example, with the samples shown in Figure 4.1(c), we can form three blocks of length 3 ($x_0 x_1 x_2$, $x_1 x_2 x_3$ and $x_2 x_3 x_4$) and four of length 2 ($x_0 x_1$, $x_1 x_2$, $x_2 x_3$ and $x_3 x_4$) for X^3 and X^2 histograms,

respectively.

- The joint probability distributions $p(x^L)$ and $p(x^{L-1})$ are estimated from the joint histograms of X^L and X^{L-1} , and then the joint entropies $H(X^L)$ and $H(X^{L-1})$ are calculated.
- Due to the potentially high dimensionality of the histograms and, consequently, the high number of components, a trade-off between the number of symbols (intensities) and the length L of the blocks has to be considered. Note that the size (number of entries) of the highest histogram is $O(N^L)$, where N is the number of different property values and L is the length of the blocks. Usually voxel models have property values of 8 bits or more, so this problem is untreatable even with short blocks. As entropy rate is a limit quantity (4.4), its computation would require an infinite number of elements and blocks infinitely long. It has to be approximated using a block of finite length. From the two possible approximations coming from (4.3) and (4.4), we have selected the last one because it approximates more accurately the entropy rate for low values of L . In our experiments, we have taken $L = 3$. We have also reduced the number of symbols of \mathcal{X} (intensity bins in the histogram) to 32, rescaling the intensity bins.

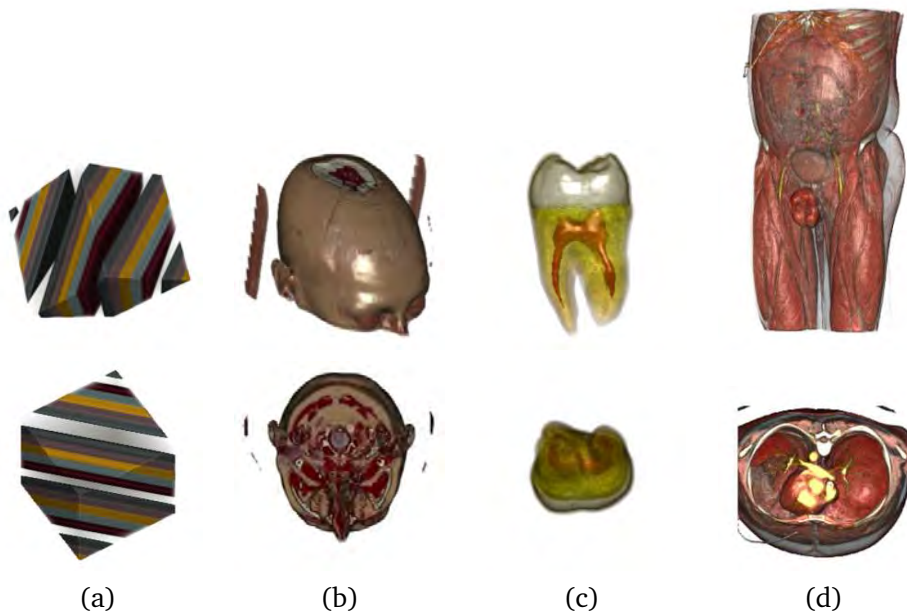


Figure 4.3: Different volumes (first row) and their corresponding most structured views (second row). From left to right: (a) a synthetic model, (b) a CT scan of a patient with hemorrhage, (c) a CT scan of a tooth and (d) a CT scan of the human body.

The strategy for the selection of the most structured view has been applied to different volume data sets. The obtained results are illustrated in Figure 4.3 where the first row represents the original model and the second row the most structured view.

From left to right, the proposed models correspond to: (a) a synthetic model, created considering six different materials (each one represented with a different color) which follow a diagonal distribution through the volume, (b) a CT-scan of a patient with an hemorrhage, (c) a CT-scan of a tooth and (d) a CT-scan of the human body. Observe how the best views show the maximum of structure in the model. This is specially noticeable in the phantom model (Fig. 4.3(a)) where the different regions have an inclination relative to the cube axis.

4.5 Evaluating similarity

To obtain the optimal partitions for the explosion of a 3D data set, two different strategies are presented. First, we analyze a top-down approach which partitions the model using a criterion of maximum gain of information. Second, we study a bottom-up strategy that groups the slices according to a similarity measure between them.

4.5.1 Model partitioning

Once the most structured direction of the model has been selected, a sequence of perpendicular partitions in that direction can be obtained using a simple greedy algorithm. This is a top-down hierarchical application of the information bottleneck method [Tishby 1999, Slonim 1999] which permits us to measure the gain of information when a model is divided into different slabs. This *gain of information* is computed using the Jensen-Shannon divergence (see Section 2.7.3.3).

The *Jensen-Shannon divergence* [Burbea 1982] between probability distributions p_1, p_2, \dots, p_N with prior probabilities or weights $\pi_1, \pi_2, \dots, \pi_N$ is defined by

$$JS(\pi_1, \pi_2, \dots, \pi_N; p_1, p_2, \dots, p_N) = H\left(\sum_{i=1}^N \pi_i p_i\right) - \sum_{i=1}^N \pi_i H(p_i), \quad (4.5)$$

where $\sum_{i=1}^N \pi_i = 1$. The JS-divergence measures how far the probabilities p_i are from their likely joint source $\sum_{i=1}^N \pi_i p_i$ and equals zero if and only if all the p_i are equal. From [Slonim 1999] (see also Eq. (2.35)), it can be seen that the gain in information when a dataset is divided into two slabs is given by

$$\delta I = \frac{v_1 + v_2}{v_T} JS\left(\frac{v_1}{v_1 + v_2}, \frac{v_2}{v_1 + v_2}; p_1, p_2\right), \quad (4.6)$$

where v_1 and v_2 are, respectively, the volumes of slabs 1 and 2, v_T is the total volume of the 3D dataset, p_1 and p_2 are, respectively, the normalized intensity histograms of slabs 1 and 2, and $JS\left(\frac{v_1}{v_1 + v_2}, \frac{v_2}{v_1 + v_2}; p_1, p_2\right)$ is the Jensen-Shannon divergence between p_1 and p_2 with the corresponding weights $\frac{v_1}{v_1 + v_2}$ and $\frac{v_2}{v_1 + v_2}$.

The gain of information when a model is divided into two parts is given by the dissimilarity between them (measured by JS) weighted by their relative volume. Note

that a slab highly structured along a given direction will have all possible partitions very similar and thus will not need to be partitioned.

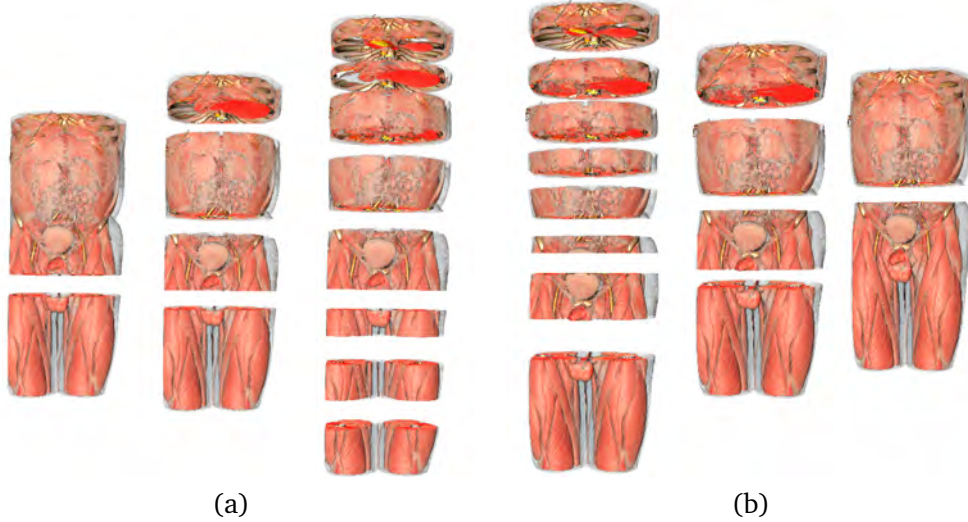


Figure 4.4: (a) The volume partitioning and (b) the slice grouping approaches applied to a CT scan of the human body.

4.5.2 Slice grouping

Given a viewing direction, the slices perpendicular to it can be grouped using a similarity measure. In this paper, the normalized mutual information is used to quantify the degree of similarity between individual slices or groups of adjacent slices (slabs). In medical imaging, many successful automatic image registration methods are based on the maximization of mutual information. This method, introduced by Viola [Viola 1995] and Maes et al. [Maes 1997], is based on the conjecture that the correct registration corresponds to the maximum mutual information between the overlap areas of the two images. Later, Studholme et al. [Studholme 1997] proposed to use the normalized mutual information as it is more robust than solely mutual information due to its greater independence of the overlap area.

Let X and Y be two random variables taking values x and y in finite sets \mathcal{X} and \mathcal{Y} with probability distributions $p(x) = \Pr[X = x]$ and $p(y) = \Pr[Y = y]$, respectively. The *mutual information* between X and Y is defined by

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \quad (4.7)$$

where $p(x, y) = \Pr[X = x, Y = y]$ is the joint probability. Mutual information is a measure of the shared information or the degree of dependence between X and Y . Mutual information is zero only if the two random variables are strictly independent.

The *normalized mutual information* is defined by

$$NMI(X, Y) = \frac{I(X; Y)}{H(X, Y)}, \quad (4.8)$$

where $H(X, Y)$ is the joint entropy of X and Y . NMI takes values in the range $[0, 1]$.

An explanation is now given to compute the NMI measure between slices and the algorithm to group them. Given two slices A and B from the volume dataset, with associated random variables X and Y , the joint probability distribution $p(x, y)$ can be estimated by simple normalization of the joint histogram $h(x, y)$ of both slices. This is obtained from the intensities of each pair (a, b) of corresponding voxels, where $a \in A$ and $b \in B$. Once the joint histogram has been calculated, the joint probability distribution and the marginal probability distributions of X and Y can be estimated: $p(x, y) = \frac{h(x, y)}{\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} h(x, y)}$, $p(x) = \sum_{y \in \mathcal{Y}} h(x, y)$ and $p(y) = \sum_{x \in \mathcal{X}} h(x, y)$. The similarity measure NMI is then evaluated.

The similarity between two slices can be extended to the similarity between two slabs $\hat{A} = \{A_1, \dots, A_n\}$ and $\hat{B} = \{B_1, \dots, B_m\}$. The random variables \hat{X} and \hat{Y} , associated with both slabs, represent the grouping of a set of random variables $\{X_1, \dots, X_n\}$ and $\{Y_1, \dots, Y_m\}$, respectively. Their joint frequency histogram is obtained from the intensities of each pair of corresponding voxels (a_i, b_j) , where $a_i \in A_i$ and $b_j \in B_j \forall i, j$. As mentioned above, the joint and marginal probability distributions can be estimated and thus the NMI measure is obtained.

Given the similarity measure NMI, the algorithm proceeds by joining the two adjacent slabs with maximum similarity. This process stops when the similarity between them is above a user-defined threshold or a number of slabs has been reached. At the beginning, every slab consists of only one slice. Then, the most similar slabs are progressively joined. To group n slices, the algorithm proceeds as follows:

- Assign n slabs such that each slab contains exactly one slice.
- Compute NMI for each pair of consecutive slabs.
- Find the two closest consecutive slabs i and $i + 1$ (with maximum NMI). If the similarity between them is higher than the given threshold, then create a new slab \hat{i} by combining i and $i + 1$ and recalculate NMI for the neighboring slabs of \hat{i} . This step stops when the similarity between each pair of consecutive slabs is lower than a fixed threshold or a number of slabs is achieved.

4.5.3 Results

The proposed approaches have been implemented and integrated into the VolumeShop framework [Bruckner 2005]. To test the methods, different synthetic and real data sets have been considered. In all the tests a sphere of 42 viewpoints has been used and the stopping criterion has been fixed by the number of slabs entered by the user.

For the first tests, the CT scan of the human body of Figure 4.4 has been used. The results obtained with the model partitioning approach for 2, 4 and 8 partitions are

Volume	Best view	P(2)	P(4)	P(8)	G(8)	G(4)	G(2)
Human body	111.0	0.4	0.9	2.0	28.9	36.2	70.5
Hematoma	55.5	0.2	0.6	1.2	6.3	10.8	14.4
Histological data	722.6	2.3	5.8	12.6	214.0	324.7	525.0

Table 4.1: Time cost in seconds for computing the most structured view, and for volume partitioning (P) and slice grouping (G) in 2, 4, and 8 slabs, respectively, for three different volume models: Human body ($256 \times 256 \times 415$), Hematoma ($512 \times 512 \times 45$), and Histological data ($587 \times 342 \times 499$).

illustrated in Fig. 4.4(a). In Fig. 4.4(b) the partitions obtained with the slice grouping approach using the same user parameters are shown. In Figures 4.5(a) and 4.5(b) we illustrate the results obtained by applying the volume partitioning and the slice grouping approaches on a CT scan of a patient with a brain hemorrhage. Observe that the damaged region is located in the second slab from top to bottom.

In Figures 4.5(c) and 4.5(d), the results obtained with the volume partitioning and the slice grouping approaches applied to an histologic data model are shown. It is important to emphasize that these techniques have been applied without prior pre-processing. Time cost for computing the most structured view, and for volume partitioning and slice grouping are given in Table 4.1. Times are given for a system equipped with an Intel Core 2 Quad Q6600 CPU at 2.4 GHz and 2 GiB of memory. Several video sequences are available as supplementary material in www.gametools.org/smartgraphics/.

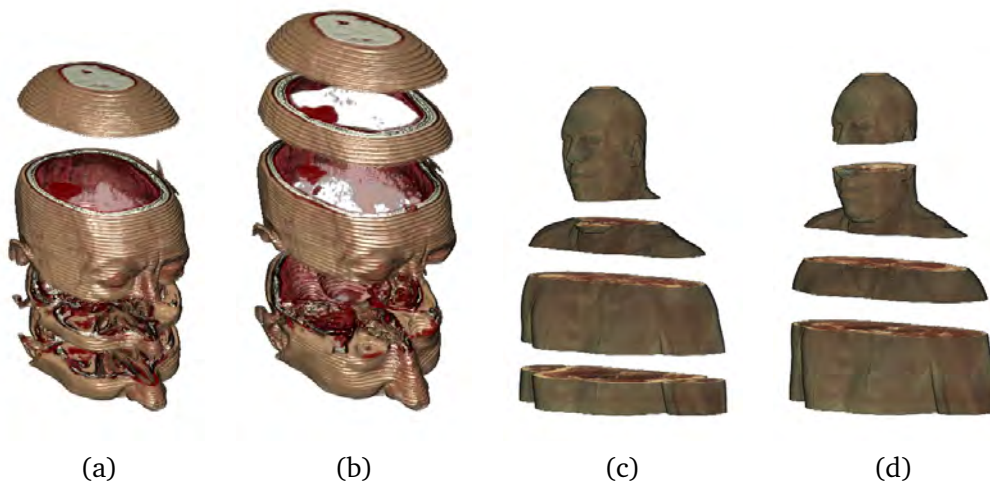


Figure 4.5: (a) The volume partitioning and (b) the slice grouping approaches applied to a CT scan of a patient with a brain hemorrhage. An histological data model decomposed with the (c) volume partitioning and (d) slice grouping methods.

4.6 Conclusions

New partitioning techniques for volumetric data decomposition and visualization using exploded views have been introduced. These techniques use an information-theoretic two-step approach to automatically partition the model. First, the view with the highest structure is identified and, then, the model is divided along this view following two alternative similarity-based methods. The presented techniques provide us an efficient tool for volume data exploration without neither a priori knowledge nor pre-processing of the data.

The proposed approach has been evaluated on different medical data sets and the obtained results have been analyzed by a group of physicians; they consider that the proposed method is potentially useful for automation of certain medical processes, such as the location of lesions. As future work we plan to do a broader study with a more exhaustive analysis of behaviour and usability of the proposed approach for different image modalities and anatomical structures. We plan to test and validate this new method in a large cohort of patients to determine its clinical effectiveness. In addition, we will study how many slices should be presented and whether additional information about the similarity distances can help the user for an optimal understanding of the volume. Also, the trade-off between quality and cost for the different parameters involved in the determination of most structured views (distances between rays, subsequences of samples on those rays, sub-sampling of intensity values) will be investigated. As further improvement, we can also investigate if a neural network trained with physicians knowledge can be incorporated into our algorithm as a guide.

Obscurance-based volume rendering framework

Contents

5.1 Introduction	65
5.2 Background	66
5.2.1 Obscurances and ambient occlusion	66
5.2.2 Volume shadowing versus obscurances	67
5.3 Obscurances for volume rendering	68
5.3.1 Algorithm	68
5.3.2 Analysis of ρ function	70
5.4 Volume saliency	73
5.5 Applications	75
5.5.1 Realistic and illustrative rendering	77
5.5.2 Saliency	77
5.6 Conclusions	78

5.1 Introduction

Global illumination is a well-known technique for producing realistic scenes. However, although it might play a decisive role in 3D volume visualization since it provides visual cues that enhance data interpretation, its application is still challenging in direct volume rendering. The main limiting factor is the computational cost of simulating global effects of light in a volume, making interactive exploration difficult [Max 1995]. On the other hand, illustrative methods aim at creating visualizations which convey information to the viewer as opposed to physically correct light interaction. Volume illustration enhances the expressiveness of volume rendering by highlighting important features within a volume while subjugating insignificant details and rendering the result in a way that resembles an illustration [Rheingans 2001]. Ideally, a volume rendering system should be able to support both realistic and illustrative renderings.

Obscurances have been introduced by Zhukov et al. [Zhukov 1998] and Iones et al. [Iones 2003] as an efficient technique that gives perceptually similar results to global illumination with a small fraction of the computational cost. Moreover, without

adding computational cost, obscurances also allow us to compute *color bleeding*, which consists in the effect that the objects around another object with intense coloration are dyed with this color [Méndez 2003a]. The obscurance technique was first used in video-game environments. Its application to volume rendering, called vicinity shading, was introduced by Stewart [Stewart 2003].

In this chapter, we present an obscurance-based volume rendering system that allows to obtain realistic and illustrative volume visualizations in an interactive manner. One important aspect of our work shows that obscurances are not only useful for realistic depiction but also for illustrative rendering. As obscurances can be interpreted as general information about the neighborhood of a voxel, they can be used as a bias for the generation of more expressive illustrative depictions of a data set (see Figure 5.1).

Saliency typically arises from contrasts between items and their neighborhood [Itti 2001, Tsotsos 2005, van de Weijer 2006] and it is considered that the most salient voxels in a 3D data set will attract the attention of the viewer. In our approach, voxel saliency is determined by the obscurance gradient, which measures the maximum variation of the obscurance field. Once the saliency of the volume is obtained, we implicitly have the saliency map of any structure contained in the volume. This saliency map can be applied to viewpoint selection and to enhance visualization. This can help to discover relevant characteristics of the model otherwise unnoticed by the observer.

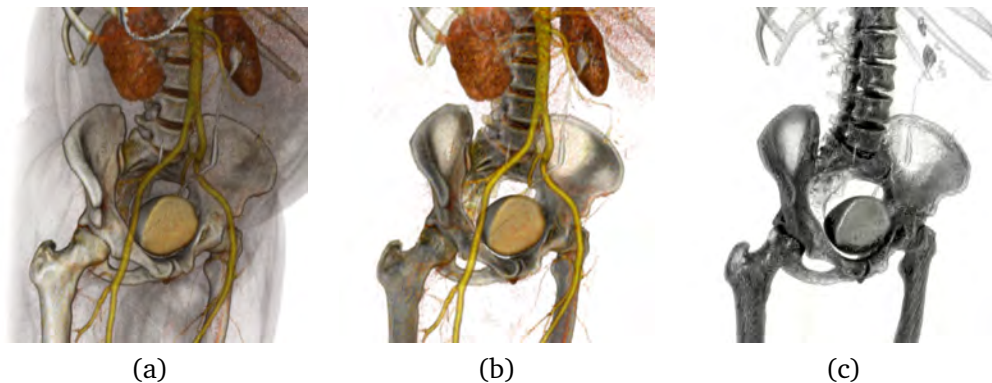


Figure 5.1: CT-human body data set rendered with the proposed obscurance-based volume rendering framework. The images have been obtained by modifying interactively the transfer function and the way in which obscurances are applied to the model.

5.2 Background

In this section, obscurances, ambient occlusion, and related illumination models are described.

5.2.1 Obscurances and ambient occlusion

For a complete background on this topic refer to Section 2.5.2. Here we briefly recall the equations for obscurances and ambient occlusion.

The obscurance O of a point p is defined as the integral

$$O(p) = \frac{1}{\pi} \int_{\Omega} \rho(d(p, \omega)) \cos \theta \, d\omega, \quad (5.1)$$

where ρ is a function of the distance $d(p, \omega)$ of the first intersection of a ray shot from point p with direction ω , p is a surface point, θ is the angle between the normal vector at p and direction ω , and the integration is over the hemisphere oriented according to the surface normal. We only consider a neighborhood of p , i.e. function ρ is set to 1 for distances greater than a maximum distance d_{\max} . Therefore, the integral function $O(p)$ captures occlusion (or openness) information of the environment of point p . Considering the extreme cases, an obscurance value of 1 means that the point is completely open, i.e. not occluded, and a value of 0 means that it is completely occluded.

Ambient occlusion [Landis 2002] is a simplified version of the obscurances illumination model. Ambient occlusion

$$A(p) = \frac{1}{\pi} \int_{\Omega} V(p, \omega) \cos \theta \, d\omega, \quad (5.2)$$

substitutes the ρ function in the obscurances equation (5.1) by the visibility function $V(p, \omega)$ that has value 0 when no geometry is visible in direction ω and 1 otherwise.

Color bleeding consists in the effect that the objects around another object with intense coloration are dyed with this color. To obtain color bleeding, Méndez et al. [Méndez 2003a] included in Equation (5.1) the diffuse reflectivity $R(q)$:

$$W(p) = \frac{1}{\pi} \int_{\Omega} R(q) \rho(d(p, \omega)) \cos \theta \, d\omega, \quad (5.3)$$

where q is the first point in direction ω that occludes p . When no occlusion is found within d_{\max} , the average reflectivity is used. Observe that adding color bleeding to obscurances is almost free.

5.2.2 Volume shadowing versus obscurances

The obscurances/ambient occlusion technique was first used in video-game environments [Zhukov 1998]. A few methods have been presented to adapt this technique to volume rendering. In particular, Stewart proposed vicinity shading [Stewart 2003], which simulates illumination of isosurfaces by taking into account neighboring voxels. An occlusion volume is computed and stored in a shading texture that is accessed during rendering. This volume has to be re-computed each time that the rendering parameters are modified and the method does not support color bleeding. In addition, it only allows to represent one of the surfaces of the volume. Since this first work, several models to illuminate the isosurfaces have been proposed. For a complete review on these techniques, see Section 2.5.1.

Note that there is a twofold difference between those local volumetric shadowing

effects and the classic obscurances (or ambient occlusion) used in our approach. Firstly, obscurances technique uses a ρ function (see discussion in Section 5.3.2) to modulate the effect of the occlusion with the distance. Secondly, obscurances compute explicitly visibility tests. This means that, although the test can be made up to a predefined maximum distance, if an occlusion is found, the rest of the environment in this direction is ignored, contrarily to local volumetric shadowing which integrates for the whole distance. In Figure 5.2, the essential difference between the local volumetric shadowing and the obscurances approach is illustrated. Observe that obscurances only take into account the distance from a voxel to the next occluding one, not what is in between. This is indeed different from Hernell's algorithm, which considers the accumulated visibility of the whole environment, being nearer to the physical realism (or at least more coherent with the transfer functions).

Obscurances (and later ambient occlusion) never claimed to be physically realistic, it was introduced as a fast photorealistic approximation to indirect illumination. Local volumetric methods have a much higher cost, with complexity proportional to the square of the number of voxels, against the complexity of obscurances computation, proportional to the number of voxels times the number of directions. Thus, on a scale of physical realism (and cost) the different strategies can be sorted into increasing order as follows: ambient occlusion (the lowest), obscurances, Hernell's [Hernell 2007] and Ropinski's [Ropinski 2008] approaches, and global illumination (the highest).

5.3 Obscurances for volume rendering

In this section we go further into the obscurance-based volume rendering by testing different distance functions for computing obscurances and providing discussion on quality assumptions.

5.3.1 Algorithm

We take as a basic implementation of the obscurance-based volume rendering approach the one proposed by Stewart in [Stewart 2003]. First, the volume data set is centered in a sphere built from the recursive subdivision of an icosahedron and the lines from each vertex to the center of the volume are taken as the directions to consider (12, 42, and 162 directions have been taken in our experiments). Then, for each direction, the volume is swept using Bresenham's algorithm. This is equivalent to casting parallel rays covering all voxels. Obscurance computation for a given voxel is based on the presence (and the distance) of occluders within a certain radius along the processed direction. A visibility test compares the densities of two voxels which can be intensity values, which are independent of the transfer function, or opacity values assigned by the transfer function. In each case, we say that voxel v_i occludes v_{i-1} if the density of v_i is greater than that of v_{i-1} . To process the voxels we use a stack which stores the previously visited and yet unoccluded voxels in a density-based decreasing order. All the voxels in a ray are traversed and for each one we check if it is the nearest occluder to one of the previous unoccluded voxels (i.e. the ones stored in the stack), and in the

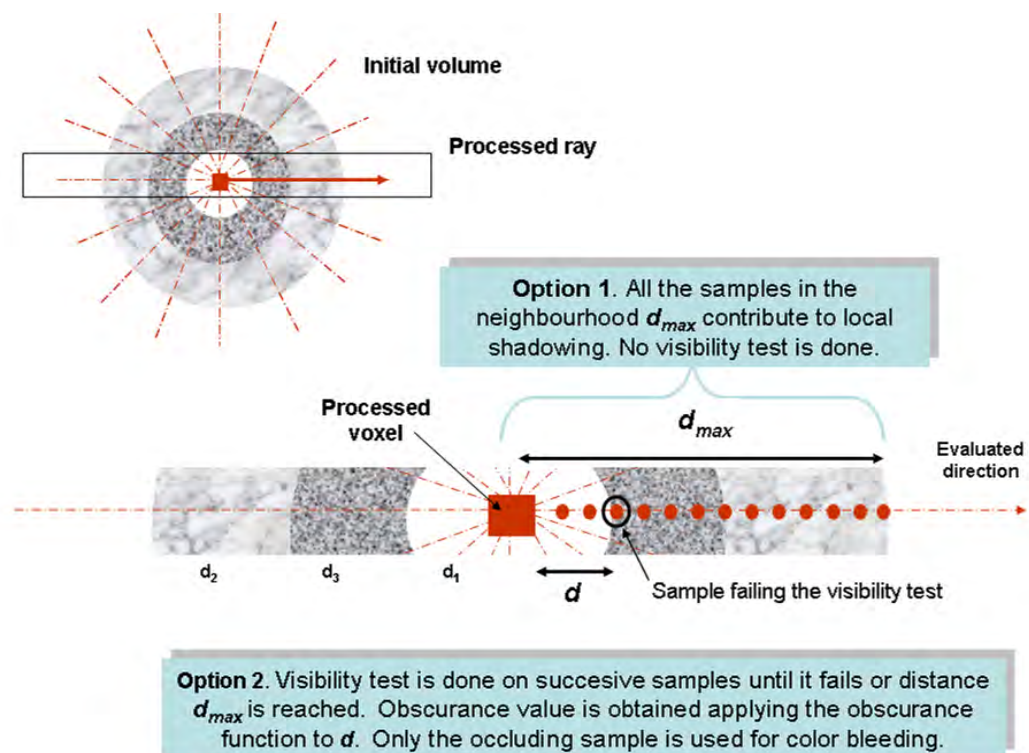


Figure 5.2: Local volumetric shadowing (option 1) and obscurance computation (option 2) for a volume model consisting of three concentric spheres with densities $d_1 < d_2 < d_3$.

occlusion test we check the distance. In each step, we start to check if the current voxel v_i occludes the one on the top of the stack v_s . If v_s is occluded then we can remove it from the stack so that it will not be processed anymore and continue applying the same procedure to v_{s-1} . If v_s is unoccluded, the rest of the stack voxels do not need to be processed since v_s is the voxel of the stack with lower density. Then, the next voxel of the ray is processed. This pre-computed obscuration is stored as vicinity shading values in a separate texture volume which is used during rendering.

In order to integrate color bleeding effects, we multiply the obscuration value by the color of the occluding voxel, and add an ambient color constant (in our case, white) to the unoccluded voxels. In this way, spectral obscurances are accumulated.

5.3.2 Analysis of ρ function

In this section, the meaning and shape of the ρ function in the obscurances definition (5.1) is discussed. First, this function should be a monotonically increasing function of d . Second, this function is bounded from above. This reflects the fact that normally ambient lighting of a given point is primarily affected by its neighborhood. This is especially true for scenes without bright light sources that may affect the illumination at large distances. From 0 to a determined value d_{\max} , the function increases from 0 to 1, and for values greater than d_{\max} the returned value is 1. This means that only a limited environment around the point p is considered and beyond this the occlusions will not be taken into account.

The shape of the ρ function is deduced from the fact that we are interested in what happens in the vicinity of a point (or voxel). The nearer the occlusion, the higher the influence it should have. This influence diminishes with increasing distance. This is reinforced by interpreting the obscuration model with the $\rho(d) = 1 - e^{-\tau d}$ function as the illumination at the non-reflecting boundaries of a non-scattering gas with opacity τ and constant volume emittance. If we consider the occlusions of the environment as having a similar damping opacity effect over the ambient light, we should use a function as similar as possible to $\rho(d) = 1 - e^{-\tau d}$ [Iones 2003]. Some candidate functions (see Figure 5.3) are: (a) all-or-nothing, $\rho(d) = 0$, used in the ambient occlusion approach (it does not allow color bleeding as the contribution of a hit is 0); (b) linear, $\rho(d) = \frac{d}{d_{\max}}$, used in the vicinity shading approach; (c) exponential, $\rho(d) = 1 - e^{-\frac{d}{d_{\max}}}$; and (d) square root, $\rho(d) = \sqrt{\frac{d}{d_{\max}}}$, introduced in [Méndez 2003a]. Note that we have considered the exponential function without normalization since the normalized exponential would become very similar to the linear function.

Different data sets have been used to analyze the effect of each function in the final visualization. The obtained images are shown in Figure 5.4 where each column represents a different ρ : (a) all-or-nothing, (b) linear, (c) exponential, and (d) square root. In the first row, we show the behavior of the above functions (including color bleeding) on a cubic phantom model with 128^3 voxels, where except for three adjacent faces with opacity = 1 (left wall: green, right wall: blue, floor: white), the rest of the voxels are transparent. The effects at the corner illustrate the behaviour of the distance

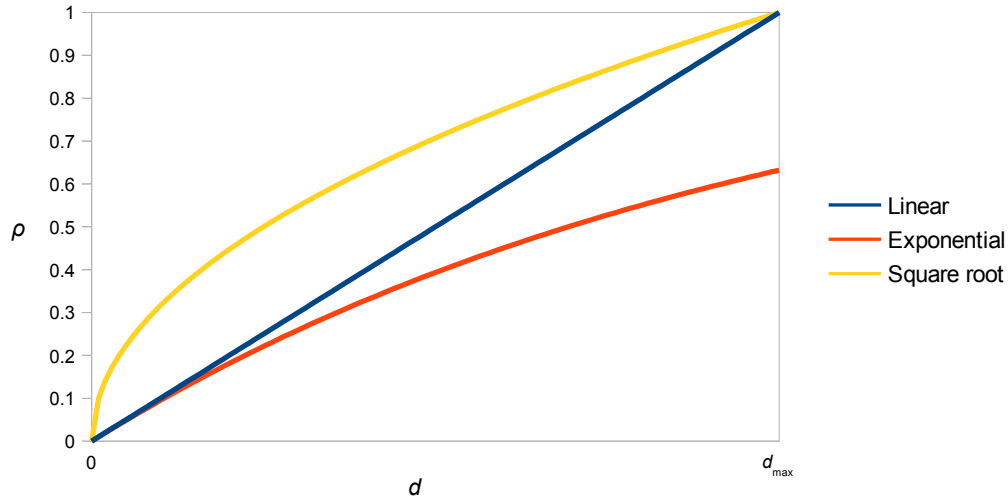


Figure 5.3: Different $\rho(d)$ functions.

function. The linear function ρ (column (b)) corrects in some way the non-smoothed effect of the all-or-nothing function (column (a)), but it considers a wider than necessary environment. The exponential ρ (column(c)) produces darker values, similar to the all-or-nothing case, due to the discontinuity leap at $d = d_{\max}$. The square root function (column (d)) is a good compromise, as it considers a nearer environment and the darkness decays smoothly. This is appropriate to enhance the details of the model. We also study the results of applying the different ρ functions on a CT-human body of $256 \times 256 \times 415$ voxels. The obscurance volumes and the visualizations obtained using an obscurance-based illumination model (see Section 5.5.1) are shown, respectively, in the second and third rows of Figure 5.4. They have been obtained considering 162 viewing directions and a maximum distance equivalent to 64 voxels. Ambient occlusion (column (a)), linear (column (b)), and exponential (column (c)) become darker, and square root (column (d)) appears less dark and more pleasant to the eye.

In the obscurances computation, d_{\max} is also a key parameter to be considered since it controls the number of voxels in a determined direction that have to be taken into account to compute occlusions (see Equation (5.1)). Therefore, if d_{\max} has a high value, the probability of finding an occlusion increases, and hence the obscurance value, leading to darker images. Conversely, if d_{\max} has a low value, the probability to be occluded decreases, leading to low obscurance values and hence lighter images. Figure 5.5 illustrates this effect in two data sets considering different d_{\max} values (8, 64, and 256, respectively). As expected, we can observe how the darkness of the image increases when d_{\max} increases. The other images in this paper have been computed using $d_{\max} = 64$. In Figure 5.5, the effect of the number of directions in the obscurances values is also shown. Results for three different number of directions (12, 42, and 162) are given. Observe that although 12 directions could be considered for fast editing, we need at

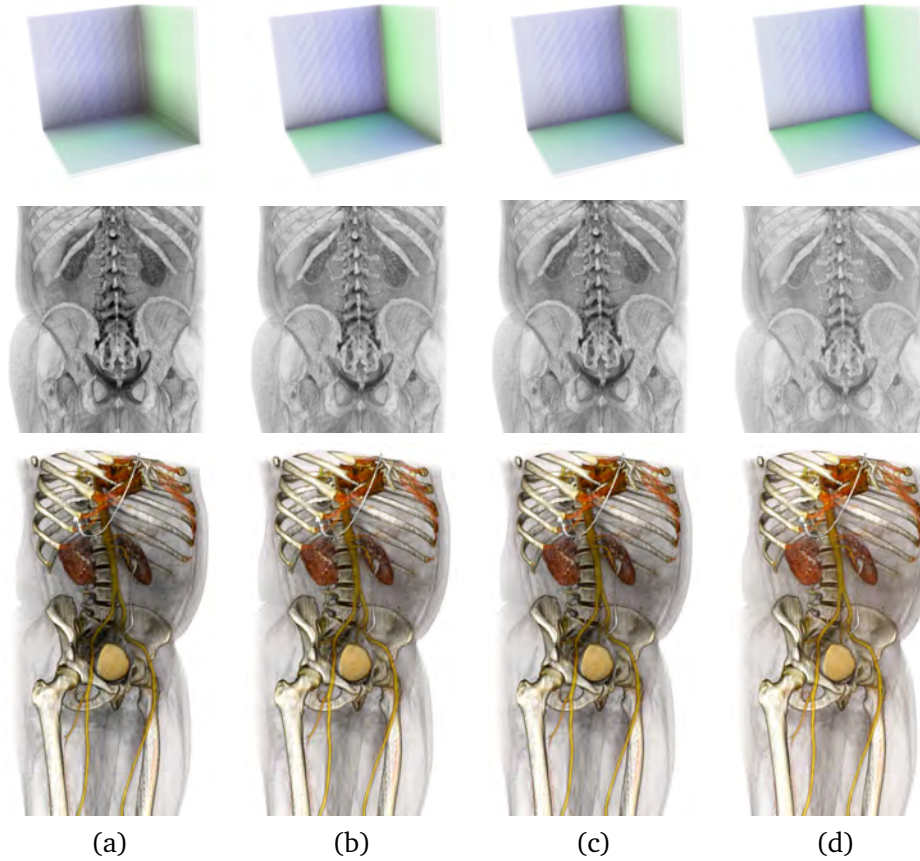


Figure 5.4: From top to bottom, obscurances with color bleeding for a synthetic model, obscurances for the CT-human body data set, and a rendering of this data set using an obscurance-based illumination model, all of them obtained considering different ρ functions. From left to right: (a) $\rho(d) = 0$, (b) $\rho(d) = \frac{d}{d_{\max}}$, (c) $\rho(d) = 1 - e^{-\frac{d}{d_{\max}}}$ and (d) $\rho(d) = \sqrt{\frac{d}{d_{\max}}}$.

least 42 directions for a good quality final image. We have used the high quality obscurances given by 162 directions for the rest of the images shown in this paper. While the obscurances volume of the CT-body has been computed from its opacity (given by the transfer function), the obscurances of the aneurism have been computed from its intensity values. The time cost for computing obscurances for the CT-body and the aneurism is shown in Table 5.1. Times are given for system with an Intel Core 2 Quad Q6600 CPU at 2.4 GHz with 2 GiB of memory. Note that, in accordance with the algorithm of Section 5.3.1, the time cost is proportional to the number of voxels times the number of directions. In the worst case, where densities are found in decreasing order, all the voxels in a ray would be pushed to the stack, and then each one would be popped, giving a cost proportional to the number of voxels in a ray. Thus, the cost of the algorithm is independent of d_{\max} .

Data	Size	12	42	162
Aneurism	$256 \times 256 \times 256$	0:36	1:56	6:56
CT-body	$256 \times 256 \times 415$	1:14	4:09	15:02

Table 5.1: Time cost (minutes:seconds) for computing obscurances using 12, 42, and 162 viewing directions.

5.4 Volume saliency

The human visual system is able to reduce the amount of incoming visual data to a small but relevant amount of information for higher-level cognitive processing. Different computational models have been proposed to interpret the selective visual attention. The biologically-inspired model of bottom-up attention of Itti et al. [Itti 2001] permits us to understand our ability to interpret complex scenes in real time. The selection of a subset of available sensory information before further processing appears to be implemented in the form of a spatially circumscribed region of the visual field, called *focus of attention*, while some information outside the focus of attention is suppressed. This selection process is controlled by a *saliency map* which is a topographic representation of the instantaneous saliency of the visual scene and shows what humans find interesting in visual scenes.

Inspired by Itti’s work, Lee et al. [Lee 2005] introduced the concept of mesh saliency, a measure of regional importance for 3D meshes, computed using a center-surround mechanism that is able to identify regions that are different from their surrounding context. Mesh saliency is captured from surface curvatures and is used in graphics applications such as mesh simplification and viewpoint selection. Feixas et al. [Feixas 2009] defined a view-based saliency of a polygon as the average information-theoretic dissimilarity between this polygon and its neighbors. In the volume rendering field, Kim and Varshney [Kim 2006] presented a visual-saliency-based operator to enhance human perception of the volume data by guiding the viewer’s attention to selected regions. A definition of voxel saliency is not provided and it is assumed that a

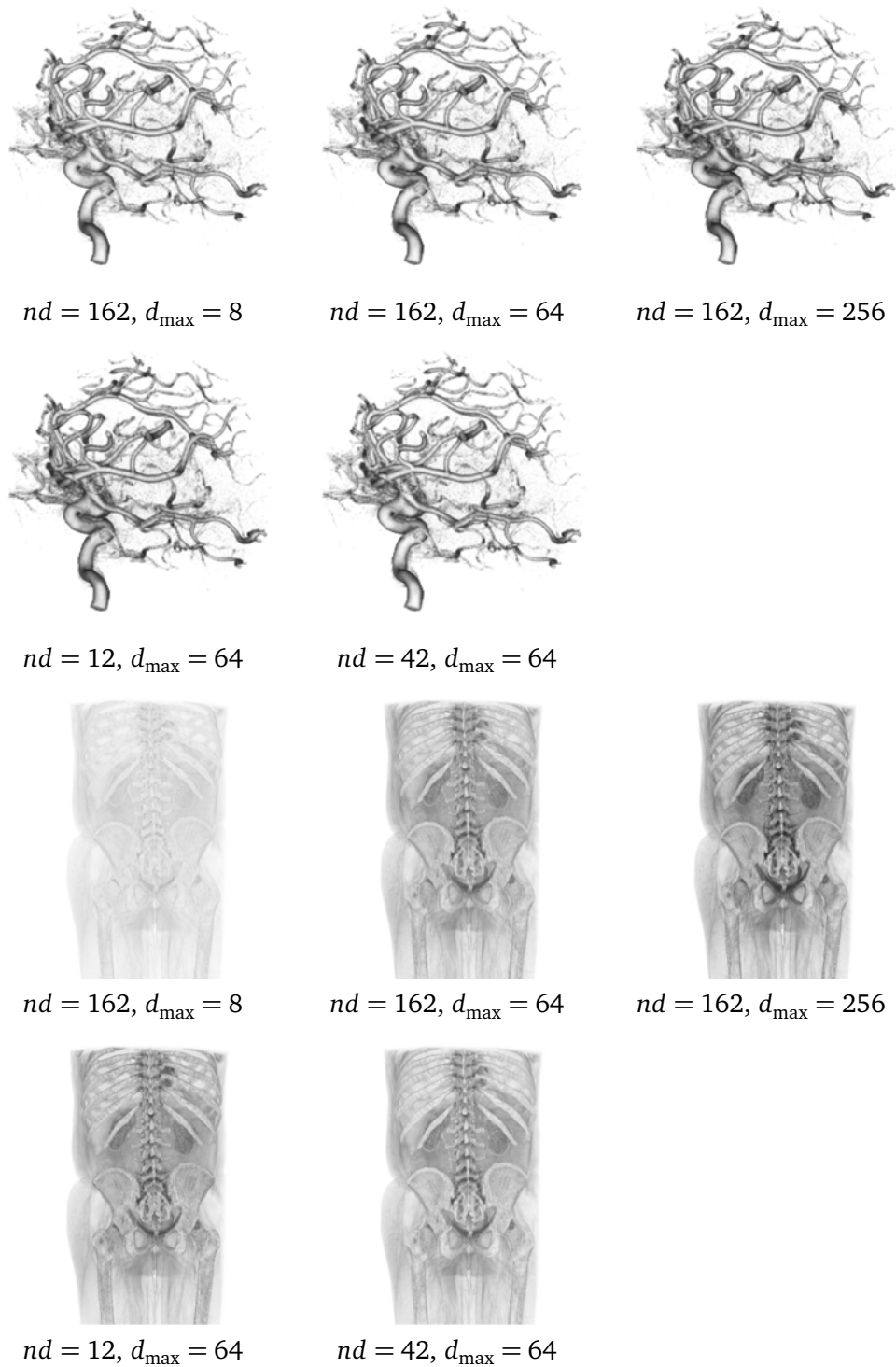


Figure 5.5: The obscuration volumes of the aneurism (first and second row) and CT-body (third and fourth row) are visualized considering different d_{\max} values and number of directions nd . The square root function has been used in all the cases. Computation times are given in Table 5.1.

saliency value is assigned to each voxel by using a user specification, eye-tracking data, or feature computation. In different works on saliency, it has been shown that attention is attracted by changes in luminance, color, curvature, texture, shape, etc. [Tsotsos 2005]. That is, salient features are generally determined from the local differential structure of images and different operators such as color or luminance gradient have been used [van de Weijer 2006]. In González et al. [González 2008], from an information theory perspective, ambient occlusion has been defined as the occlusion information associated with each polygon of the model.

In this chapter, a definition of voxel saliency based on the gradient of obscurance field is proposed. Considering that obscurance represents occlusion information associated with a voxel, its variation with respect to its surround can indeed be perceptually salient, i.e. it can be considered as a salient feature of the volume. This saliency would be most noticeable at edges, occlusion variations, and corners. On the other hand, a smooth or uniform region would produce low saliency values, as it is intuitively expected.

The voxel saliency is defined as the magnitude of the gradient of obscurances estimated by using the 4D linear regression method proposed in [Neumann 2000]:

$$S(z) = \sqrt{A^2 + B^2 + C^2}, \quad (5.4)$$

where voxel z is located at the origin of the coordinate system, and A , B , and C are the components of the obscurance gradient $[A, B, C]$. These components are computed as $A = \sum_k w(k) O(k) x(k)$, $B = \sum_k w(k) O(k) y(k)$, and $C = \sum_k w(k) O(k) z(k)$, where k stands for the voxels in the neighborhood centered at voxel z , $w(k)$ is the inverse of the distance between voxels z and k , $O(k)$ is the obscurance of voxel k , and $x(k)$, $y(k)$, and $z(k)$ are, respectively, the x , y , and z components of the vector from voxel z to voxel k . In our experiments, the neighborhood of z is given by a cube of 5^3 voxels, since smoother results are obtained than by using a cube of 3^3 voxels as in [Neumann 2000]. For each data volume, the saliency has been scaled ranging from 0 to 1. Analogously to mesh saliency [Lee 2005], the gradient of obscurances is scale-dependent (i.e., the saliency value depends on the size of the neighborhood considered). We have to emphasize that our definition of saliency can be generalized to the local volumetric shadowing methods [Hernell 2007, Ropinski 2008]. Figures 5.6(b), 5.6(d) and 5.6(f) show the color-coded saliency maps obtained for the CT-human body corresponding to Figure 5.6(a).

5.5 Applications

In this section, we describe how obscurances can be applied to volume rendering to interactively produce realistic and illustrative images. Two applications of saliency maps are also presented.

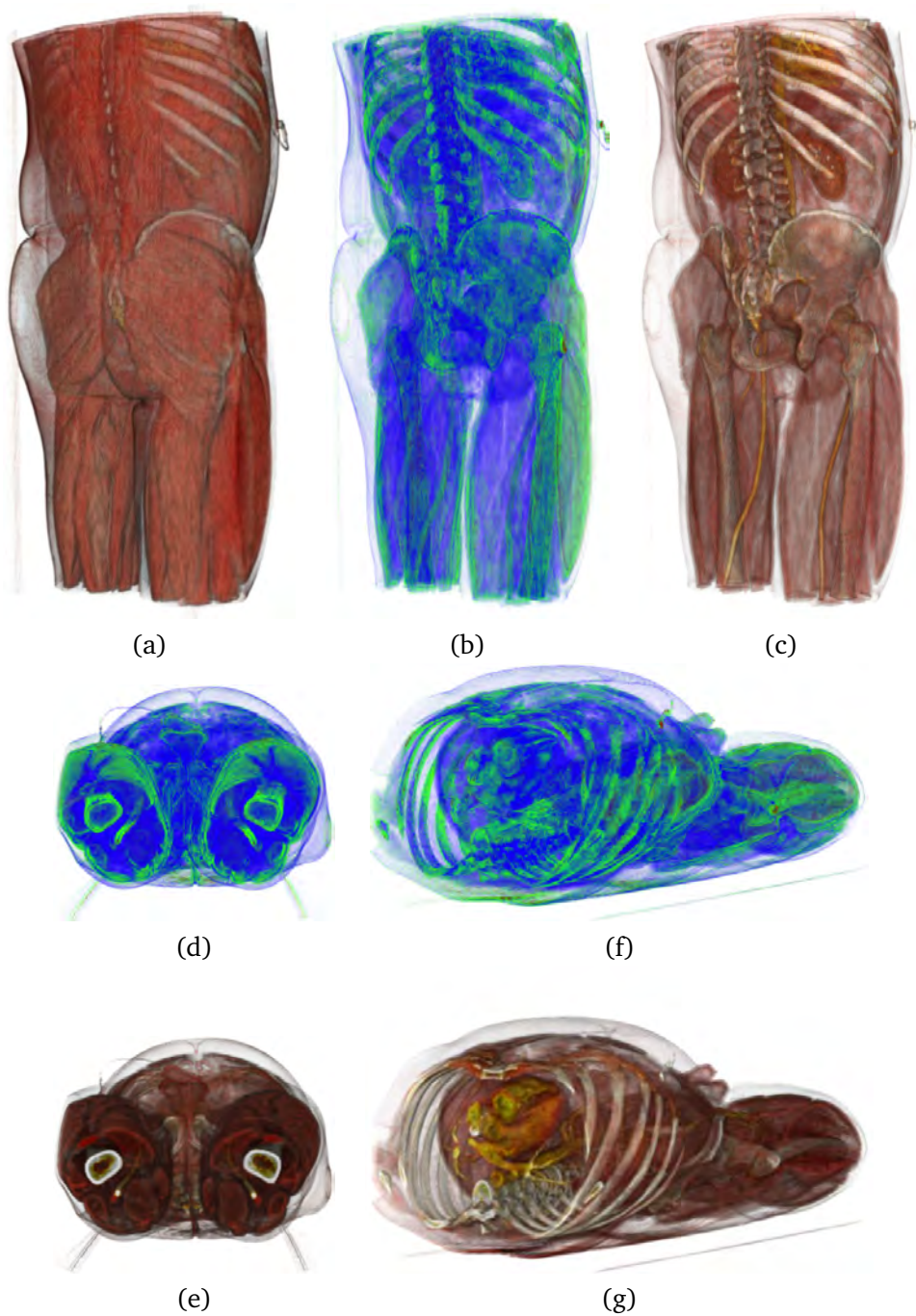


Figure 5.6: (a) Original CT-body data set. (b), (d), (f) Color-coded (from blue to red) saliency maps corresponding to (b) the most salient view, (d) the least salient view, and (f) the most salient view per unit area. (c), (e), (g) Illustrative visualizations obtained with a saliency-based opacity modulation.

5.5.1 Realistic and illustrative rendering

To apply the obscurances to the visualization, we use the Blinn-Phong shading model where the color resulting from the local lighting of each voxel z is multiplied by its obscurance value:

$$I(z) = (k_d N(z) \cdot L + k_s (N(z) \cdot H)^n) O(z), \quad (5.5)$$

where k_d and k_s are the diffuse and specular lighting coefficients, $N(z)$ the normal of the voxel, L the light vector, H the half-angle vector between L and the direction to the viewer, n is the Phong exponent, and $O(z)$ the obscurance of voxel z which has been adjusted to the range $[0, 1]$. Figure 5.4 (third row) illustrates the result of applying the obscurance-based Blinn-Phong model on the CT-body data set.

We also introduce two parameters, *low* and *high*, such that from 0 to *low* obscurances are set to 0 (making the voxel completely black), from *low* to *high* they follow a linear distribution (preserving their original value), and from *high* to 1 their value is set to 1 (thus the voxel becomes completely unobscured). Increasing the *low* threshold will turn more voxels black, and this can be used to increase the contrast of the resulting image. Decreasing the *high* threshold means that more voxels are not darkened by their obscurance, and so become brighter, thus increasing slightly the contrast too. In the limit we could set *low* and *high* to the same value to have only some voxels with obscurance 0 and others with obscurance 1. Thus, there would be voxels with their own color (modified only by local lighting if applied) and the others would be black. The user can modify *low* and *high* parameters to obtain the desired effect interactively. Figure 5.1 shows different renderings of the CT-body data set. Figure 5.1(a) shows the visualization of the model resulting from the application of obscurances with *low* = 0 and *high* = 1. Figure 5.1(b) has been obtained with *low* = 0.4 and *high* = 0.6, and making the skeleton transparent. Finally, in Figure 5.1(c) all the structures have been set to white and the obscurances assignment has been adjusted with *low* = 0.6 and *high* = 0.7.

5.5.2 Saliency

As a measure of importance, the volume saliency is applied to obtain the most salient views and to enhance volume visualization by modifying the transfer function according to the computed saliency.

Similar to [Lee 2005], where mesh saliency was used to select the best views, a method to calculate the saliency of a viewpoint is proposed. Given the saliency of all the voxels, we can find the viewpoint which maximizes the visible saliency. The *viewpoint saliency* is defined by

$$S(v) = \sum_{z \in \mathcal{Z}} S(z) \text{vis}(z|v), \quad (5.6)$$

where v is a given viewpoint, \mathcal{Z} is the set of voxels of the volume data, $S(z)$ is the saliency of voxel z , and $\text{vis}(z|v)$ is the visibility of voxel z from v .

We also present an automated technique to enhance volume visualization by emphasizing (increasing the opacity of) the most salient voxels and de-emphasizing (reducing the opacity of) the least salient ones. So, the viewer's attention is guided towards the most salient parts of the model.

In Figure 5.6, the original CT-body data set (5.6(a)), the most salient view (5.6(b), 5.6(c)), the least salient view (5.6(d), 5.6(e)), and the most salient view per unit area (5.6(f), 5.6(g)) are shown. Images 5.6(c), 5.6(e), and 5.6(g) have been obtained by multiplying the opacity by the saliency. Figure 5.7 shows the original CT-body data set (5.7(a)) and two different renderings obtained by scaling the opacity according to the saliency values (5.7(b), 5.7(c)). In Figure 5.7(b), voxels with saliency lower than 0.2 have been made transparent and the opacity of the most salient ones has been preserved. In Figure 5.7(c), voxels with saliency lower than 0.2 have been made transparent while the opacity of the most salient ones has been doubled.

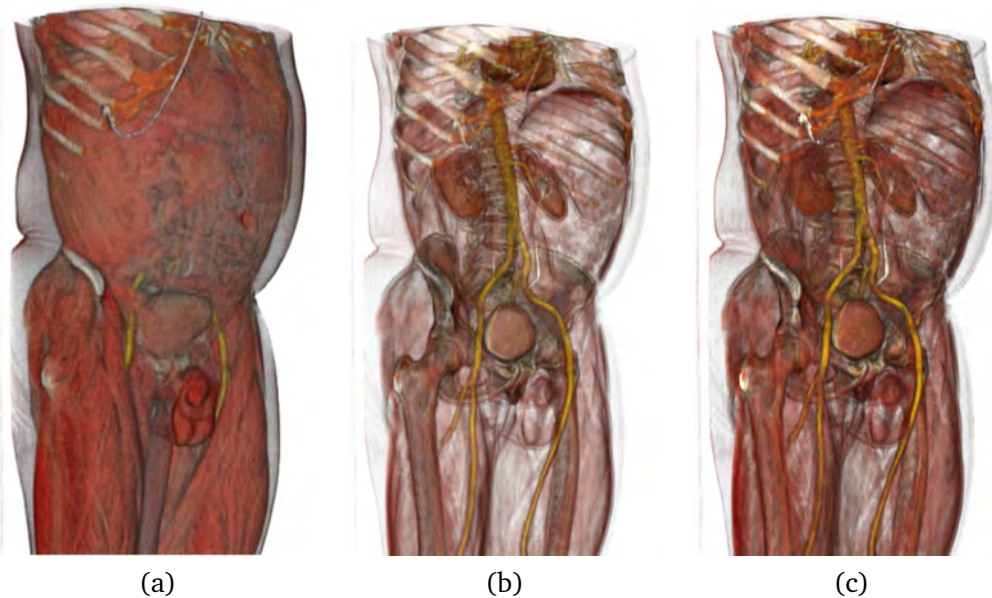


Figure 5.7: (a) Original CT-body data set. (b), (c) Images obtained by scaling the opacity according to the saliency values.

5.6 Conclusions

In this chapter, we have analyzed obscure-based volume rendering by evaluating the main parameters involved in its computation, such as the obscure function and the number of viewing directions. From this study, we conclude that the square root function gives better results than other analyzed functions and that 42 directions are enough to obtain obscures of a certain quality, although for high quality results we have used 162 directions. In addition, we have introduced two new applications of obscures. The first is a technique to obtain illustrative renderings and the second is a

method to compute the saliency map as the gradient of obscurances. Saliency has been used to enhance visualization and to select the most salient views. All our proposals have been integrated in a common framework and tested on several volume data sets.

As future work, we plan to program a GPU version of the obscurances algorithm to obtain real time or interactive obscurance computation.

Volumetric ambient occlusion for volumetric models

Contents

6.1 Introduction	81
6.2 Background	82
6.2.1 Obscurances and ambient occlusion	82
6.2.2 Volume shadowing	83
6.3 New approaches for volumetric ambient occlusion	83
6.3.1 Ambient occlusion as CDF	84
6.3.2 Reconstruction of the cumulative distribution	85
6.3.3 Gaussian cumulative distribution	86
6.4 Results	86
6.4.1 Ambient occlusion maps	87
6.4.2 Opacity modulation	90
6.5 Conclusions	93

6.1 Introduction

Obscurances or ambient occlusion (AO) [Zhukov 1998, Iones 2003, Landis 2002] has been proved to be an efficient technique that gives perceptually similar results to global illumination with a small fraction of the computational cost. The obscurance/ambient occlusion technique was first used in video-game environments. Its application to volume rendering, called vicinity shading, was introduced by Stewart [Stewart 2003]. In Chapter 5 of this thesis we have presented another volumetric adaptation of obscurances that overcomes some of the limitations in Stewart’s approach.

This chapter presents the volumetric ambient occlusion for volumetric datasets. The proposed approach is based on the volumetric ambient occlusion model [Szirmay-Kalos 2010, Penner 2008], where ambient occlusion is interpreted as how big portion of the tangent sphere of a surface belongs to the set of occluded points. The integrand of the new formula has low variation, thus can be estimated accurately with a few samples. The algorithm can effectively be used in real time systems and games to cheaply approximate global illumination effects. In our approach we extend this interpretation

to compute the ambient occlusion of a voxel. The proposed approach runs in interactive time on current GPUs due to the use of separable filtering. It does not require any pre-processing and thus can be applied to general dynamic models, providing high quality smooth shading.

6.2 Background

In this section we describe the obscurance and related illumination models. We also review previous work on volumetric ambient occlusion.

6.2.1 Obscurances and ambient occlusion

As we have seen in Section 2.5.2, the obscurance O of a point p is defined as the integral

$$O(p) = \frac{1}{\pi} \int_{\Omega} \rho(d(p, \omega)) \cos \theta \, d\omega, \quad (6.1)$$

where ρ is a function of the distance $d(p, \omega)$ of the first intersection of a ray shot from point p with direction ω , p is a surface point, θ is the angle between the normal vector at p and direction ω , and the integration is over the hemisphere oriented according to the surface normal. Ambient occlusion

$$A(p) = \frac{1}{\pi} \int_{\Omega} V(p, \omega) \cos \theta \, d\omega, \quad (6.2)$$

substitutes the ρ function in the obscurances equation (2.7) by the visibility function $V(p, \omega)$ that has value 0 when no geometry is visible in direction ω and 1 otherwise.

The evaluation of the directional integral in the previous equations ((6.1) and (6.2)) requires rays to be traced in many directions, which is rather costly and needs complex GPU shaders. Under the hypothesis of at most a single intersection of a traced ray within the radius d_{\max} and taking $\rho = d^3$, the directional integral can be transformed to a volumetric one (Equation (6.3)) and the expensive ray tracing operation is substituted by a simple containment test [Szirmay-Kalos 2010]. The domain of integration, the hemisphere above the surface, becomes now the tangent sphere.

Thus, the volumetric ambient occlusion is the relative volume of the unoccluded part of the tangent sphere S . Formally, the volumetric ambient occlusion function is defined as:

$$VAO(p) = \frac{\int_S \mathcal{I}(q) \, dq}{|S|}, \quad (6.3)$$

where $\mathcal{I}(q)$ is 1 if point q in the tangent sphere is in the unoccluded region and zero otherwise, and $|S| = \frac{4\left(\frac{d_{\max}}{2}\right)^3 \pi}{3}$ is the volume of the tangent sphere, which makes sure that the volumetric ambient occlusion is also in $[0, 1]$.

6.2.2 Volume shadowing

Several approaches to volume shadowing are reviewed in Section 2.5.1. In addition, Penner and Mitchell [Penner 2008] computed an approximation of ambient occlusion using the same formula as Eq. (6.3), where the domain of integration was the neighborhood around a surface point (they take both hemisphere and tangent sphere), based on the heuristic assumption that the percentage of occluding voxels surrounding the surface provides a good approximation to the percentage of rays that would be occluded while traversing through the same space. They used one-sided Chebyshev's inequality to compute them as *variance occlusion maps*, but stated in the conclusions that, although Chebyshev's inequality was a quick method to approximate complex lighting effects, better CDF (cumulative distribution function) approximations should be used.

This chapter is related to this approach and proposes more robust estimates for ambient occlusion.

6.3 New approaches for volumetric ambient occlusion

We propose two methods to approximate the ambient occlusion from the Cumulative Distribution Function (CDF) of the density values. The parameters of the estimation are computed by separable filters. Following the basic assumptions in [Szirmay-Kalos 2010], we will keep to the tangent sphere for a more robust interpretation of the results as Eq. (6.1).

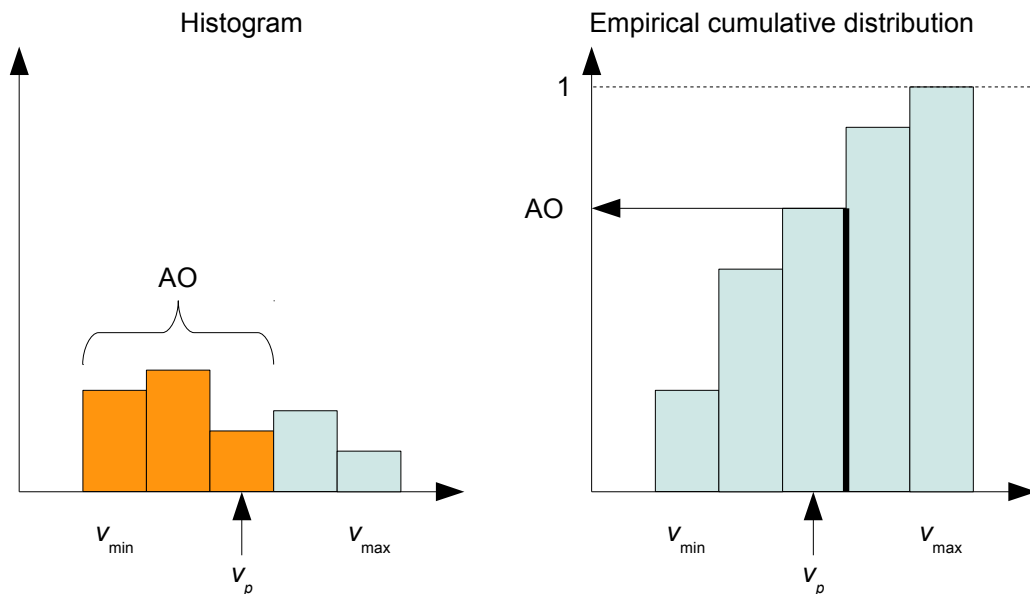


Figure 6.1: Ambient occlusion as the cumulative distribution function.

6.3.1 Ambient occlusion as CDF

The ratio of the volumes of the unoccluded points of the tangent sphere and the volume of the tangent sphere is the geometric probability that a uniformly distributed point q in the tangent sphere belongs to the unoccluded region, i.e. its scalar value $v(q)$ is smaller than the scalar value v_p of the shaded point p or the isosurface. If the scalar field is defined by a voxel grid, then we have exact values at voxel centers, but have no information about the function $v(q)$ in between the voxel centers. Note that the usual assumption on tri-linearly varying function is just an approximation. Instead, we can also assume that the scalar value $v(q)$ is a random variable whose distribution is defined by the N voxel values v_1, \dots, v_N included in the tangent sphere. With these, the geometric probability of the occlusion is

$$VAO(p) = \Pr[v \leq v_p] = \frac{\sum_{i=1}^N \varepsilon(v_p - v_i)}{N}, \quad (6.4)$$

where $\varepsilon(x)$ is the step function. Note that $\Pr[v \leq v_p] = F(v_p)$ is the cumulative probability distribution of random variable v for value v_p (Fig. 6.1).

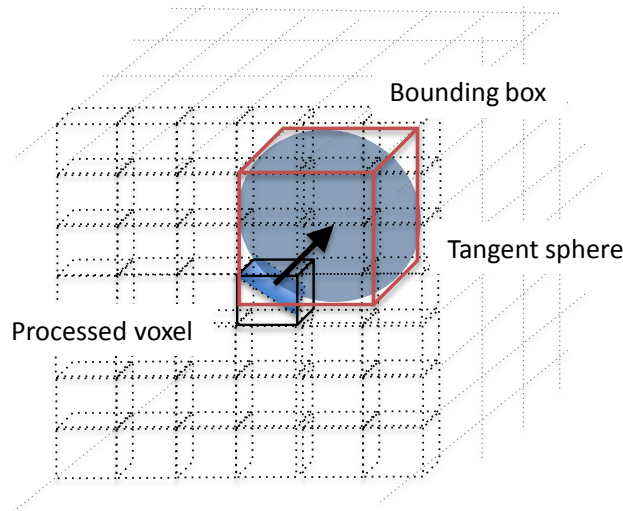


Figure 6.2: The ambient occlusion of a voxel is approximated by computing the probability that the density of a voxel in the bounding box of the tangent sphere is greater than the density of the processed voxel.

Unfortunately, the application of this formula requires N fetches from the voxel array, which is time-consuming unless N is small. However, selecting a small tangent sphere that contains just a few voxel centers degrades the quality of ambient occlusion and replaces nice soft shadows by smaller hard ones. On the other hand, when the set of voxels contained by the tangent sphere changes, there is an abrupt change in the ambient occlusion values, which results in stripe artifacts. So it is highly desirable to find a continuous approximation of the probability distribution, which is defined by just

a few representative variables [Gruen 2008]. Keeping real time rendering in mind even for dynamically evolving scalar fields, we should find these parameters to allow their computation with separable filtering, so the computation cost of a single representative variable is just $O(\sqrt[3]{N})$ instead of $O(N)$. To make separable filtering possible, we extend the domain from the tangent sphere to its bounding box (see Fig. 6.2).

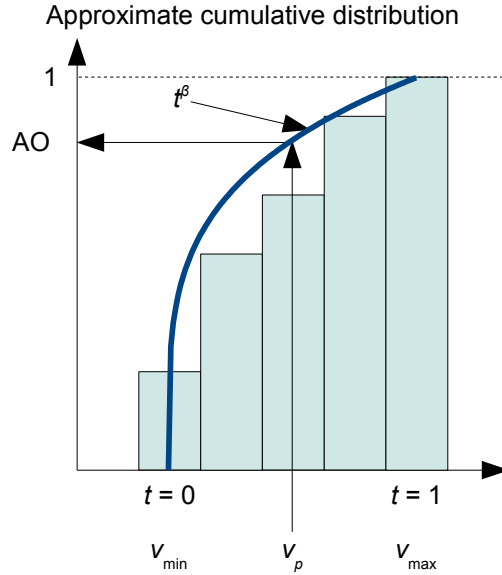


Figure 6.3: Approximation of the cumulative distribution function with t^β .

6.3.2 Reconstruction of the cumulative distribution

In this approach we reconstruct the cumulative distribution from three quantities that can be obtained by separable filtering inside the bounding box of the tangent sphere: the minimum density value v_{\min} , the maximum density value v_{\max} and the mean of the density values $E[v]$.

The yet unknown cumulative distribution $F(v_p)$ must be zero if $v_p \leq v_{\min}$, equal to 1 if $v_p \geq v_{\max}$, and non-decreasing in between. For notational simplicity, we introduce the following normalized parameter:

$$t = \frac{v_p - v_{\min}}{v_{\max} - v_{\min}} = \frac{v_p - v_{\min}}{\Delta v}, \quad (6.5)$$

where $\Delta v = v_{\max} - v_{\min}$.

Using the normalized parameter, the cumulative distribution must be zero if $t \leq 0$, and 1 if $t \geq 1$, and non-decreasing in interval $[0, 1]$. Considering all possible distributions of the density values, the cumulative distribution may be a step function at v_{\min} at one extreme case and another step function at v_{\max} in the other extreme case. Thus, we have to find a function that increases from 0 to 1 and has the flexibility to adapt to the two extreme cases. A very simple function meeting all these requirements is t^β

where β is the parameter of data fitting (Fig. 6.3). Thus the cumulative distribution is

$$F(v_p) = t^\beta \quad \text{where} \quad v_p(t) = t \Delta v + v_{\min}. \quad (6.6)$$

Let us consider the constraint on the mean:

$$\begin{aligned} E[v] &= \int_{v_{\min}}^{v_{\max}} v_p dF = \int_0^1 v_p(t) \frac{dF}{dt} dt \\ &= \int_0^1 (t \Delta v + v_{\min}) \beta t^{\beta-1} dt = \frac{\beta \Delta v}{\beta + 1} + v_{\min}. \end{aligned} \quad (6.7)$$

Solving this equation for β , we get:

$$\beta = \frac{E[v] - v_{\min}}{v_{\max} - E[v]}. \quad (6.8)$$

6.3.3 Gaussian cumulative distribution

Our second approach is to approximate the distribution by a Gaussian, and thus use the Gaussian CDF (Fig. 6.4):

$$\begin{aligned} F(x; \mu, \sigma^2) &= \int_{-\infty}^x f(t; \mu, \sigma^2) dt = \Phi\left(\frac{x - \mu}{\sigma}\right) \\ &= \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right) \right], \quad x \in \mathbb{R}, \end{aligned} \quad (6.9)$$

where $\mu = E[x]$, $\sigma^2 = E[x^2] - E^2[x]$, and $\operatorname{erf}(x)$ is the *Gauss error function*, which can be approximated by

$$\operatorname{erf}(x) \approx \sqrt{1 - \exp\left(-x^2 \frac{\frac{4}{\pi} + ax^2}{1 + ax^2}\right)}, \quad (6.10)$$

where

$$a = \frac{8(\pi - 3)}{3\pi(4 - \pi)} \approx 0.140012; \quad (6.11)$$

this approximation is better than $4 \cdot 10^{-4}$ in relative precision [Winitzki 2008].

In this case we compute by separable filtering the mean and the variance of the density values and use them in the previous equation to estimate the percentage of occluding voxels of a given voxel.

6.4 Results

The proposed methods have been implemented in CUDA and their performance has been measured on a system with an Intel Core 2 Quad Q9550 CPU and a Nvidia GeForce GTX 280 graphics card. To evaluate them we have used two different datasets,

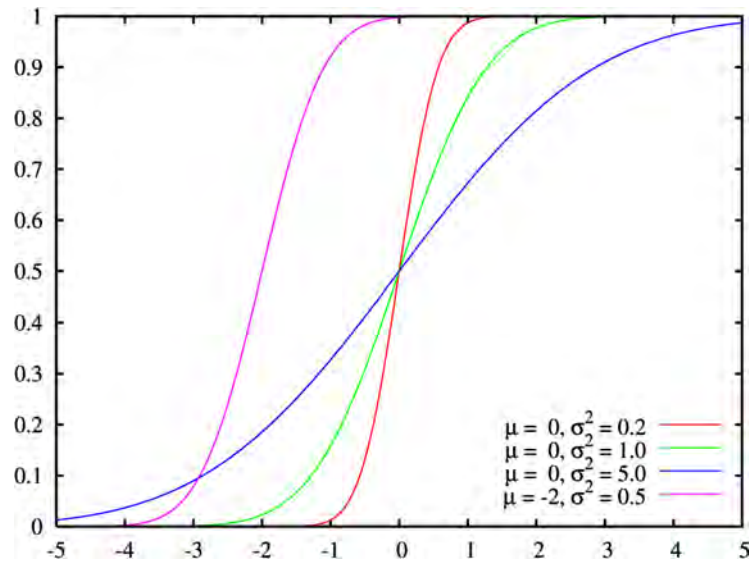


Figure 6.4: The Gaussian CDF with different parameters.

a CT-body of $256 \times 256 \times 415$ voxels from the Visible Human project and the Chapel Hill CT-head of $256 \times 256 \times 113$ voxels from the Volume Library.

6.4.1 Ambient occlusion maps

The proposed approaches have been compared with three different ambient occlusion strategies: the methods of Stewart [Stewart 2003], our obscurances (Chap. 5), and Penner [Penner 2008]. In Fig. 6.5 we present the obtained results. From Fig. 6.5(a) to Fig. 6.5(e), respectively, the maps have been obtained with the methods of Stewart, obscurances, Penner, the proposed CDF reconstruction, and the Gaussian CDF. In the case of Stewart and obscurances, the maps have been obtained considering 42 viewpoints. The other maps have been obtained taking a radius of 10 voxels. Observe that CDF reconstruction and Gaussian CDF methods generate lighter and more pleasant maps than the Penner method, and a smoother shading than the Stewart and obscurances maps.

One of the parameters that can be modified in our approaches is the radius of the filter kernel, that is taken equal to the diameter of the tangent sphere. To illustrate the effect of this value on the final images we present in Fig. 6.6 the maps obtained with different methods and radii. From row (i) to row (iii), the maps are generated with the methods of Penner, CDF reconstruction and Gaussian CDF. Columns (a) and (b) are obtained with a radius of 5 and 10 voxels, respectively. It can be seen that the quality of the maps is improved when the radius is increased. Also, we can see that CDF reconstruction and Gaussian CDF have a greater quality than Penner's approach. Finally, we consider the ambient occlusion value as an ambient lighting AL term. In this

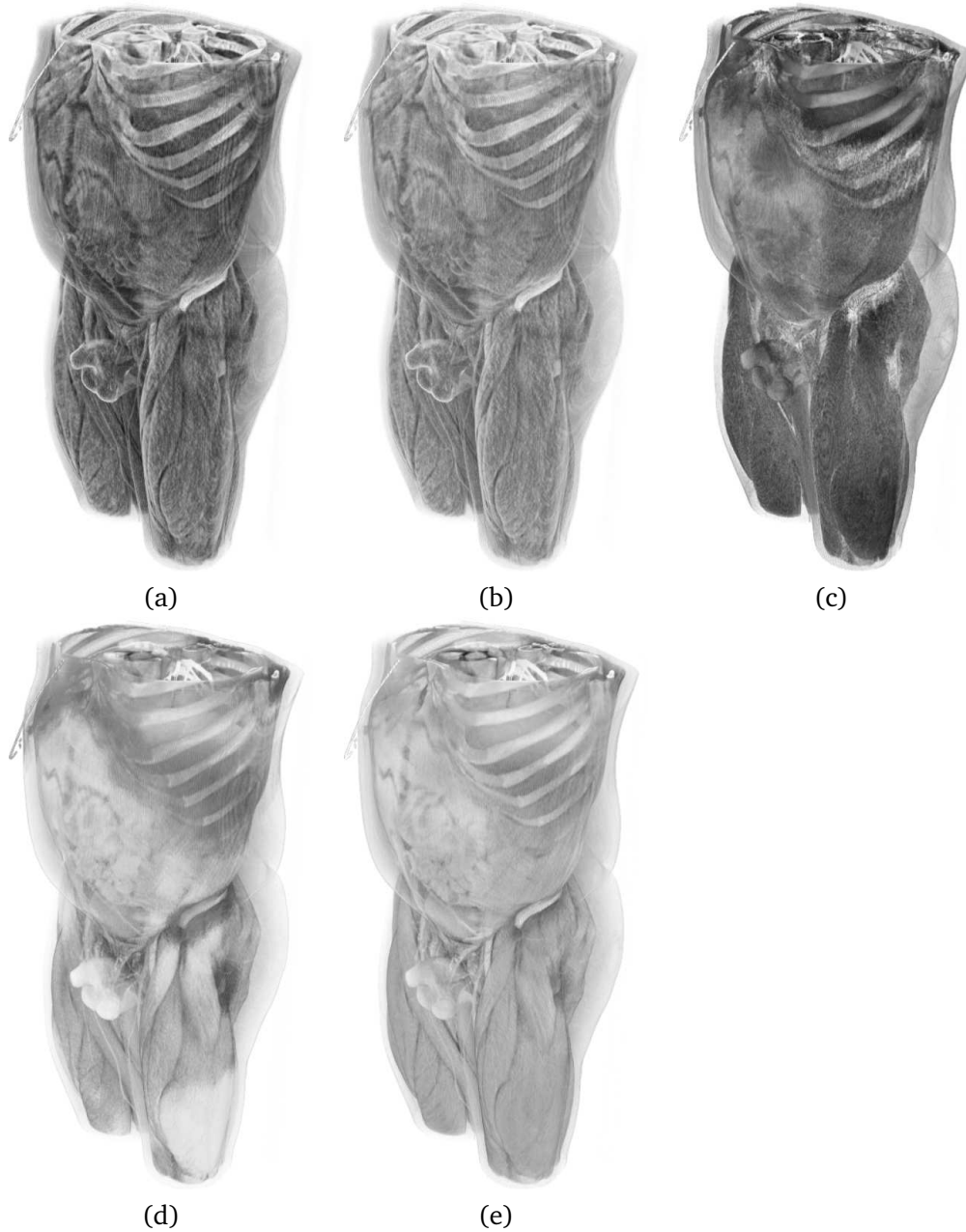


Figure 6.5: AO maps for CT-body using different methods: (a) Stewart, (b) obscurances, (c) Penner, (d) reconstruction of the CDF, and (e) Gaussian CDF. (a) and (b) have been computed with 42 viewpoints, and (c), (d), and (e) with a radius of 10 voxels.

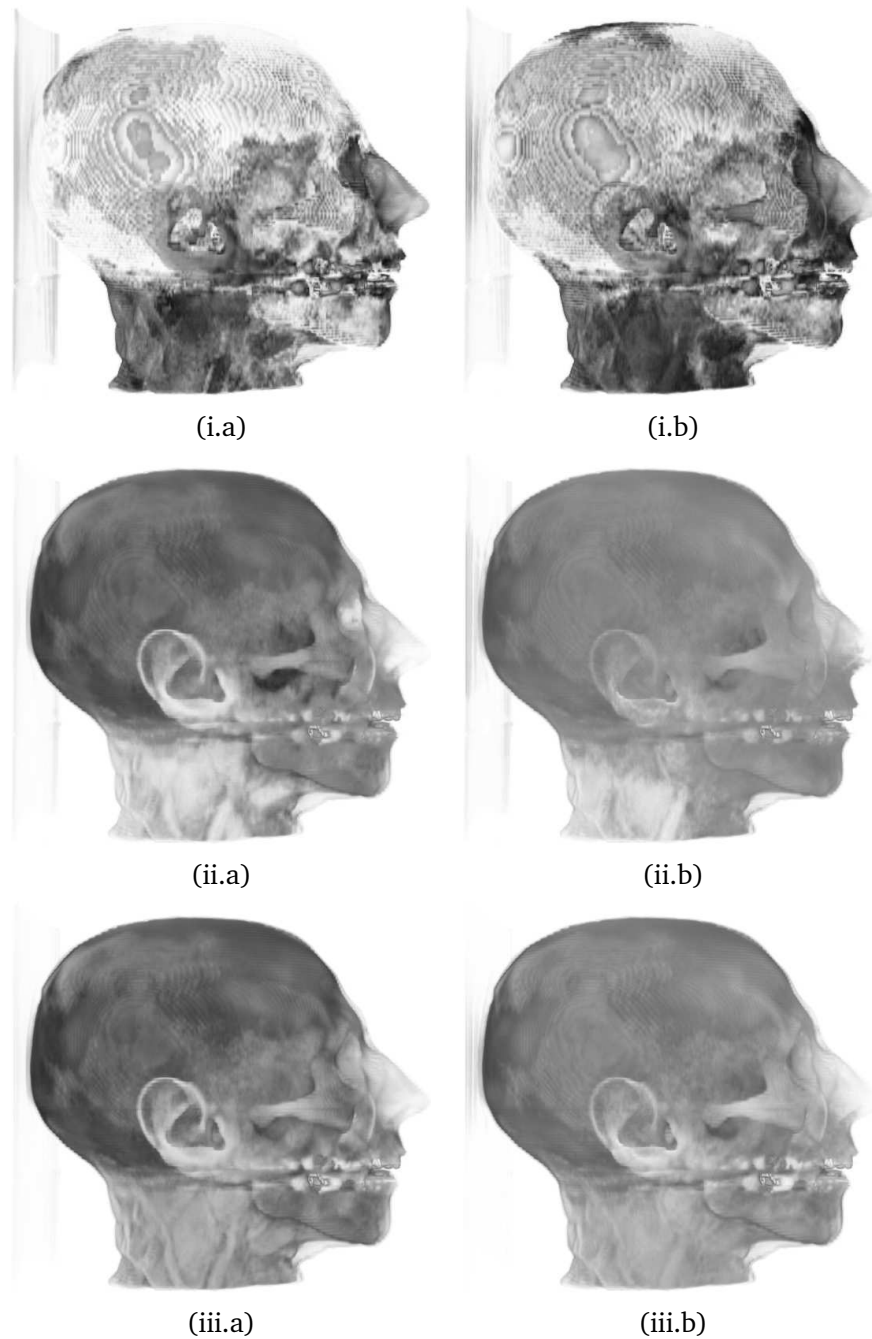


Figure 6.6: AO maps for CT-head using different methods and radii. Rows: (i) Penner, (ii) reconstruction of the CDF, and (iii) Gaussian CDF. Columns: (a) radius of 5 voxels, and (b) radius of 10 voxels.

	Radius	Penner	CDF reconstruction	Gaussian CDF
CT-body	5	713	565	717
	10	958	734	965
CT-head	5	200	161	203
	10	268	205	270

Table 6.1: Times in milliseconds to compute the AO maps for the different ambient occlusion methods and datasets and two radii.

case the color of a voxel p is obtained as

$$C(p) = AL(p) = k_i O(p) C_{TF}(p), \quad (6.12)$$

where k_i is a constant factor that modulates the intensity of ambient occlusion $O(p)$ and $C_{TF}(p)$ is the pure color of the voxel as defined in the transfer function. In Figures 6.7 and 6.8 we illustrate the effect that can be obtained with the CT-body and the CT-head, respectively. From (a) to (d), respectively, we show the model without ambient occlusion, and applying the AO maps obtained with the methods of Penner, CDF reconstruction, and Gaussian CDF, in all cases with a radius of 10. As expected, by adding the ambient occlusion maps, the overall features of the volume models become more distinguishable. With the CDF reconstruction and the Gaussian CDF methods, context information is better captured, giving an improved depth perception over Penner’s method, which generates a too dark final appearance. This is because, in the Chebyshev estimation used by Penner’s method, voxels with a value lower than the mean are assumed to be not occluded, and this creates a big contrast.

Table 6.1 shows the times in milliseconds to compute the volumetric AO maps for the CT-body and CT-head datasets, with radii 5 and 10, and for the three different methods. These times include all the filtering passes, memory transfers, and synchronizations.

6.4.2 Opacity modulation

As we proposed in Section 5.4, the gradient of the ambient occlusion indicates where the salient regions are, so it is worth modulating the opacity value by this gradient, which is estimated by a 4D regression filter [Neumann 2000]. The gradient calculation cannot be done using a separable filter, which increases the computational cost.

Instead of the gradient we propose the application of the variance of the intensity value, which also indicates where the volume changes quickly. The variance can be obtained as the difference of the second moment and the square of the mean value and both of them can be obtained by separable filtering. Thus modified opacity α' is:

$$\alpha' = \alpha \cdot \min \left(\frac{\sigma^2}{\sigma_{\max}^2}, 1 \right) = \alpha \cdot \min \left(\frac{E[v^2] - E^2[v]}{\sigma_{\max}^2}, 1 \right), \quad (6.13)$$

where σ_{\max}^2 is a user-defined constant.

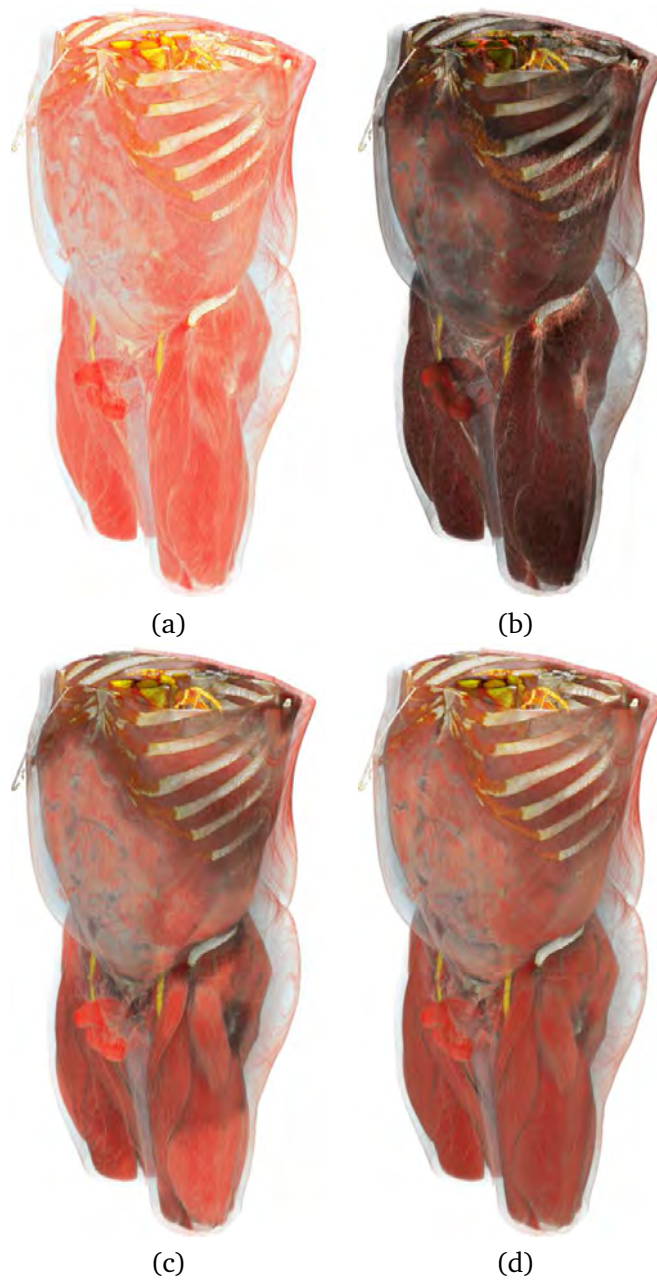


Figure 6.7: Ambient lighting for CT-body using different methods and a radius of 10 voxels: (a) original without ambient occlusion, (b) Penner's, (c) reconstruction of the CDF, and (d) Gaussian CDF.

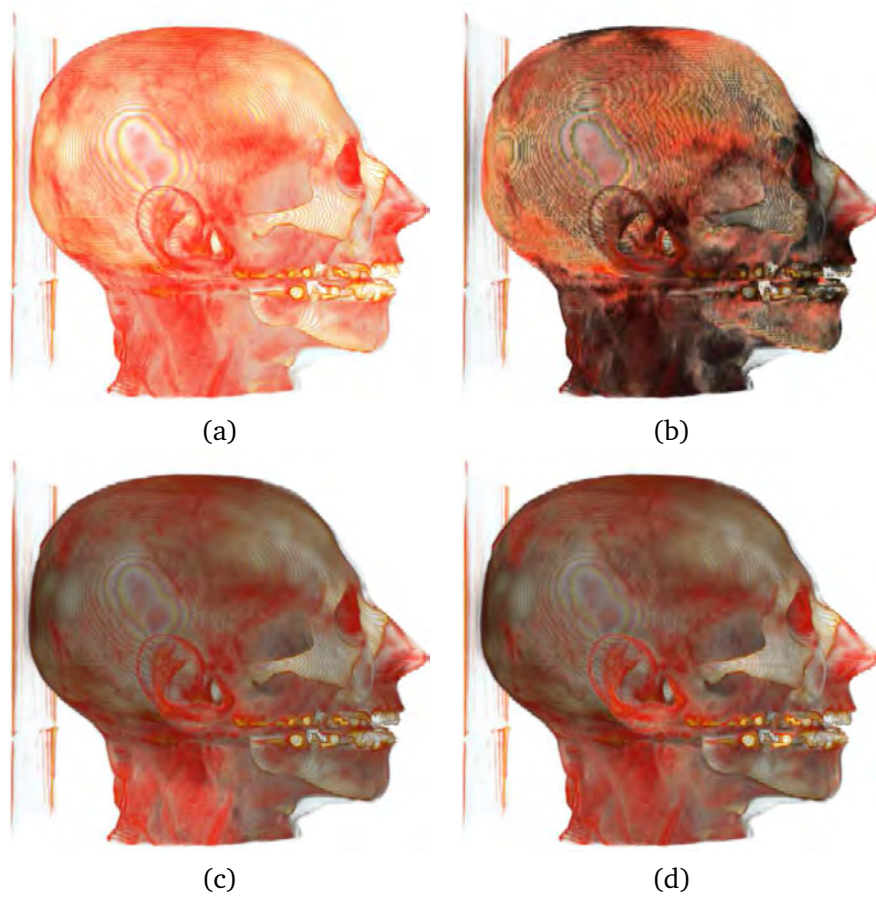


Figure 6.8: Ambient lighting for CT-head using different methods and a radius of 10 voxels: (a) original without ambient occlusion, (b) Penner's, (c) reconstruction of the CDF, and (d) Gaussian CDF.

In Figures 6.9 and 6.10 we present the opacity modulation effects that can be obtained using different methods. From column (a) to (e), respectively, we show the original model rendered with diffuse lighting, and modulated using the gradient of the AO map obtained with the methods of Penner, CDF reconstruction, and Gaussian CDF, and finally, with the intensity variance using a radius of 10 voxels and $\sigma_{\max}^2 = 5$. The result of this modulation is that the opacity of less occluded parts, such as muscles, is decreased, allowing to better see the inner structures. The time for the intensity variance computation with a radius of 10 voxels is 863 ms for the CT-body and 243 ms for the CT-head.

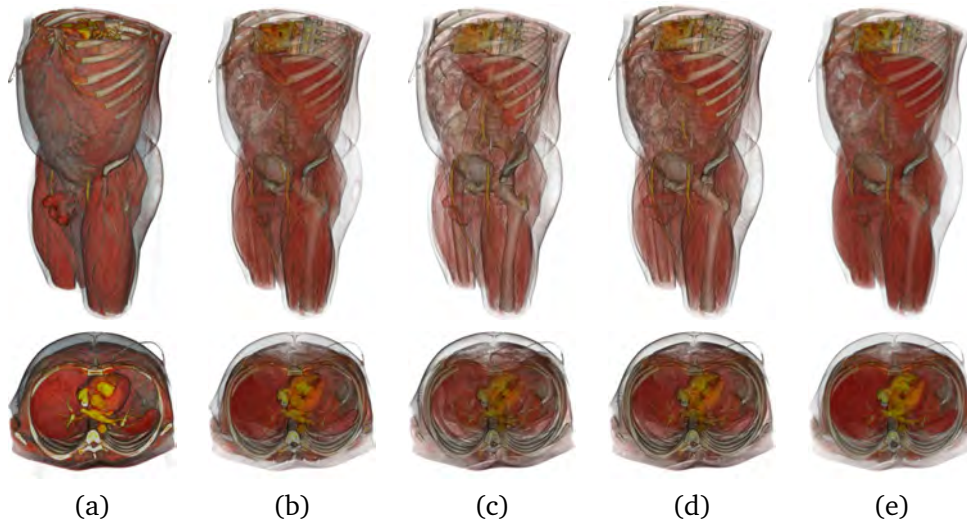


Figure 6.9: Opacity modulation of CT-body using different methods: (a) original rendering with diffuse lighting, (b) gradient of Penner's AO map, (c) gradient of CDF reconstruction AO map, (d) gradient of Gaussian CDF AO map, and (e) intensity variance with a radius of 10 voxels.

6.5 Conclusions

Volumetric ambient occlusion is a powerful technique that computes ambient occlusion by considering the portion of tangent sphere that is unoccluded. We have presented here new methods for the evaluation of the volumetric ambient occlusion for volumetric models. They produce high quality smooth shading and work at interactive rates thanks to the use of separable filtering to compute the needed parameters, and compare well with other ambient occlusion techniques for volumetric data. We have also shown how the gradient of the volumetric ambient occlusion nicely modulates the transfer function, and finally we have given a quick way to simulate this gradient.

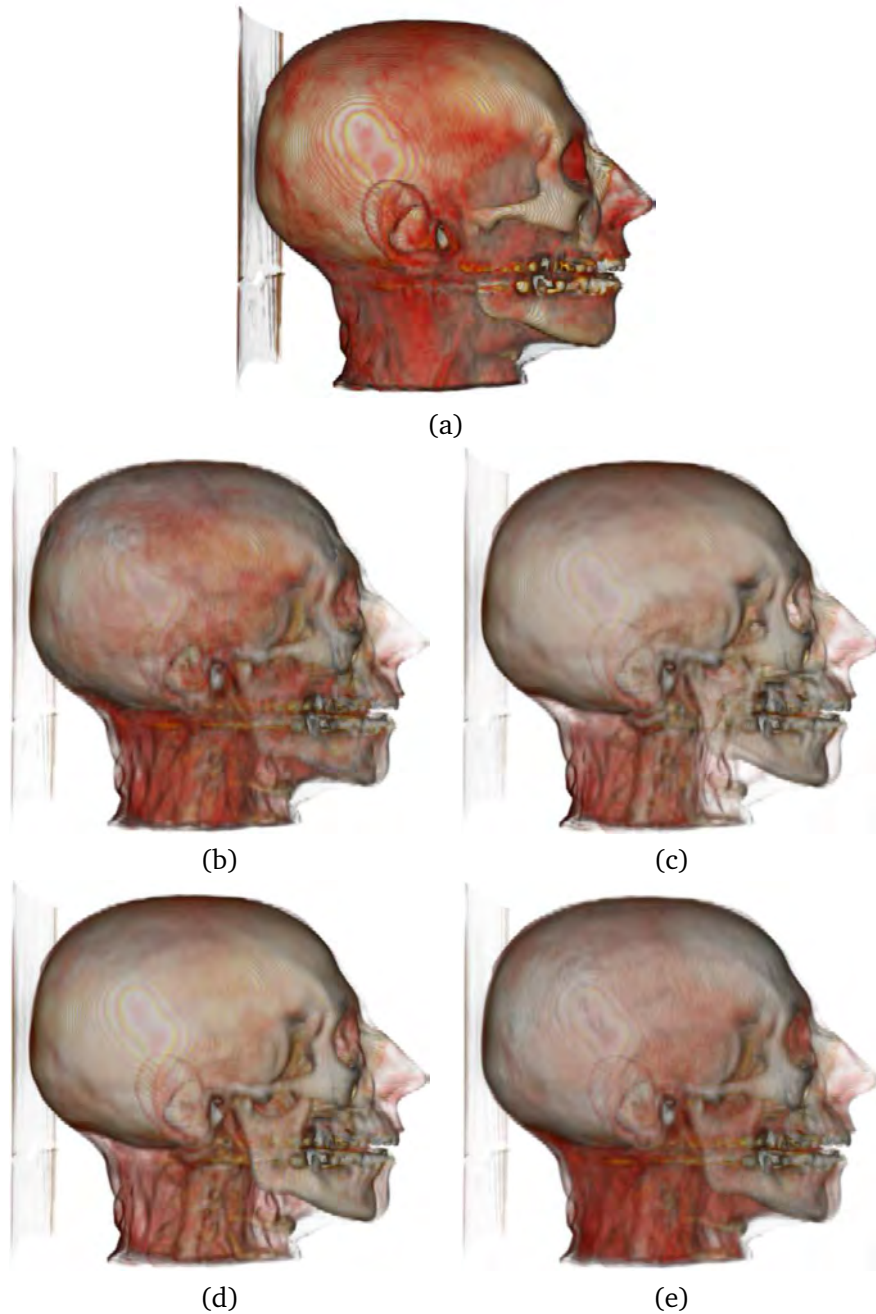


Figure 6.10: Opacity modulation of CT-head using different methods: (a) original rendering with diffuse lighting, (b) gradient of Penner's AO map, (c) gradient of CDF reconstruction AO map, (d) gradient of Gaussian CDF AO map, and (e) intensity variance.

Interactive volume illustration using intensity filtering

Contents

7.1 Introduction	95
7.2 Background	97
7.3 Proposed approach	97
7.3.1 Spatial importance map	97
7.3.2 Implementation	99
7.3.3 Modulation and illustration	103
7.4 Applications	103
7.4.1 Color modulation	103
7.4.2 Opacity modulation	105
7.4.3 Stippling	107
7.5 Conclusions	109

7.1 Introduction

A main step in direct volume rendering is the definition of the transfer function. This assigns optical properties such as color and opacity to the original values of the data to visualize the internal parts of the volume. In the case of one-dimensional transfer functions, there is a direct mapping between optical values and voxel scalar values. On the other hand, multidimensional transfer functions take more information into account, such as first and second derivatives [Kniss 2002a]. This additional gradient information allows a better separation between materials and thus better visualizations. The need for a good detection of boundaries becomes crucial for the quality of the rendering. Knowledge of boundaries facilitates the understanding of the volume data set by focusing on the most pertinent subset of data, i.e., which can be considered as the most salient parts of data [Itti 2001].

Although it has already been used by different authors [Kim 2006], the concept of saliency in a volume model, different to an image or a 3D polygonal scene, is not yet a well defined concept. Saliency should facilitate learning by focusing on the most pertinent subset of available sensory data. On the other hand, saliency of a voxel (or

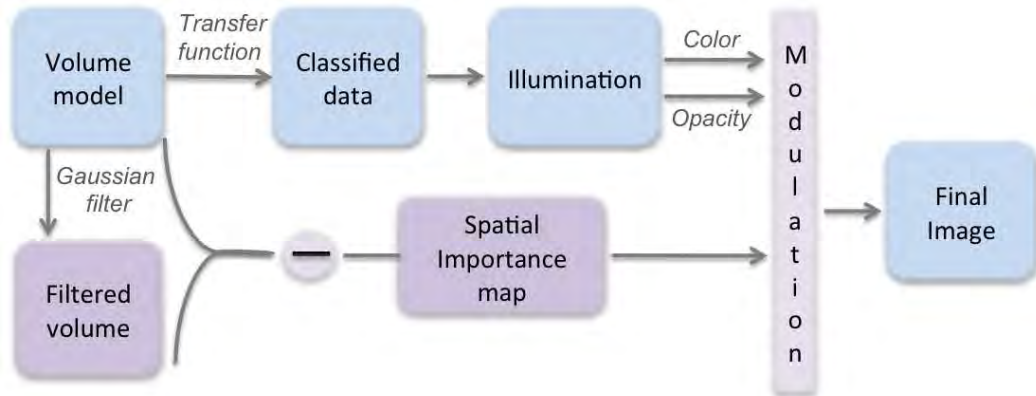


Figure 7.1: Main steps of the proposed approach.

region) should arise from contrast between this voxel (or region) and its neighborhood. Thus, it should include recognition of the boundaries of the inner parts of the volume data, as well as finding local deviations. When examining a 3D polygonal model, we just have one isosurface, we do not have to single it out, we just want to single out the particular oddities or irregularities of this surface. In the case of volume data, we first have to single out the components, i.e., the boundaries that make up its structure. On the other hand, the observation of the volume data model comes always via a transfer function, thus modulating an original, neutral transfer function with this extended saliency will allow us to learn about the data. A quantity very simple to compute that fills the above requirements of generalized saliency is an unsharped mask. This is the difference between low-pass filtered data and the original one.

Different feature enhancement strategies based on unsharp masking have been proposed. Cignoni et al. [Cignoni 2005] performed unsharp masking to the normal field over the 3D surfaces, to enhance the perception of discontinuous geometric features. Luft et al. [Luft 2006] enhanced depth perception by unsharp masking the depth buffer. The difference between the filtered image and the original one was called spatial importance map. Ritschel et al. [Ritschel 2008] coherently enhanced the scene by unsharp masking the outgoing radiance field over the mesh surface. Tao et al. [Tao 2009] use the difference between the radiance volume and the smooth radiance volume to enhance local contrast of features. In this section, we propose a similar approach but instead of using radiance we propose to operate with the original intensity values, having lower memory requirements. We use the spatial importance map to obtain both an enhanced visualization of the data, by modulating color and opacity, and illustrative effects such as stippling.

The structure of the chapter is as follows. In Section 7.2, we review related work. In Section 7.3, we describe the proposed approach. Then, in Section 7.4 we show the different effects that can be obtained in the proposed approach. Finally, conclusions are presented in Section 7.5.

7.2 Background

Direct volume rendering techniques allow to explore the structures embedded in the volume data set by varying the opacities and colors assigned to them. For the rendering to be effective it is required a transfer function that assigns colors and opacities to the different materials, but the definition of a proper transfer function is challenging. The main limitation is the classification process required to define the density intervals corresponding to the structures, i.e., the boundaries that separate the different materials. Several approaches have been presented to assist the user in transfer function design. Refer to Section 2.4 for an overview.

Another issue of special interest in volume rendering is how to determine the regions of interest of the rendered volume. Different computational models have been proposed to interpret the selective visual attention. Several works on volume saliency have been reviewed in Section 5.4. Considering that obscurance represents occlusion information associated with a voxel, its variation with respect to its surround can indeed be perceptually salient, i.e., it can be considered as a salient feature of the volume.

7.3 Proposed approach

Detection of the structures is an important step towards the interpretation of volume data. In general, each structure is represented by an interval of densities. The identification of these intervals can be obtained from the identification of the boundaries that separate from each other. Spatial importance maps can be used to identify these boundaries. The idea of our method is to exploit this fact to provide a simple strategy for exploring volume data models and also obtain different illustrative effects that enhance the perceptual quality and interpretation of the images taking only the original intensity values.

The main steps that compose the proposed approach are represented in Figure 7.1. First, in a preprocessing step, we compute the spatial importance map by filtering the input model and subtracting the obtained result from the original model. Then, the spatial importance map is integrated in the visualization pipeline and used to modulate the color and/or the opacities returned from the illumination process in order to produce different illustrative effects. A detailed description of these steps and main implementation details are given below.

7.3.1 Spatial importance map

The first step of our approach consists in the computation of the spatial importance map, that can be considered the core of the method. This map is the extension to 3D of the spatial importance function ΔD proposed by Luft et al. [Luft 2006]. In contrast to Tao et al. [Tao 2009], who used radiance, we use intensity values. The advantages of our approach are that it does not depend on the transfer function and lighting, and thus can be done as a preprocessing step, and that it requires less memory.

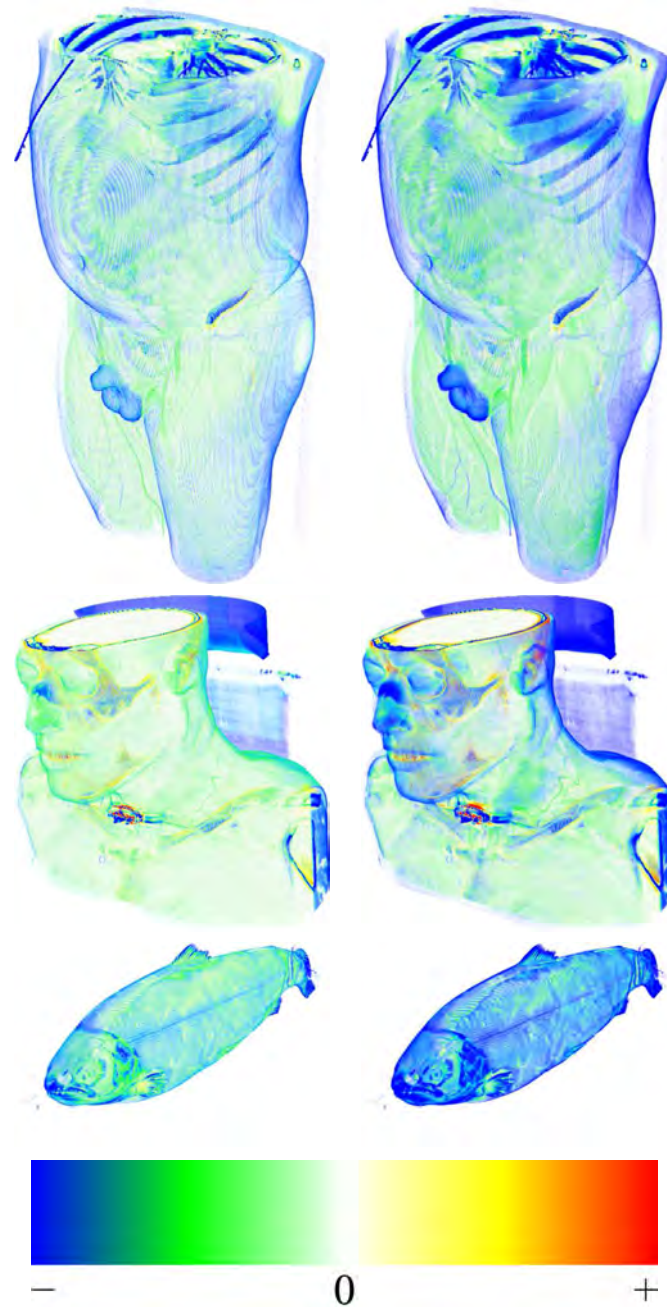


Figure 7.2: From top to bottom, spatial importance maps obtained for CT-body ($256 \times 256 \times 415$), CT-head ($512 \times 512 \times 297$), and CT-salmon ($336 \times 173 \times 511$). The images of the first and second columns have been obtained with a radius of 5 and 10 for the Gaussian filter, respectively.

Given a volume model $V : \mathbb{N}^3 \rightarrow \mathbb{Z}$ where \mathbb{Z} represents the scalar values of the voxels, we compute the spatial importance map ΔD by

$$\Delta D = G * V - V, \quad (7.1)$$

where $G * V$ is a Gaussian blur of the volume. A Gaussian blur is the convolution of an image with the Gaussian function

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad (7.2)$$

where x is the distance from the origin and σ is the standard deviation of the Gaussian distribution. The above function is for the 1D case, for 2D it is the product of two Gaussians and in 3D the product of three Gaussians, one per direction. The effect of a Gaussian blur is the reduction of the high frequency elements of the image, so it is a low-pass filter.

In Figure 7.2, we illustrate the spatial importance maps obtained for different models, a CT-body ($256 \times 256 \times 415$), a CT-head ($512 \times 512 \times 297$), and a CT-salmon ($336 \times 173 \times 511$) and different radii of the Gaussian kernel. The maps are colored using the thermal scale represented at the bottom, where warm colors correspond to positive importance values and cool colors to negative ones. Observe that the voxels that are near density boundaries have values different from zero, because in the blurred volume the boundaries are smoothed and have values lower than the originals at one side, and higher at the other side. So, positive difference values indicate voxels near a boundary with a higher density material since this makes that those voxels have higher intensity in the filtered volume. In a similar way, negative difference values correspond to voxels near a boundary with a lower density material because in the filtered volume their intensities are lowered. Note that both positive and negative values are important.

7.3.2 Implementation

The spatial importance map is the result of subtracting the original model from the filtered model. Therefore, the main step is the computation of the filtered volume model. To implement the Gaussian blur we take advantage of the separability property of the 3D Gaussian function. That is, the 3D Gaussian function can be separated into the product of three 1D functions. From a practical point of view, this means that convolving the volume with a 3D Gaussian function is equivalent to convolving the volume with a 1D Gaussian function along one axis, then convolving the result again with a 1D Gaussian function along another axis, and finally convolving this last result with a 1D Gaussian function along the other axis. This results in a computationally cheaper implementation.

Moreover, we have to take into account another feature of the Gaussian function in the implementation. The Gaussian function extends infinitely at both sides, so we would have to take into account the full image to compute the filtered version of each voxel. In practice, though, values further than 3σ are small enough to be negligible, so

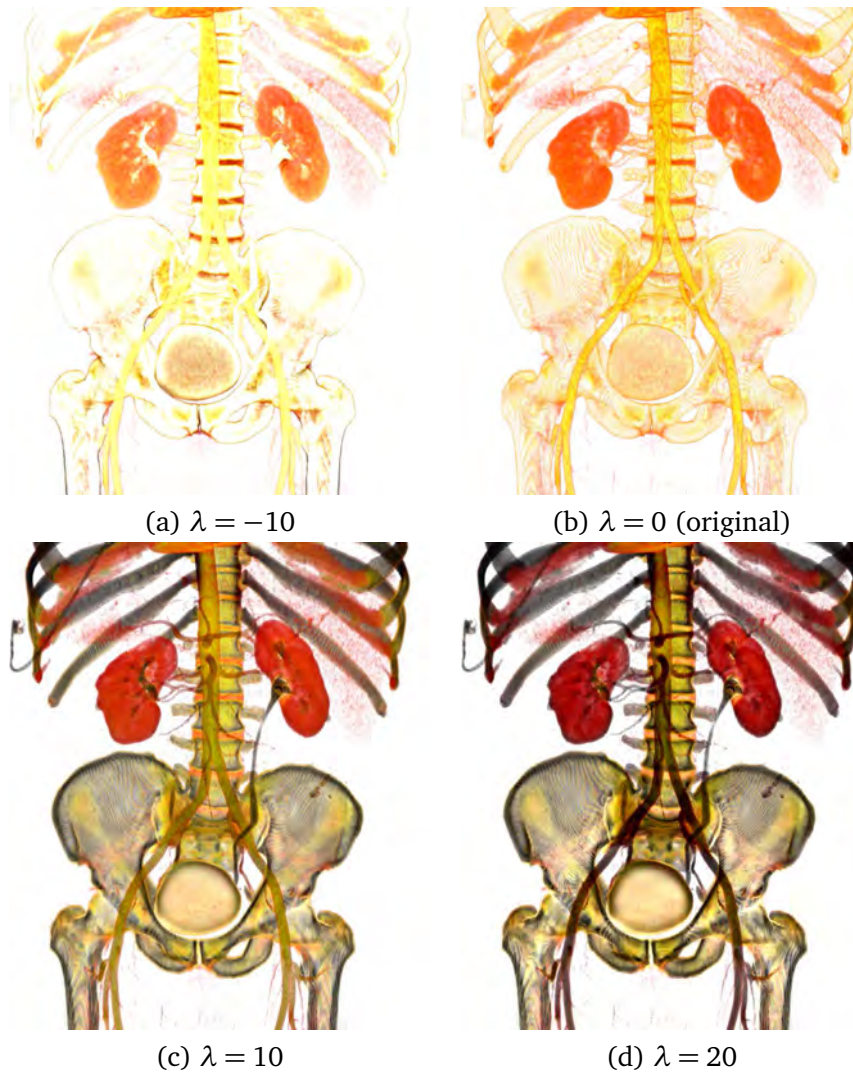


Figure 7.3: Color modulation driven by spatial importance map of the CT-body with ambient lighting.

Volume	R	1D filter (ms)	Subtract (ms)	Total (ms)
CT-body	1	20.85	18.03	335.05
	5	52.96		438.13
	10	93.59		560.23
CT-head	1	57.90	51.90	928.41
	5	155.61		1221.18
	10	272.77		1570.10
CT-salmon	1	22.66	19.66	376.37
	5	59.03		490.04
	10	104.93		617.24

Table 7.1: Times in milliseconds (ms) to compute the spatial importance map for the CT-body ($256 \times 256 \times 415$), the CT-head ($512 \times 512 \times 297$), and the CT-salmon ($336 \times 173 \times 511$) with different radii (R) of the Gaussian kernel. The *1D filter* column reports the time to do a 1D convolution with the kernel (this process has to be done once for each axis). The *Subtract* column reports the time required to do the subtraction. The *Total* column includes the time to do the 3 convolutions, the subtraction, and all the other additional operations like memory allocations and transfers.

we can clamp the function at that distance. In our implementation, we have only one parameter regulable by the user, which is the radius of the Gaussian kernel measured in pixels (see Figure 7.2). Then, we define σ to be one third of the radius, so that all the values inside the radius are significant. The values in the kernel are sampled from the Gaussian function at the center of each voxel, and then we normalize them so that they sum 1.

The method has been implemented with CUDA to achieve real time performance. We take advantage of the fact that both the filtering and the difference are parallelizable for each voxel. The steps of the CUDA implementation are the following:

1. Prepare volume data converting it to floating point values.
2. Copy volume data to a 3D CUDA array and bind a 3D texture to it. This texture returns the real values (not scaled), and is accessed with non-normalized texture coordinates with clamp address mode and with nearest neighbour interpolation, so that it works like an array.
3. Allocate space in global memory to store the result of the filtering.
4. Compute the Gaussian kernel according to the radius and copy it to global memory.
5. Filter the array data in the X axis.
6. Copy the result of the previous filtering to the volume 3D array so it is the source in the next step.
7. Filter the array data in the Y axis.

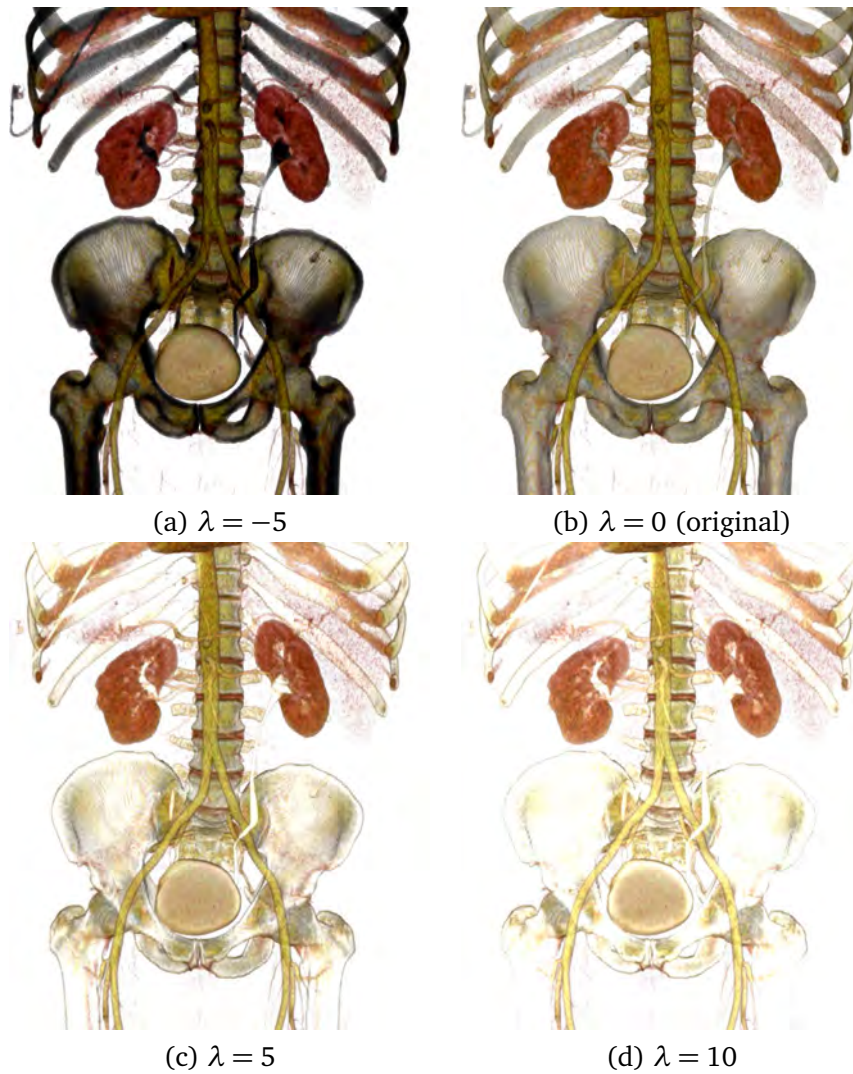


Figure 7.4: Color modulation driven by absolute spatial importance map of the CT-body with local lighting.

8. Copy the result of the previous filtering to the volume 3D array so it is the source in the next step.
9. Filter the array data in the Z axis.
10. Copy the original volume data again to the 3D array.
11. Subtract the original volume from the filtered volume (the last result).
12. Copy the final result to host memory.
13. Clean up all allocated memory in the graphics card.

In Table 7.1, we collect the time required to compute the spatial importance map for the CT-body, the CT-head, and the CT-salmon considering different radii of the Gaussian kernel. The *1D filter* column reports the time to do a 1D convolution with the kernel (this process has to be done once for each axis). The *Subtract* column reports the time required to do the subtraction. The *Total* column includes the time to do the 3 convolutions, the subtraction, and all the other additional operations like memory allocations and transfers previously described.

7.3.3 Modulation and illustration

As it is illustrated in Figure 7.1, once the spatial importance map has been obtained it is integrated at the end of the visualization pipeline to obtain different effects. The modulation process considers the map and also the colors and/or opacities assigned by the illumination module to the volume. The set of effects that can be obtained depends on the considered parameters. We can use the raw importance values or convert them to absolute values. In addition, we can consider applying the modulation to the color assigned to the voxel or to the opacity. Moreover, we can use it to regulate the density of the dots in a stipple rendering. In the next section, we give a detailed description of the most representative effects.

7.4 Applications

To describe the different effects that can be obtained with the proposed approach, we grouped them into three different sections. First, we consider the modulation of the illumination model (effects on the colors), then the modulation of the transfer function (effects on the opacities), and at last the modulation of dot density in stippling. In all the following examples, the radius of the Gaussian filter is 10.

7.4.1 Color modulation

As we have shown in Section 7.3.1, the spatial importance map contains information about the boundaries distributed in positive and negative values, which can be interpreted as both sides of boundaries. To emphasize them, importance values can be used

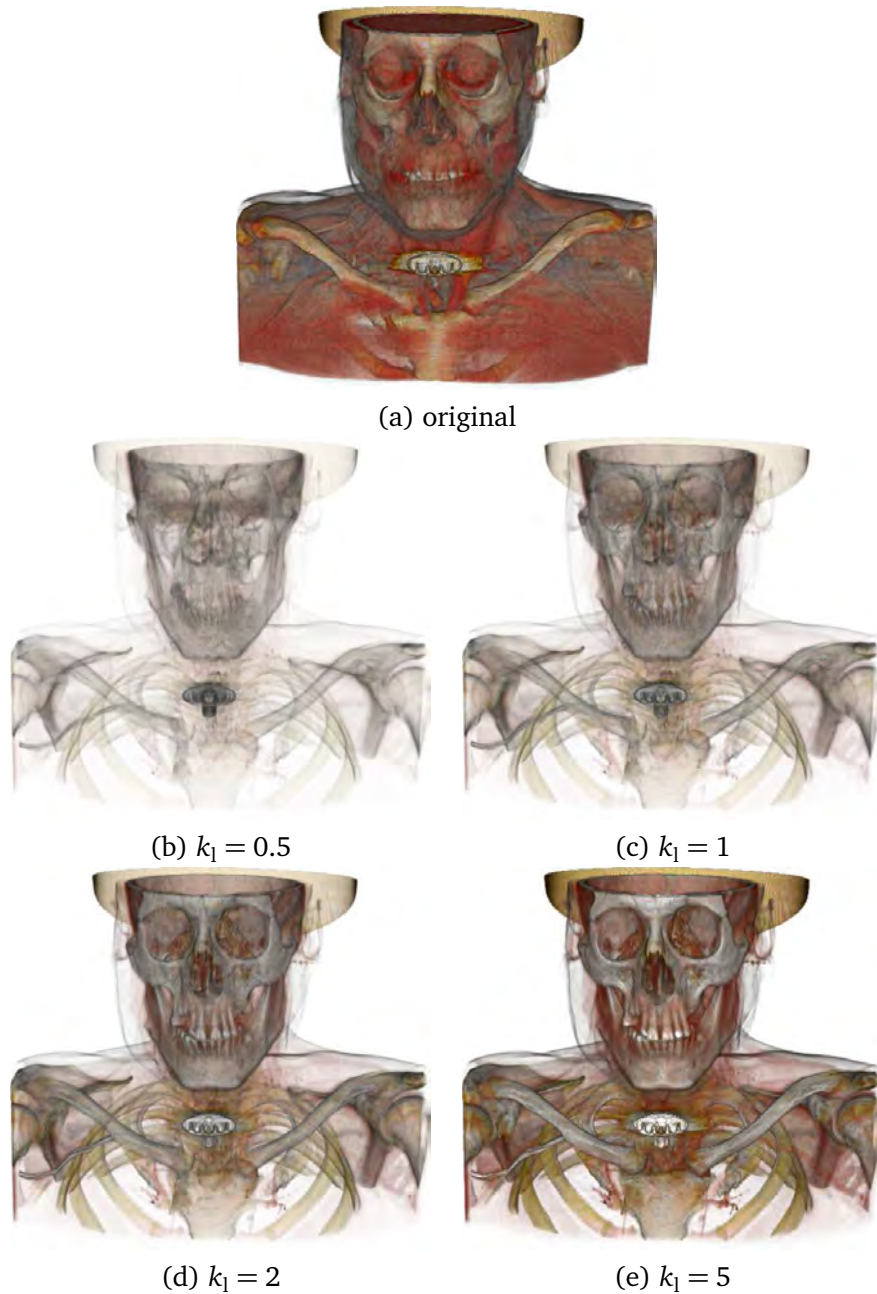


Figure 7.5: Different opacity modulation effects with different parameters applied to the CT-head model. (a) Original model, and (b) to (e) with $t_1 = 1$, $t_h = 1$, $k_h = 1$, and modifying the k_1 parameter.

to modulate the color and contrast of the volume dataset. We propose different strategies that differ in how we consider the values of the map for the modulation. Being I the color of a voxel, the first strategy for adding the spatial importance value ΔD is by considering the raw values. In this case, we apply

$$I' = I + \Delta D \cdot \lambda, \quad (7.3)$$

where λ is a factor used to modulate the effect.

The second effect we obtain is by taking absolute importance values instead of the raw ones. In this case, we obtain the final color with

$$I' = I + |\Delta D| \cdot \lambda. \quad (7.4)$$

All these effects can be applied both to the ambient lighting or the local illumination. In Figure 7.3, we show the effects obtained using original importance values and ambient lighting on the CT-body. From left to right, the λ values are -10 , 0 , 10 , and 20 . In Figure 7.4, we show the effects on the same model with local lighting and absolute importance values. From left to right, the λ values are -5 , 0 , 5 , and 10 . Observe that with raw values and a positive λ we obtain a darker image, due to the fact that most values in the map are negative; on the other hand, a negative λ produces a brighter image. With absolute values, the darkening and brightening effects are reversed because all the values are positive.

7.4.2 Opacity modulation

Now we describe how to modulate the opacity. In this case, the idea is to use the information of the spatial importance map to increase or decrease the opacity in order to emphasize the most salient parts.

Being $A(z)$ the opacity of the voxel z , we compute the new opacity $A'(z)$ by

$$A'(z) = \begin{cases} A(z) k_l \overline{|\Delta D(z)|}, & \text{if } \overline{|\Delta D(z)|} < t_l, \\ A(z) k_h \overline{|\Delta D(z)|}, & \text{if } \overline{|\Delta D(z)|} > t_h, \\ A(z), & \text{otherwise,} \end{cases} \quad (7.5)$$

where t_l and t_h are the low and high thresholds respectively, k_l and k_h are factors to regulate the effect of the modulation, and $\overline{|\Delta D(z)|}$ is the absolute spatial importance value of the voxel z normalized in the range $[0, 1]$. We use absolute values because we are interested in detecting the boundaries, and both positive and negative values give this information.

In Figure 7.5, we illustrate the opacity modulation effects applied to the CT-head. Column (a) corresponds to the original model without any modulation. Columns (b) to (e) are obtained with $t_l = 1$, $t_h = 1$, $k_h = 1$ and k_l set to 0.5 , 1 , 2 , 5 . From Equation (7.5) we can see that k_h has no influence on the rendering since $t_h = 1$. Furthermore, since $t_l = 1$, nearly all voxels are multiplied by the spatial importance and the

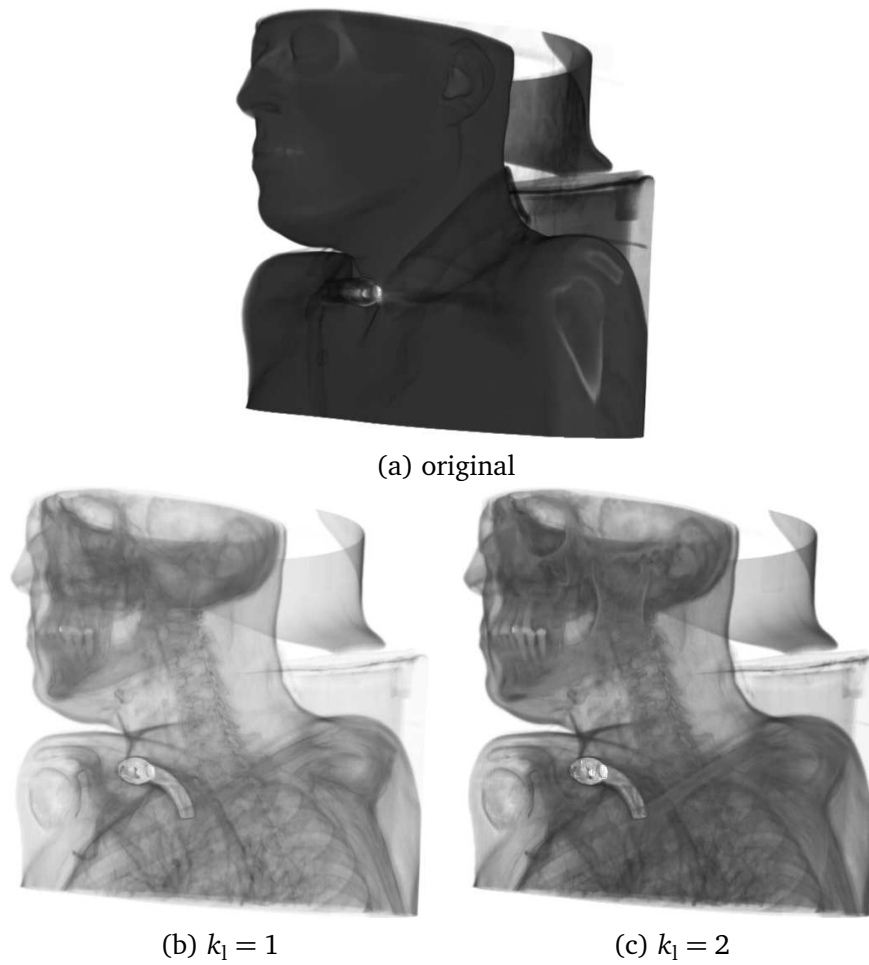


Figure 7.6: Different opacity modulation effects using a default transfer function applied to the CT-head model. (a) Original model, and (b) and (c) with $t_l = 1$, $t_h = 1$, $k_h = 1$ and modifying the k_l parameter.

weighting factor k_1 . As k_1 increases, more detailed information about inner structures is captured.

Figure 7.6 shows that the same previous effects can be applied to the original model considering a default transfer function that is obtained by linearly mapping intensities to opacities and gray values. Note that the proposed approach can be used for first explorations of a volume dataset since no a priori knowledge is required.

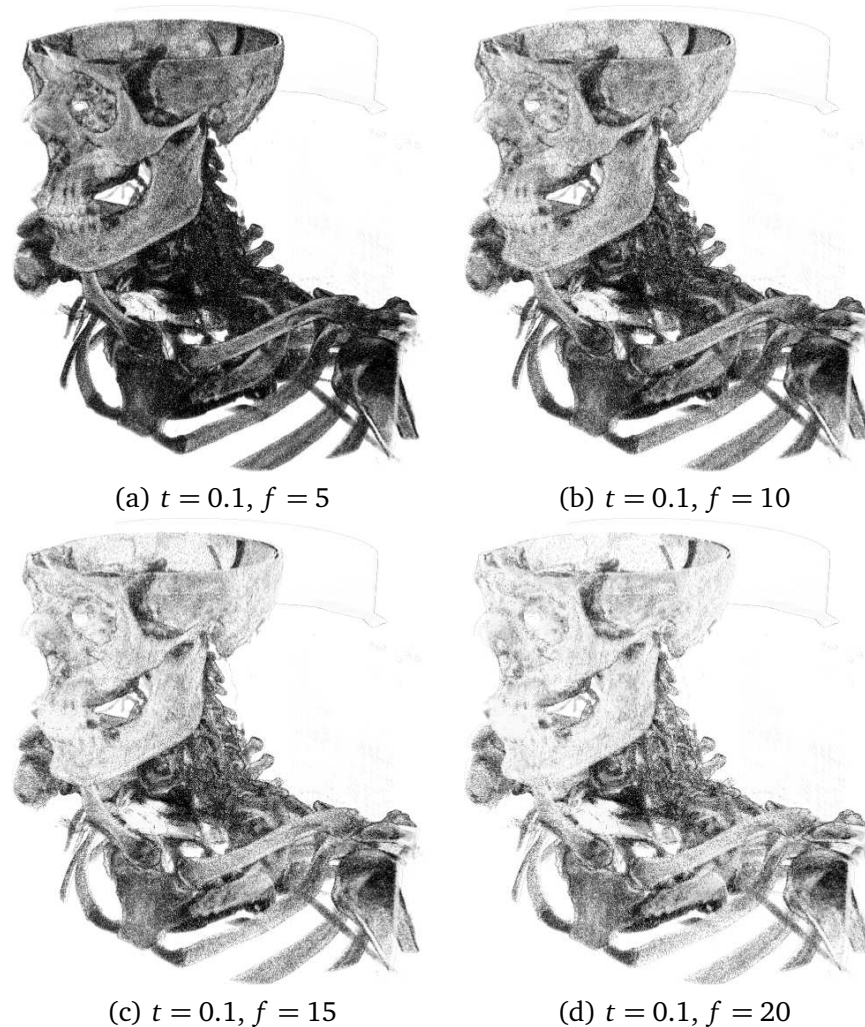


Figure 7.7: Stipple rendering of the CT-head modulated by the spatial importance map with different parameters.

7.4.3 Stippling

Stippling is an illustration technique in which the image is drawn using dots. This technique has been simulated algorithmically by several authors. Deussen et al. [Deussen 2000] applied half-toning techniques to arrive at an initial stipple distribution and then interactively applied relaxation based on centroidal Voronoi diagrams. Secord [Sec-

ord 2002] used a fast probabilistic method in which stipples are automatically packed more densely in dark regions and more sparsely in lighter regions. Schlechtweg et al. [Schlechtweg 2005] created a multi-agent system to position the stipples. Sousa et al. [Sousa 2003] approximate stippling by using short, serrated ink strokes modeled directly over the mesh's edge.

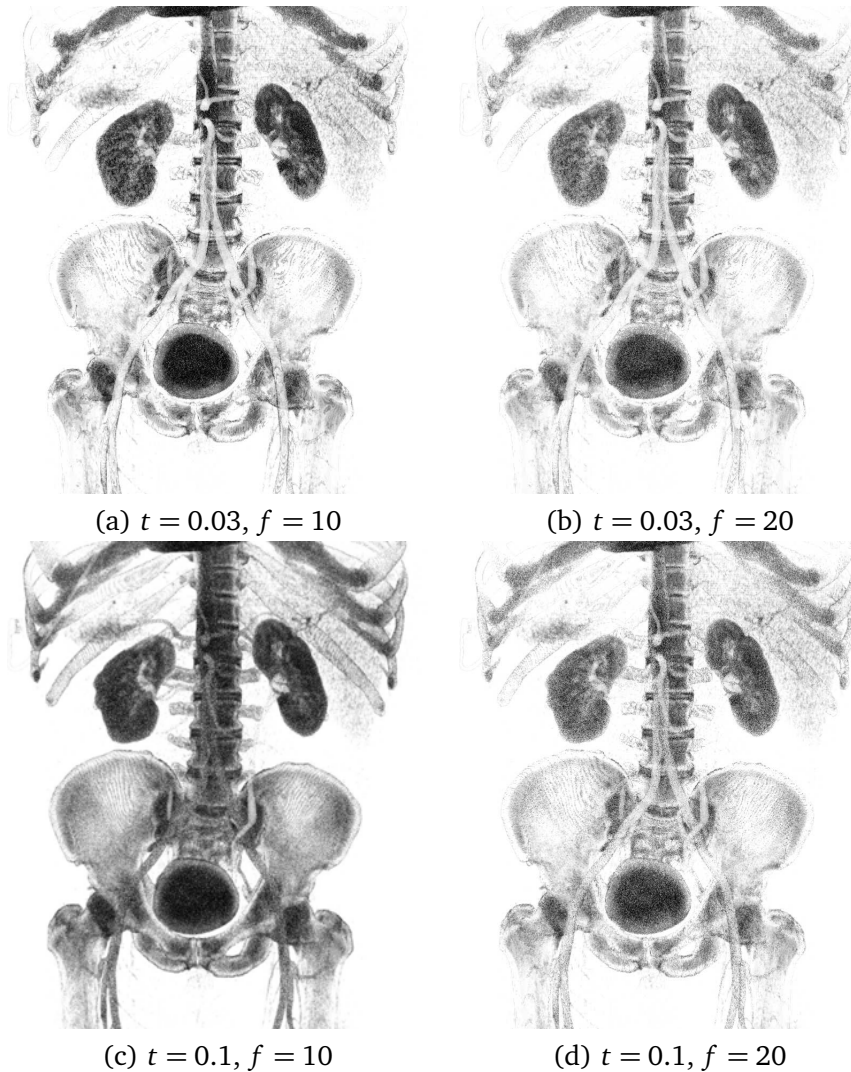


Figure 7.8: Stipple rendering of the CT-body modulated by the spatial importance map with different parameters.

In our approach, we propose to regulate the density of the dots according to the normalized absolute spatial importance map $|\Delta D|$. We have two user-defined parameters: a threshold t above which everything is white, and a factor f that regulates the density scale. In addition, there is a random value $r(s)$ generated for each sample s in

the ray casting. The color of each sample $C(s)$ is defined as

$$C(s) = \begin{cases} \text{white,} & \text{if } |\overline{\Delta D(s)}| \geq t \text{ or } f \cdot |\overline{\Delta D(s)}| \geq r(s), \\ \text{black,} & \text{if } |\overline{\Delta D(s)}| < t \text{ and } f \cdot |\overline{\Delta D(s)}| < r(s). \end{cases} \quad (7.6)$$

In Figures 7.7 and 7.8, this stippling effect has been used to render the CT-head and the CT-body, respectively, considering different parameters. This effect can also be applied in combination with the opacity modulation, as shown in Figure 7.9.



Figure 7.9: The CT-body rendered combining stippling ($t = 1$, $f = 1$) and opacity modulation ($t_l = 1$, $t_h = 1$, $k_l = 1$, $k_h = 1$) effects.

7.5 Conclusions

We have presented a new approach for obtaining illustrative volume renderings. The method computes a spatial importance map that captures information about the most salient parts of the model. This map is integrated in the visualization pipeline allowing to modulate color and opacity values. Such modulations have been used to obtain different effects that enhance volume data interpretation giving visual clues about struc-

tures contained in the volume. In addition, we have used the spatial importance map to modulate the density of the dots in a stipple rendering.

In our future work, we will explore the use of other low-pass filters, such as the trilateral filter, and evaluate how they behave compared to the Gaussian filter.

Automatic transfer functions based on informational divergence

Contents

8.1 Introduction	112
8.2 Background	113
8.2.1 Transfer function design	113
8.2.2 Information theory in computer graphics	113
8.3 Framework and motivation	113
8.3.1 Visibility channel	113
8.3.2 Motivation	115
8.4 Method	117
8.4.1 Target distribution	118
8.4.2 Objective function	120
8.4.3 Optimizer	121
8.5 Results	126
8.5.1 Limitations	133
8.6 Conclusions	133



Figure 8.1: Volume renderings of the tooth data set using transfer functions obtained with different target distributions. From left to right, the target distributions used are occurrence weighted by intensity, occurrence weighted by importance (1 for enamel and 0.5 for the rest), occurrence weighted by gradient, and occurrence weighted by importance using a mask of the nerve.

8.1 Introduction

A crucial step in volume rendering is the transfer function definition. This function assigns color and opacity to the data being visualized determining which structures of the volume will be visible and how they will be rendered. Transfer functions assume that volume values map directly to physical properties. The main issues involved in transfer function specification are volume data classification and management of visual properties.

As we have mentioned in Section 2.4, different strategies have been proposed to simplify the transfer function specification [Pfister 2001]. Classical approaches use scalar volume data to detect boundary information between tissues and then define the transfer function. Unfortunately, overlaps between data intervals corresponding to different materials make boundary detection difficult. To overcome this limitation, derived attributes, such as first and second order derivatives, are considered to isolate materials [Levoy 1988, Kindlmann 1998, Hladůvka 2000, Kindlmann 2003]. In this case, the transfer function definition becomes more complex and user interaction is required. Other techniques define the transfer function on the basis of rendered images and the user selects one or more favorite images that guide further image selection [He 1996, Marks 1997, Wu 2007]. The goal of these methods is to obtain good renderings but not necessarily good transfer functions. More recent strategies for transfer function design propose to consider other parameters, such as visibility [Correa 2009b, Correa 2011], or measures that can be derived from information theory [Haidacher 2008, Bruckner 2010]. Despite the advances of these methods, the automatic definition of the transfer function is still an open research problem.

In this chapter, we present an automatic approach for transfer function design that does not require previous knowledge nor segmentation of the model. Our method is defined from a visibility channel between a set of viewpoints and a set of bins of the volume data set, and is formulated as an optimization process that obtains the opacity transfer function that minimizes the informational divergence between the visibility distribution (normalized visibility histogram) captured by the set of viewpoints and a target distribution. This target distribution is proposed by the user and reflects the data importance, or highly relevant data value interval, or spatial segmentation. Different target distributions, based on scalar volume data or derived attributes such as the gradient, are proposed following two different strategies to obtain the transfer function: a global strategy that assumes no a priori knowledge of the model and an importance-based strategy that emphasizes regions or intensity bins selected by the user in the spirit of view-dependent cutaways. The performance of the method is analyzed for different volume data sets.

This chapter is organized as follows. In Section 8.2, we present related work on transfer function design and on information-theoretic applications to computer graphics. In Section 8.3, the framework and the motivation of the method are presented. In Section 8.4, the proposed method is described. Experimental results are shown and discussed in Section 8.5. Finally, our conclusions and future work are given in Section 8.6.

8.2 Background

In this section, we review different strategies that have been proposed in transfer function design. Since our approach is based on information-theoretic tools, we also present related work concerning information theory in computer graphics.

8.2.1 Transfer function design

Transfer function design is a difficult and time-consuming task. Several approaches, reviewed in Section 2.4, have been presented to help the user in the transfer function definition. In our approach, as in [Correa 2011], the visibility is used as a main parameter to be considered for the transfer function specification. But, contrary to most techniques, that require a previous segmentation or classification of the volume data set to automate the process, in our approach no a priori knowledge is required, although this knowledge can also be used to generate importance-driven visualizations.

8.2.2 Information theory in computer graphics

In computer graphics, the most basic information-theoretic measures have been used in scene complexity, global illumination, light positioning, and viewpoint selection for polygonal scenes. In visualization, information theory has been applied to fields such as view selection, flow visualization, time-varying volume visualization, and transfer function definition. Relevant contributions in this area are reviewed in Section 2.8. In our approach information theory is used to define a new framework capable to automatically generate good transfer functions.

8.3 Framework and motivation

In this section, we present the visibility channel that constitutes the framework for the information-theoretic measures used in this paper. Then, we give motivation for our approach to automatic transfer function definition.

8.3.1 Visibility channel

Viola et al. [Viola 2006] defined a *visibility channel* between a set of viewpoints and the set of objects of a volume data set. From this channel, different information-theoretic measures for viewpoint selection and illustrative visualization have been defined (see Section 2.9).

In this paper, we consider an information channel $V \rightarrow B$ between random variables V and B that are defined over the alphabets \mathcal{V} (set of viewpoints) and \mathcal{B} (set of intensity bins), respectively. Thus, the visibility channel proposed by Viola et al. [Viola 2006] has been slightly modified to deal with the intensity bins of the volume data set instead of objects or voxels. Note that in the explanation of this channel we consider that each bin corresponds to the set of voxels that have the same intensity value but

this perspective can be extended to consider other binning strategies such as clusters of intensities or pairs (*intensity, gradient*). Viewpoints are indexed by v and intensity bins by b , and the capital letters V and B as arguments of $p(\cdot)$ are used to denote probability distributions. For instance, while $p(v)$ denotes the probability of a single viewpoint v , $p(V)$ denotes the input distribution of the set of viewpoints. Although diverse configurations of viewpoints can be used, it is assumed here that all the volume data sets are centered in a sphere of viewpoints and the camera is looking at the center of this sphere.

The main elements of the channel $V \rightarrow B$ are the following:

- The *transition probability matrix* $p(B|V)$, that is constituted by the conditional probabilities $p(b|v)$, given by the normalized projected visibility of intensity bin b over a viewpoint v . A row of that matrix is denoted by $p(B|v)$ and satisfies that the sum of its elements is equal to 1: $\sum_{b \in \mathcal{B}} p(b|v) = 1$. Conditional probability $p(b|v)$ is given by

$$p(b|v) = \frac{\text{vis}(b|v)}{\text{vis}(v)}, \quad (8.1)$$

where $\text{vis}(b|v)$ is the visibility of intensity bin b from viewpoint v and $\text{vis}(v) = \sum_{b \in \mathcal{B}} \text{vis}(b|v)$ is the captured visibility of all intensity bins over the sphere of directions centered at v . The visibility $\text{vis}(b|v)$ of an intensity bin b from a viewpoint v is the sum of the visibilities from viewpoint v of all voxels that have intensity b : $\text{vis}(b|v) = \sum_{f(z)=\text{intensity}(b)} \text{vis}(z|v)$, where $f(z)$ is the intensity value of voxel z . The visibility $\text{vis}(z|v)$ of a voxel z from a viewpoint v is equal to the contribution of voxel z to the final image according to its opacity and also to the opacity of the preceding voxels in each ray that visits it [Levoy 1988]. For example, a fully opaque voxel that is seen from one ray, and that is not occluded at all by any other voxel in this ray, has a visibility of 1.

- The *input distribution* $p(V)$, that contains the probability of each viewpoint. An element $p(v)$ of this probability distribution can be interpreted as the importance of viewpoint v and is given by

$$p(v) = \frac{\text{vis}(v)}{\sum_{i \in \mathcal{V}} \text{vis}(i)}. \quad (8.2)$$

- The *output distribution* $p(B)$, where each element $p(b)$ is given by

$$p(b) = \sum_{v \in \mathcal{V}} p(v)p(b|v) \quad (8.3)$$

and expresses the average projected visibility of intensity bin b from all viewpoints.

Note that all probabilities of this channel depend on the applied transfer function. Thus, different transfer functions will generate different transition probability matrices, and input and output distributions.

In this framework, the *conditional entropy* $H(B|V)$ is given by the weighted average entropy for all viewpoints:

$$H(B|V) = - \sum_{v \in \mathcal{V}} p(v) \sum_{b \in \mathcal{B}} p(b|v) \log p(b|v) = \sum_{v \in \mathcal{V}} p(v) H(B|v), \quad (8.4)$$

where $H(B|v) = - \sum_{b \in \mathcal{B}} p(b|v) \log p(b|v)$ is the *viewpoint entropy* [Vázquez 2001] of viewpoint v . As the viewpoint entropy quantifies the visibility uncertainty from a given viewpoint v , the conditional entropy expresses the average visibility uncertainty from all viewpoints.

The degree of dependence or correlation between a set of viewpoints \mathcal{V} and the intensity bins \mathcal{B} is expressed by the *mutual information* $I(V; B)$:

$$I(V; B) = \sum_{v \in \mathcal{V}} p(v) \sum_{b \in \mathcal{B}} p(b|v) \log \frac{p(b|v)}{p(b)} = \sum_{v \in \mathcal{V}} p(v) I(v; B), \quad (8.5)$$

where $I(v; B) = \sum_{b \in \mathcal{B}} p(b|v) \log \frac{p(b|v)}{p(b)}$ is the *viewpoint mutual information*, which measures the degree of dependence between the viewpoint v and the set of bins (see [Viola 2006]).

For the discussion in the next subsection, we introduce the definition of the informational divergence. The *informational divergence* or *Kullback-Leibler distance* $D_{\text{KL}}(p||q)$ between two probability distributions p and q [Cover 1991, Yeung 2008], that are defined over the alphabet \mathcal{X} , is given by

$$D_{\text{KL}}(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \quad (8.6)$$

The conventions that $0 \log \frac{0}{0} = 0$ and $a \log \frac{a}{0} = \infty$ if $a > 0$ are adopted. $D_{\text{KL}}(p||q)$ can be interpreted as a divergence measure between the *true* probability distribution p (observed data) and the *target* probability distribution q (theoretical model or description). The informational divergence satisfies the information inequality $D_{\text{KL}}(p||q) \geq 0$, with equality if and only if $p = q$. The informational divergence is not strictly a metric since it is not symmetric and does not satisfy the triangle inequality. It is important to emphasize that both the viewpoint mutual information and the viewpoint entropy can be obtained from the informational divergence as follows: $I(v; B) = D_{\text{KL}}(p(B|v)||p(B))$ and $H(B|v) = \log |\mathcal{B}| - D_{\text{KL}}(p(B|v)||\{1/n\})$, where $\{1/n\}$ is the uniform distribution (see [Cover 1991]).

8.3.2 Motivation

As we commented above, the automatic definition of a transfer function for a volumetric data set is still a big challenge. This has been also emphasized in the recent survey “Information theory in scientific visualization” by Wang and Shen [Wang 2011].

Taking into account that mutual information and conditional entropy represent, respectively, the information transfer and the average uncertainty in a channel, one

might wonder whether within the previous framework we can use the maximization (or minimization) of mutual information (or conditional entropy) to find out the most informative transfer functions. In Section 8.3.1, we have observed that the viewpoint mutual information $I(v; B)$ is a Kullback-Leibler distance (Equation (8.6)) between the visibility distribution $p(B|v)$ over the viewpoint v and the average visibility $p(B)$ of the intensity bins, and that the mutual information $I(V; B)$ is the weighted average of the distances $I(v; B)$. Thus, the maximization of mutual information would be achieved when the visibility distributions $p(B|v)$ of each viewpoint v significantly diverged from the average visibility $p(B)$. As $I(V; B)$ expresses the correlation between viewpoints and intensity bins, its maximization would imply that each view saw a very specific part of the volume data set and different from what the other viewpoints were seeing. In general, the maximization of mutual information would tend to produce a very opaque transfer function since more occlusions would permit to see different objects from different viewpoints. On the contrary, the minimization of mutual information would produce a very transparent visualization due to the fact that all the viewpoints should see all the bins of the data set with a similar projected visibility. From Equation (8.4), we observe that conditional entropy $H(B|V)$ is the weighted average of the entropy $H(B|v)$ for all viewpoints. Thus, the minimization or the maximization of the conditional entropy would try to see, respectively, only one intensity bin from each viewpoint (i.e., minimal uncertainty) or all the bins with the same probability (i.e., maximal uncertainty). In general, the previous options are too specific and, thus, are not good candidates to accomplish our purpose of finding appropriate transfer functions.

As we have mentioned in the previous section, the informational divergence can be interpreted as a distance between the observed data and a theoretical description which can include the relevance of data. From this reasoning and the previous considerations on the convenience of using mutual information or conditional entropy, we adopt a more global strategy to define an informative transfer function: *to minimize the informational divergence between the average projected visibility distribution from all viewpoints (calculated for a given transfer function) and a target distribution which expresses our theoretical objective*. Thus, using an iterative process, the transfer function will evolve to fulfill the requirements of the target distribution. As we will see in the next section, all these measures can be used to deal with both 1D and multidimensional transfer functions.

It is interesting to note that this strategy could be extended to consider the average of the informational divergences between the visibility distribution of each viewpoint and a global target distribution. In this case, for specific target distributions, both the mutual information and the conditional entropy would be obtained as particular cases (see Section 8.3.1). In this chapter, we focus our attention on the most global form of the informational divergence and future research will be done comparing it with the average informational divergence.

8.4 Method

The basic idea of our transfer function specification method is to minimize the distance between the distribution of the projected visibility of a volumetric data set (from an initial arbitrary transfer function that evolves towards the desired transfer function) and a target distribution that expresses the data importance, or highly relevant data value interval, or spatial segmentation.

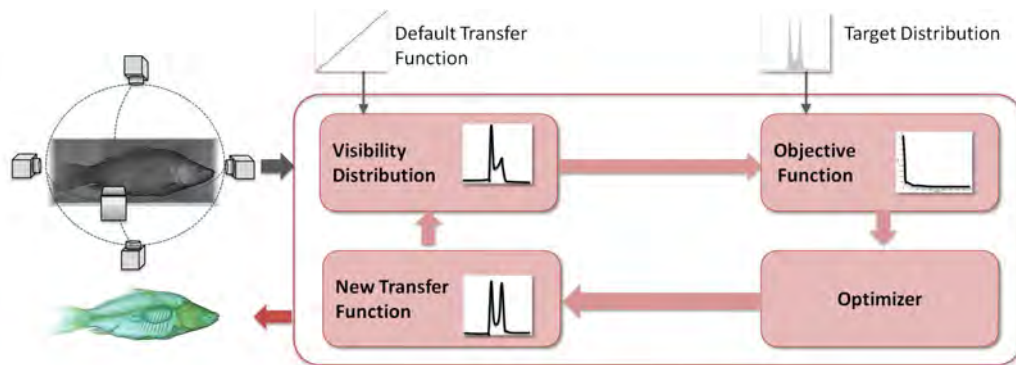


Figure 8.2: The volume data set, a default transfer function, and a target distribution that reflects the data importance are all entered into the system. Then, an optimization process is performed and the opacity function that minimizes the distance to the target distribution is obtained.

The process of finding the optimal transfer function is represented in Figure 8.2. This process begins with an initial transfer function given by the user (if no specific function is provided the linear ramp is used). With this initial transfer function and the input data, the visibility distribution is computed for a set of viewpoints. Then, the objective function, based on the informational divergence or Kullback-Leibler distance between the obtained visibility distribution and the target one, is evaluated. From this value, the optimizer, based on the steepest gradient descent algorithm, assesses a new transfer function in the direction of the gradient. The process is repeated with the new transfer function until the value of the objective function is below a given threshold or a given number of iterations has been performed.

One of the limitations of the steepest gradient descent is the determination of its constant step value. If this is too high, the process converges quicker but the accuracy is lower. On the other hand, if the step value is too low, the convergence is reached with more iterations and the probability of falling in a local minimum is higher. In order to improve the convergence process, we have implemented two variants of the basic method. First, the algorithm adjusts the step value for each bin and, therefore, there is not a single global value. Second, if the sign of the gradient at a given bin changes for a given iteration, which means that we have exceeded the minimum, then the step size for this intensity is halved in order to have more accuracy.

Next, the main parts of this process are described: the target distributions, the objective function, and the optimizer.

8.4.1 Target distribution

The target distribution represents an importance-based description of what the user expects to be visualized, i.e., the probability of each bin at the final rendered image. We propose two main strategies to define the target distribution: a global strategy, which considers general features of the volume data set, and an importance-based strategy, which exploits a priori knowledge of the data.

8.4.1.1 Global strategies

We present here different target distributions based on global properties that can be derived from the volume data set. There are many possible target distributions, amongst them:

- The uniform distribution:

$$q(b) = \frac{1}{N_B}, \quad (8.7)$$

where N_B is the number of intensity bins. In this case, we are assigning the same probability to each intensity bin. Observe that, if we have an intensity bin with only one voxel, our algorithm will tend to give too much importance to this bin, and too little importance to the bins with the highest occurrence.

- The probability distribution obtained from the occurrence of each intensity bin:

$$q(b) = \frac{\text{occurrence}(b)}{\sum_{i \in \mathcal{B}} \text{occurrence}(i)}. \quad (8.8)$$

This approach requires that each intensity bin is visualized according to its probability in the volume data set.

- The probability distribution obtained from the depth of bins:

$$q(b) = \frac{\text{depth}(b)}{\sum_{i \in \mathcal{B}} \text{depth}(i)}, \quad (8.9)$$

where $\text{depth}(b)$ is computed as $\text{depth}(b) = d_{\max} - \max_{f(z)=\text{intensity}(b)} |z - c|$, d_{\max} is the maximum distance of any voxel to the volume center c , z is the position of a voxel, and $f(z)$ is the intensity value at voxel z . In this approach, the intensities which are close to the center of the image will have a higher probability of being projected. This strategy has an important drawback since the intensity values located close to the image center with low occurrence could have a high target probability, and therefore the method would try to magnify its projected probability by increasing its opacity and decreasing the opacities of the other intensities, giving a very transparent final solution. In order to overcome this drawback, the depth distribution will be weighted by the occurrence of the intensity value. Note that the new distribution will have to be normalized again.

- The probability distribution obtained from the intensity value itself:

$$q(b) = \frac{\text{intensity}(b)}{\sum_{i \in \mathcal{B}} \text{intensity}(i)}. \quad (8.10)$$

This approach assigns more probability of being projected to the highest intensities. For instance, in medical imaging, when a contrast agent enhances the vessels in MR angiography or a PET tracer highlights metabolic spots, the highest intensity values are the most relevant. As in the previous approach, we will weight this value by the occurrence of each intensity value.

These global strategies can be easily extended to multidimensional transfer functions. In particular, we use the 2D space generated by intensity and gradient. In this case, the four previous strategies can be utilized in the same way than in the 1D transfer function by quantizing the intensity and the gradient to a limited number of clusters. Our experiments work with 256 intensity clusters and 16 gradient magnitude clusters (both uniform). Thus, we are using a random variable with an affordable alphabet of 4096 possible values. Using this extension, another target can be defined:

- The probability distribution obtained from the gradient values:

$$q(b) = \frac{\text{gradient}(b)}{\sum_{i \in \mathcal{B}} \text{gradient}(i)}. \quad (8.11)$$

Note that \mathcal{B} represents now the joint variable (*intensity, gradient*). In this case, the voxels with high gradients will be highlighted. As in other previous strategies, this distribution will be commonly weighted by the occurrence.

8.4.1.2 Importance-based strategies

It is very common to have a priori knowledge about which intensities are relevant or which regions of the data should be emphasized. For instance, in CT data the intensity ranges are determined by the Hounsfield scale. Thus, the user can manually determine which intensity ranges are relevant and which not. It is also common to be interested in preserving the context of the area of interest. This context has to be visualized in the final image with less importance than the relevant areas.

To achieve the above objectives, the target distributions presented in the previous section can be weighted by an importance function *importance(b)*. In this way, a priori knowledge of the data is combined with statistical features of the data. Since the sum of all elements of the target distribution must be one, a normalization step has to be done after the combination of the global and importance weights.

In medical imaging, physicians are usually interested on certain spatial regions which share the same intensity ranges with other non-relevant anatomical regions. Since the proposed basic method only deals with intensity values, it will not be able to emphasize only the interest region. To tackle this problem, segmentation strategies that create a mask containing the region of interest are applied. This mask can be obtained

either by an image processing algorithm or by a manual segmentation. Our method can be easily adapted to take this spatial information into account. If we assume that the original volume data set takes intensity values from 0 to max , a new data set is generated by keeping the intensity values of the voxels outside the mask and adding the value $max+1$ to the intensity values of the voxels inside the mask. After this simple process, the new volume data set can take values from 0 to $2max+1$. This is equivalent to add a bit to the volume intensity according to the mask. Then, our method can be applied to this new data set.

8.4.2 Objective function

The kernel of our method is given by a divergence measure between the visibility distribution and a target distribution. A common measure of probability distribution distance in information theory is the informational divergence or Kullback-Leibler distance [Cover 1991, Yeung 2008] (see Equation (8.6)).

From the informational divergence, two different measures between the projected visibility of each bin and a target distribution $q(B)$ can be defined depending on how the visibility is estimated:

- *Global informational divergence* (GID), which is defined as

$$D_{\text{KL}}(p(B)||q(B)) = \sum_{b \in \mathcal{B}} p(b) \log \frac{p(b)}{q(b)}, \quad (8.12)$$

where $p(b) = \sum_{v \in \mathcal{V}} p(v)p(b|v)$, and $p(v)$ is obtained from the normalization of the projected visibility of the volume data set over the viewpoint v (see Equations (8.2) and (8.3)). Thus, $p(B)$ represents the mean visibility of each intensity considering all the viewpoints. As $p(B)$ has to become similar to $q(B)$ to obtain the desired transfer function, our objective is to minimize the global informational divergence (Equation (8.12)).

- *Viewpoint informational divergence* (VID). When we only consider the current viewpoint v , Equation (8.12) becomes

$$D_{\text{KL}}(p(B|v)||q(B)) = \sum_{b \in \mathcal{B}} p(b|v) \log \frac{p(b|v)}{q(b)}, \quad (8.13)$$

where $p(B|v)$ represents the visibility of each intensity by considering only the current viewpoint (see Equation (8.1)). Note that this measure is view dependent and will have to be recomputed each time the viewpoint changes.

Considering that the target distribution $q(B)$ and the set of viewpoints are constant during the visualization, observe that the previous measures only depend on the opacities of each intensity in the transfer function definition. From now on, this opacity vector will be denoted as $A = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, where n is the number of intensity bins.

The main drawback of the minimization of informational divergence is that the final transfer function could be given by too low opacity values, i.e., the final solution could be too transparent. This is due to the fact that the informational divergence is based on probability distributions, which can be seen as ratios of visibility, and not on absolute visibility values. Since the excessive transparency is not a desired feature, an opacity constraint can be included. Thus, a possible loss of opacity can be compensated by the requirement of maximizing the global visibility. Hence, our *objective function* is defined as

$$F(A) = (1 - \beta)D_{\text{KL}}(p||q) - \beta \frac{E}{E_{\text{max}}}, \quad (8.14)$$

where $D_{\text{KL}}(p||q)$ stands for $D_{\text{KL}}(p(B)||q(B))$ or $D_{\text{KL}}(p(B|v)||q(B))$, E is the absorbed energy for all the rays during the ray casting, E_{max} is the maximum achievable energy absorption, which corresponds to the one when the transfer function is completely opaque, and β is a parameter that weighs the contribution of the opacity constraint term. Observe that E also depends only on the opacity vector A .

Figure 8.3 plots the visibility distributions for the CT-body using the global strategy that considers the occurrence distribution as target. In the plot, the target distribution is represented by a grey area, the initial visibility distribution using the default transfer function by a blue line, and the final visibility distribution by a red line, respectively. For this experiment, the parameter β has been set to 0 and, thus, $F(A) = D_{\text{KL}}(p||q)$. For the initial distribution, $F(A) = 0.188$, while for the final distribution, $F(A) = 9.02 \cdot 10^{-9}$.

8.4.3 Optimizer

The proposed method is implemented using the steepest gradient descent optimizer [Press 1992]. The goal of this method is to find the vector of opacities A which minimizes the objective function $F(A)$ defined in Equation (8.14). This method requires an initial step size s , which is set to 1 in our implementation. The optimization process consists of an iterative algorithm in which each opacity value of the new transfer function at iteration t is computed as

$$A^t = A^{t-1} - s^{t-1} \nabla F(A), \quad (8.15)$$

where the symbol ∇ represents the gradient and, therefore, $\nabla F(A) = \left(\frac{\partial F(A)}{\partial \alpha_0}, \frac{\partial F(A)}{\partial \alpha_1}, \dots, \frac{\partial F(A)}{\partial \alpha_{n-1}} \right)$. The gradient computation is detailed in the next subsection. The higher the s value, the faster the convergence but the lower the accuracy. In the basic method this s value is constant for all iterations and intensity bins, but it can be easily modified in order to have an adaptive step for each bin. In our experiments we use an adaptive method. Once the new transfer function is defined, the new value of the objective function has to be computed. If this value is lower than the global minimum, the current transfer function is considered as the new optimal one. Once the program has converged, or has done all the iterations, the volume data set is rendered using the optimal transfer function.

Figure 8.4 plots the $F(A)$ values for the CT-body case using the occurrence target

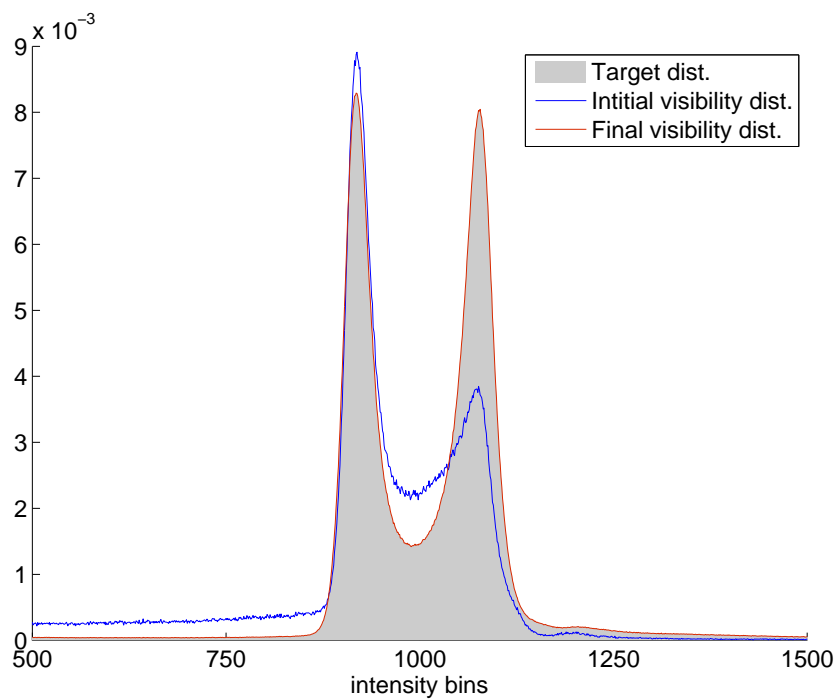


Figure 8.3: Visibility distribution obtained with GID, 6 viewpoints, and the occurrence target distribution applied to the CT-body: (grey area) the target distribution, (blue line) the initial visibility distribution using the linear ramp transfer function, and (red line) the final visibility distribution.

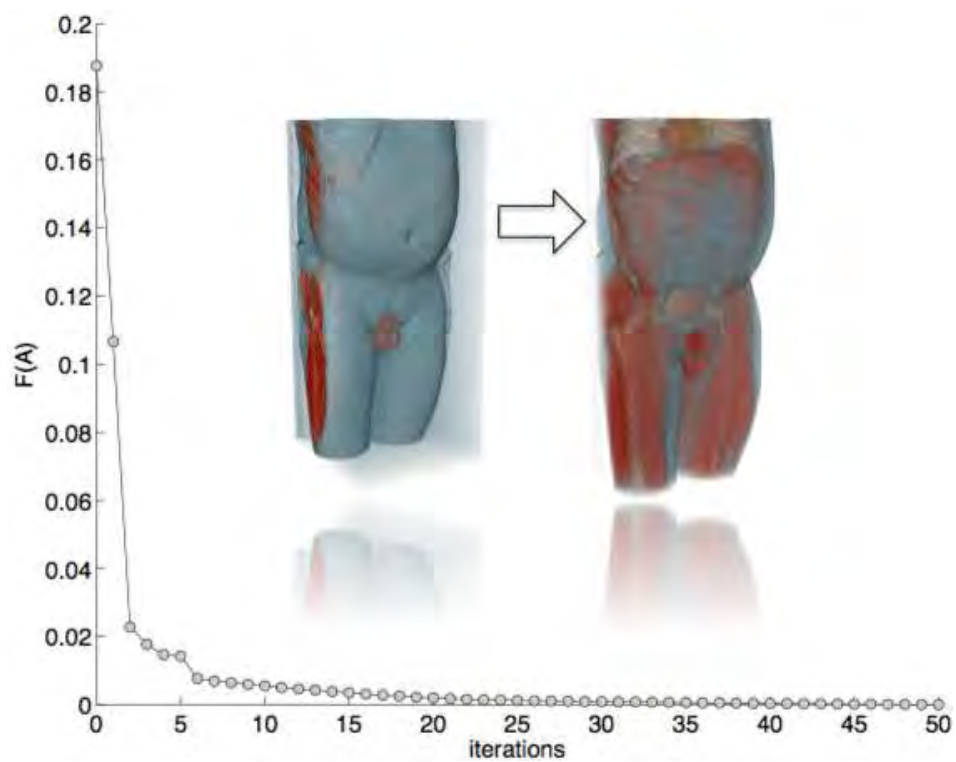


Figure 8.4: Evolution of the objective function $F(A)$ for GID with respect to the number of iterations for the CT-body using the occurrence target distribution, $\beta = 0$, and 6 viewpoints.

distribution, $\beta = 0$, and 6 viewpoints with respect to the number of iterations. As it can be seen, the measure tends to 0 and converges with less than 50 iterations. As $D_{\text{KL}}(p||q) \geq 0$, when $\beta = 0$, we will consider that the method will have converged when $F(A)$ achieves a value lower than a given threshold, typically 0.001.

8.4.3.1 Gradient computation

In typical gradient descent optimizer scenarios, the gradient computation is one of the most computationally demanding steps, since its estimation requires that the objective function is computed $2n$ times, where n is the number of the degrees of freedom. In our case, n corresponds to the number of intensity bins, which can be such a large number that can make this approach unfeasible. In this case, an analytical computation, or at least a reasonable good approximation, becomes the best way to tackle the problem. In this section, we describe how this approximation can be computed.

In order to compute $\nabla F(A)$, we will have to determine $\frac{\partial F(A)}{\partial \alpha_i}$ for all intensities i . From the linearity property of the derivative, we have that

$$\frac{\partial F(A)}{\partial \alpha_i} = (1 - \beta) \frac{\partial D_{\text{KL}}(p||q)}{\partial \alpha_i} - \frac{\beta}{E_{\text{max}}} \frac{\partial E}{\partial \alpha_i}. \quad (8.16)$$

Therefore, we can divide the computation into two parts: $\frac{\partial D_{\text{KL}}}{\partial \alpha_i}$ and $\frac{\partial E}{\partial \alpha_i}$.

First, we would like to compute the derivative of the Kullback-Leibler divergence between the observed intensities distribution and the objective distribution when the opacity of a single intensity is modified in the transfer function.

Let X be the original observed intensities with a distribution $X = \{p_1, p_2, \dots, p_n\}$ and Y be the objective distribution $Y = \{q_1, q_2, \dots, q_n\}$. Let see what happens in X when the opacity α_i corresponding to intensity i is modified. During the probability computation process, the probability of the intensity i is estimated as

$$p_i = \frac{\sum_k I_k \alpha_i}{\sum_i \sum_k I_k \alpha_i} = \frac{e_i \alpha_i}{E}, \quad (8.17)$$

where k represents a sample with intensity i , I_k is the remaining transparency corresponding to the sample k , and, then, $e_i = \sum_k I_k$ is a multiplicative factor which depends on the number of samples of this intensity and their remaining transparencies. Finally, $E = \sum_i \sum_k I_k \alpha_i$ is a normalization factor (to ensure that the distribution sums 1) which represents the absorbed energy of all intensities. Then, we can assume that if the opacity of the intensity i is increased by $\Delta \alpha_i$, the new probability will be approximately

$$\begin{aligned} p'_i &= \frac{\sum_k I'_k(\alpha_i + \Delta \alpha_i)}{E'} \\ &\approx \frac{\sum_k I_k(\alpha_i + \Delta \alpha_i)}{E} = \frac{e_i(\alpha_i + \Delta \alpha_i)}{E} = p_i \left(1 + \frac{\Delta \alpha_i}{\alpha_i} \right). \end{aligned} \quad (8.18)$$

We assume here that the remaining transparency will be approximately the same, since the opacities remain the same except for $\Delta\alpha_i$. Hence, we can define k_i as

$$k_i = 1 + \frac{\Delta\alpha_i}{\alpha_i}. \quad (8.19)$$

We can also assume that all the other probabilities will be modified by a multiplicative factor K_i that is the same for all of them. In order to have the summation of the distribution normalized to 1, we get the equation $k_i p_i + K_i(1 - p_i) = 1$. Thus, we can define K_i as

$$K_i = \frac{1 - p_i k_i}{1 - p_i}. \quad (8.20)$$

Hence, the probability distribution X' of the intensities after the modification of the opacity will be $X' \approx \{K_1 p_1, K_1 p_2, \dots, k_i p_i, \dots, K_i p_n\}$.

Then, we would like to compute the derivative of $D_{\text{KL}}(p||q)$ for each intensity opacity in the transfer function. In order to compute the derivative we will use

$$\frac{\partial D_{\text{KL}}(p||q)}{\partial \alpha_i} = \lim_{\Delta\alpha_i \rightarrow 0} \frac{D_{\text{KL}}(p'||q) - D_{\text{KL}}(p||q)}{\Delta\alpha_i}, \quad (8.21)$$

where p' is the visibility probability when the transfer function with the modified opacity is applied.

Then, the Kullback-Leibler divergence between X' and the objective distribution Y (which remains constant after the modification of the opacity) is computed as

$$\begin{aligned} D_{\text{KL}}(p'||q) &= \sum_{j=1}^n p'_j \log \frac{p'_j}{q_j} \\ &= K_i D_{\text{KL}}(p||q) + (k_i - K_i) p_i \log \frac{p_i}{q_i} \\ &\quad + K_i(1 - p_i) \log K_i + k_i p_i \log k_i. \end{aligned} \quad (8.22)$$

To compute the derivative we use

$$\begin{aligned} \frac{\partial D_{\text{KL}}(p||q)}{\partial \alpha_i} &= \lim_{\Delta\alpha_i \rightarrow 0} \frac{D_{\text{KL}}(p'||q) - D_{\text{KL}}(p||q)}{\Delta\alpha_i} \\ &= \lim_{\Delta\alpha_i \rightarrow 0} \frac{(K_i - 1) D_{\text{KL}}(p||q)}{\Delta\alpha_i} \\ &\quad + \lim_{\Delta\alpha_i \rightarrow 0} \frac{(k_i - K_i) p_i \log \frac{p_i}{q_i}}{\Delta\alpha_i} \\ &\quad + \lim_{\Delta\alpha_i \rightarrow 0} \frac{K_i(1 - p_i) \log K_i}{\Delta\alpha_i} \\ &\quad + \lim_{\Delta\alpha_i \rightarrow 0} \frac{k_i p_i \log k_i}{\Delta\alpha_i}. \end{aligned} \quad (8.23)$$

By computing these limits, we obtain the following analytical expression:

$$\begin{aligned} \frac{\partial D_{\text{KL}}(p\|q)}{\partial \alpha_i} &= \frac{-p_i}{\alpha_i(1-p_i)} D_{\text{KL}}(p\|q) + \frac{p_i}{\alpha_i(1-p_i)} \log \frac{p_i}{q_i} \\ &\quad + \frac{-p_i}{\ln(2)\alpha_i} + \frac{p_i}{\ln(2)\alpha_i} \\ &= \frac{p_i}{\alpha_i(1-p_i)} \left(\log \frac{p_i}{q_i} - D_{\text{KL}}(p\|q) \right). \end{aligned} \quad (8.24)$$

The second derivative term in Equation (8.16) corresponds to $\frac{\partial E}{\partial \alpha_i}$. As we have previously shown, we have that

$$E = \sum_i \sum_k I_k \alpha_i = \sum_i e_i \alpha_i. \quad (8.25)$$

Then, its derivative can be approximated by

$$\frac{\partial E}{\partial \alpha_i} \approx \sum_k I_k = e_i = \frac{p_i E}{\alpha_i}, \quad (8.26)$$

where p_i is the visibility probability, E is the absorbed energy for all the rays during the ray casting, and α_i is the opacity of intensity i .

Observe that this computation can be done in constant time for each intensity. This allows us to work with the initial intensity resolution with reasonably low computational costs. In our experiments, the computational time spent in this step is much lower than the computation of D_{KL} , where the visibility probability computation is more demanding.

8.5 Results

In this section we present the several experiments that have been carried out to evaluate the proposed approach. For our tests, we have used the following data sets: the CT-head ($512 \times 512 \times 460$ and 4096 intensity bins), the CT-body ($256 \times 256 \times 415$ and 4096 intensity bins), the CT-carp ($256 \times 256 \times 512$ and 2872 intensity bins), the CT-tooth ($256 \times 256 \times 161$ and 1279 intensity bins), and two synthetic brain models from the BrainWeb database [Cocosco 1997]: the MRI-healthy ($181 \times 217 \times 181$ and 4096 intensity bins) and the MRI-damaged ($181 \times 217 \times 181$ and 4096 intensity bins). All the experiments have been carried out on a PC equipped with an Intel Core 2 Quad Q9550 CPU, 4 GiB of RAM, and a NVIDIA GeForce GTX 280 graphics card.

In the following experiments, we have used by default the global informational divergence (GID), $\beta = 0$, a stopping threshold value of the objective function equal to 0.001, and 6 uniformly distributed viewpoints. To show the results we have applied different predefined color transfer functions and local illumination. For comparison purposes, the Euclidean distance proposed by Correa and Ma [Correa 2011] to define a transfer function has been implemented using the steepest gradient descent optimizer.

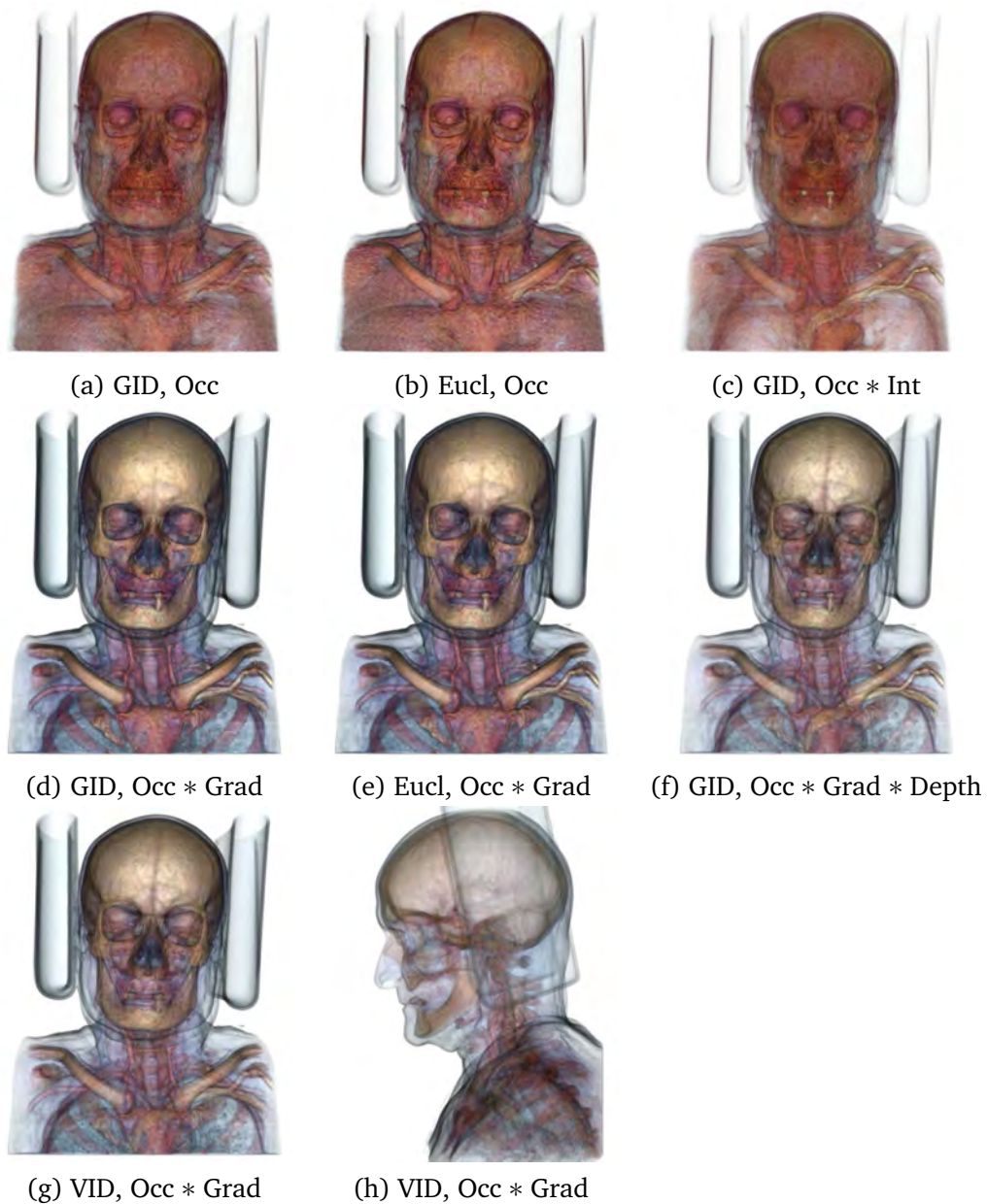


Figure 8.5: Global strategy applied to the CT-head. The following target distributions have been considered: 1D transfer functions with occurrence (Occ) and occurrence weighted by intensity (Occ * Int); 2D transfer functions with occurrence weighted by gradient (Occ * Grad) and occurrence weighted by both gradient and depth (Occ * Grad * Depth). The global informational divergence (GID) has been used in (a), (c), (d), and (f), the Euclidean distance in (b) and (e), and the viewpoint informational divergence (VID) in (g) and (h), where only the viewpoint shown has been used.

Similarly to informational divergence, we have obtained an analytical expression of the gradient of Euclidean distance to speed up the optimization process.

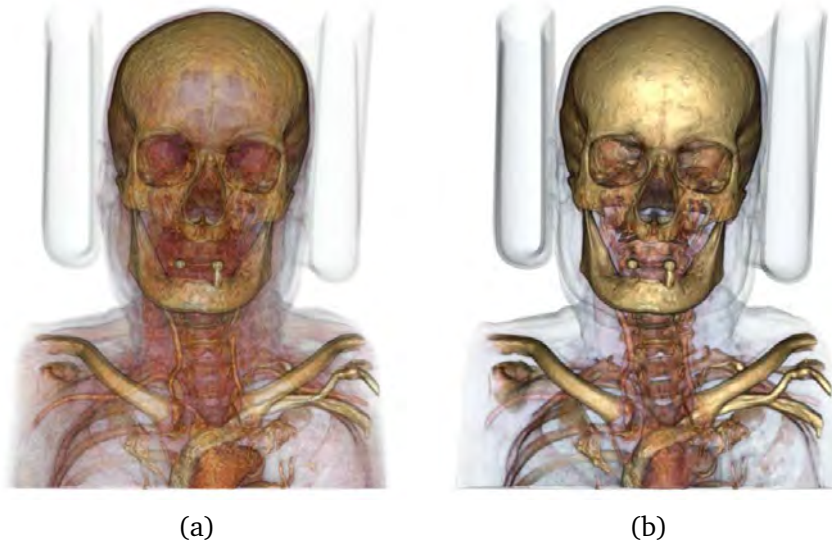


Figure 8.6: Importance-based strategy applied to the CT-head assigning importance 1 to the intensities corresponding to bone structure and 0.2 to the rest, using as target distributions: (a) occurrence and (b) occurrence weighted by gradient.

Figure 8.5 shows the results obtained with the CT-head using the global strategy for both 1D transfer functions (8.5(a), 8.5(b), 8.5(c)) and 2D transfer functions (8.5(d), 8.5(e), 8.5(f)), and considering different target distributions. In 2D transfer functions we have used 256 intensity bins and 16 gradient bins. Fig. 8.5(a) and 8.5(d) have been obtained with occurrence (Occ) and occurrence weighted gradient (Occ * Grad) target distributions, respectively. Fig. 8.5(b) and 8.5(e) show the visualizations obtained with the same target distributions and the Euclidean distance. The visual results obtained from both informational divergence and Euclidean distance are very similar. In average, the time for each iteration and the number of iterations for a given threshold are also similar. However, the use of informational divergence, which is the most natural measure of discrimination in information theory (IT) [Cover 1991, Yeung 2008], enables us to unify in the same framework the viewpoint selection and the transfer function design. Fig. 8.5(c) and 8.5(f) have been obtained using occurrence weighted by intensity (Occ * Int), and occurrence weighted by gradient and depth (Occ * Grad * Depth), respectively. In Figure 8.5(a), which corresponds to the occurrence distribution, the muscle has large visibility due to its greater volume with respect to the other anatomical structures such as bone or skin. Observe that when weighting occurrences by intensity (Figure 8.5(c)) the method assigns high opacities to the bone structure because this has the highest intensity values in CT images. Note that 2D visualizations are clearer than the ones obtained with 1D transfer functions. Since muscle has a low gradient magnitude, it is almost transparent and allows to visualize inner structures, specially with the target of occurrence weighted by gradient (Figure 8.5(d)). Gradi-

ent information provides a better delineation of organ boundaries and hence a clearer representation is obtained. We also observe that introducing depth information into the target distribution (Figure 8.5(f)) the obtained image is more transparent, but the boundaries of the organs can still be perceived. The last two images (Fig. 8.5(g) and 8.5(h)) have been obtained with the viewpoint informational divergence (VID) from the viewpoint shown and the occurrence weighted by gradient target distribution. In this case, image cluttering increases because the method assigns the opacities so that all structures are seen according to the target distribution and, consequently, since only one viewpoint is considered, the structures have to become more transparent.

The next experiment has been designed to illustrate the importance-driven strategy supported by our approach. To assign importance to the volume we can consider two alternatives: a relevant intensity range and a segmentation mask. For each alternative, an importance value is given to the most relevant part and a lower one to the context. The first alternative is illustrated in Figure 8.6, where the CT-head is rendered after assigning importance to the intensity range of the bone structure. Figure 8.6(a) has been obtained with the target of occurrence weighted by importance (1 for the bone and 0.2 for the rest). If we compare this image with the one of Figure 8.5(a), obtained with the same target but without importance, we can observe how the bone is highlighted. Figure 8.6(b) has been obtained using the same importance distribution and the target distribution given by occurrence weighted by gradient. Comparing this image with the one of Figure 8.5(d), obtained with the same target but without importance, we can observe how the bones are more prominent. The result of giving importance to an intensity range is also illustrated in the second image of Figure 8.1, where the target distribution is occurrence weighted by importance, and we have given importance 1 to the enamel and 0.5 to the rest.

To illustrate the second alternative, which uses a segmentation mask, we apply the importance-driven strategy to the MRI brain data sets. In figures 8.7(a), 8.7(b), 8.7(c), and 8.7(d), we show the original MRI-healthy model and the importance-based visualizations obtained by assigning importance to the white matter, the grey matter, and the glial tissue masks, respectively. The last image (Figure 8.7(e)) has been obtained from the MRI-damaged data set by assigning importance to the damaged area. In all the cases, we have assigned importance 1 to the mask. The rest of the model has importance 0.2 in Fig. 8.7(b) and 8.7(c), and 0.1 in Fig. 8.7(d) and 8.7(e). The volume data set is rendered in greyscale and using color in the important area. Observe that, although the regions of interest have different sizes and occluding tissues, they are always highly visible while the context is kept. In the fourth image of Figure 8.1, we can see another example of giving importance to a segmented region. In this case, the nerve has been highlighted.

The next experiments aim at showing the effects of using the parameter β and different number of viewpoints. Figure 8.8 shows the visualization of the CT-body, using the target distribution given by occurrence weighted by gradient. These visualizations have been obtained by setting β to 0, $\frac{2}{3}$, and $\frac{8}{9}$. Remember that this parameter weights the contribution of the opacity constraint in the objective function. The higher the value of β , the higher the total opacity of volume visualization. As it can be seen, by

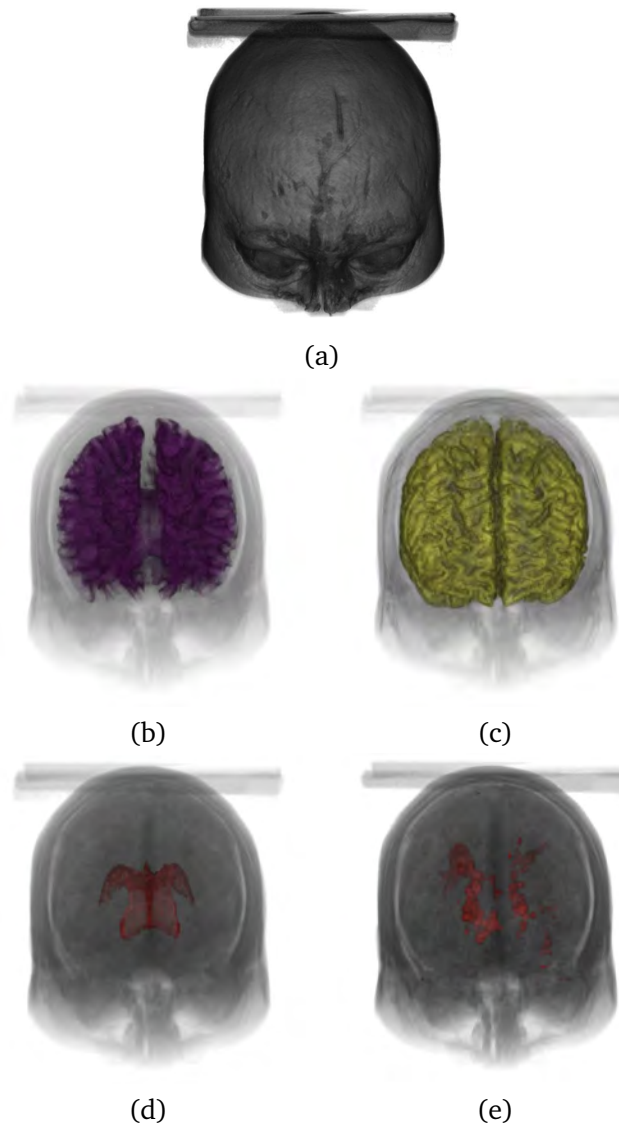


Figure 8.7: Importance-based strategy applied to two MRI brain data sets: (a) initial visualization of the MRI-healthy and the resulting visualization giving importance to (b) white matter, (c) grey matter, (d) glial tissue. (e) Visualization of the MRI-damaged giving importance to the lesion area. In (b) and (c) the importance is 1 for the masks and 0.2 for the rest. In (d) and (e) the importance is 1 for the masks and 0.1 for the rest.

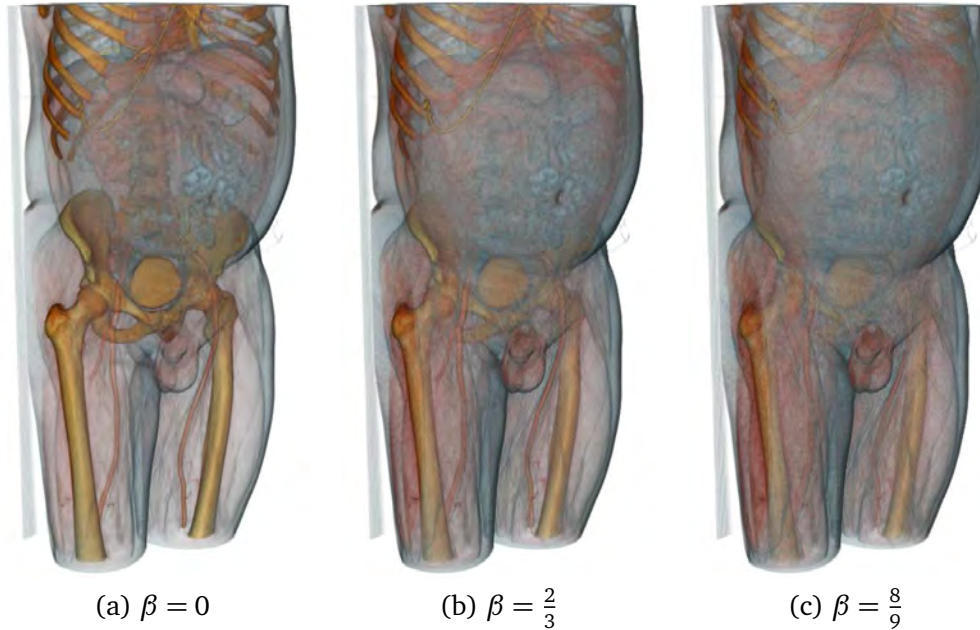


Figure 8.8: Visualization of the CT-body obtained from the target function given by the occurrence weighted by gradient and β values equal to 0, $\frac{2}{3}$, and $\frac{8}{9}$, respectively.

increasing the parameter β the final transfer function becomes more opaque.

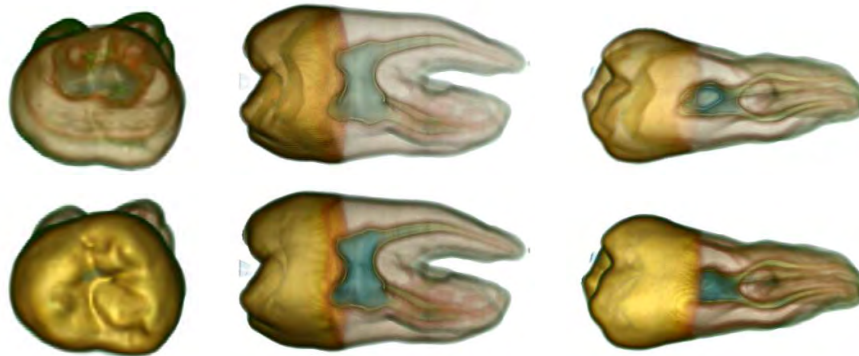


Figure 8.9: Visualization of the CT-tooth obtained from the target distribution given by the occurrence weighted by gradient for three different viewpoints (first row) and the same corresponding views computed with 42 viewpoints (second row).

Figure 8.9 shows the visualization of the CT-tooth obtained using the target distribution occurrence weighted by gradient for three different viewpoints (first row) and the same corresponding views computed with 42 viewpoints (second row). Note how the transfer functions are clearly dependent on the selected viewpoints. In the case of a single viewpoint, all structures are visible from each viewpoint, while considering more viewpoints occlusions do not allow to perceive all structures from a single viewpoint.

Figure 8.10, shows the visualization of CT-carp obtained using the target distribu-

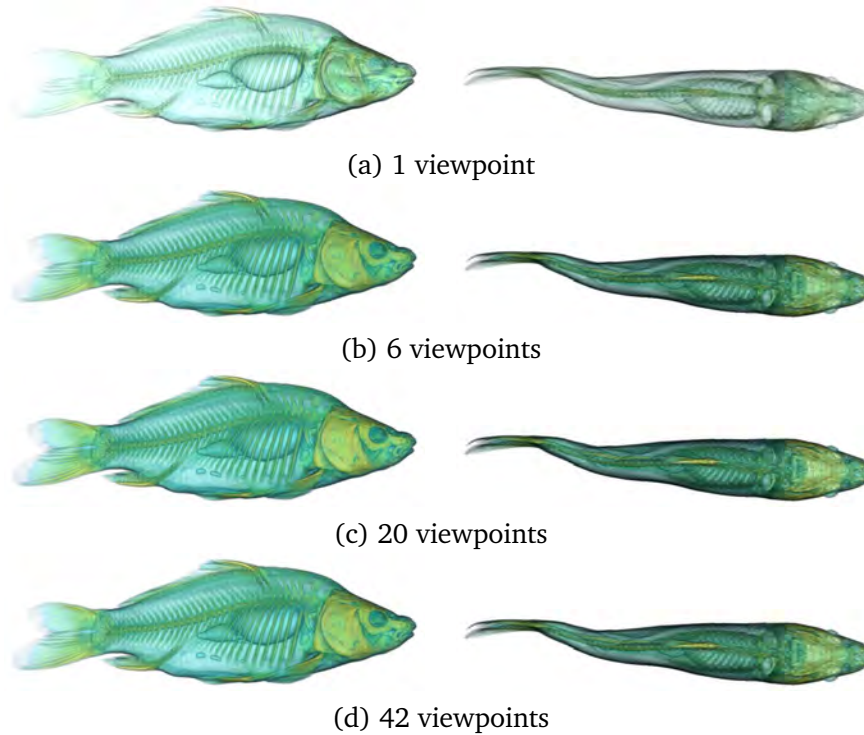


Figure 8.10: CT-carp is visualized considering 1, 6, 20, and 42 viewpoints using as target distribution the occurrence weighted by gradient with $\beta = 0.5$.

tion occurrence weighted by gradient with $\beta = 0.5$, and for 1, 6, 20, and 42 viewpoints. Note that the differences obtained using 6, 20, or 42 viewpoints are minimal and, hence, the use of 6 viewpoints is a good trade-off between quality and speed for the global informational divergence.

To evaluate the computation time, we have considered different data sets, target distributions, and stopping threshold values of the objective function. Table 8.1 reports the obtained results. From left to right, the first two columns indicate the evaluated model and the used target distribution, and the next columns are the thresholds of the distances considered to stop the process. For each configuration we collect the computation time in seconds and the number of iterations required by different target distributions. In the two bottom rows, we show respectively the computation time for the CT-carp when only one viewpoint is considered, and the computation time for the MRI-healthy when the importance is driven by the white matter mask. The performance of our methods only benefits from the GPU implementation of the visibility computation, because the rest of computations have been done in CPU. The results usually converge in less than 50 iterations. With more iterations, the results are barely improved from a perceptual point of view. If we compare the computational time of our experiments with manual editing, that ranges from 5 to 20 minutes [Kohlmann 2007], the time improvement obtained with our method is significant.

Data set	Target	≤ 0.01	≤ 0.001	≤ 0.0001
CT-body	GID, Occ	7 it.	30 it.	50 it.
		4.76	20.70	34.60
CT-tooth	GID, Occ * Imp	17 it.	31 it.	63 it.
		5.12	9.49	19.24
CT-head	GID, Occ * Grad * Int	15 it.	19 it.	34 it.
		22.57	29.45	56.55
CT-carp	VID, Occ * Grad	20 it.	24 it.	32 it.
		3.40	4.86	6.98
MRI-healthy	GID, Occ * Imp	21 it.	31 it.	55 it.
		18.76	29.32	54.26

Table 8.1: Number of iterations and time cost (in seconds) for different data sets and stopping threshold values. Target distributions are: occurrence (Occ), occurrence weighted by importance (Occ * Imp), occurrence weighted by gradient and intensity (Occ * Grad * Int), and occurrence weighted by gradient (Occ * Grad).

8.5.1 Limitations

The automatic definition of a good transfer function is a big challenge. There are few methods that fully automate the transfer function design and most of them require previous knowledge or a pre-segmentation of the volume model. Our approach requires to define the target distribution and the parameter β has to be assigned by the user as well. Strictly speaking, because of these user settings, the method cannot be classified as fully automatic. However, we have experimentally observed that good results are already achieved with $\beta = 0$. Also, the target distribution is much more intuitive to define for the user (similarly to importance definition) than the transfer function.

In our approach, we have focused on opacity transfer functions considering that opacity is the main factor that affects the visibility of structures. In the medical scenario this would be sufficient as the radiologists use primarily grey levels to depict structures when it comes to visualization of single imaging modality. In our examples we have used pre-defined color transfer functions to increase the contrast in printed images by additional contrast in chromatic channel. Adapting the color distribution automatically can be seen as natural continuation of our research.

8.6 Conclusions

We have presented a new information-theoretic framework for automatic transfer function design that, based on a user-defined target distribution, obtains the opacity transfer function whose visibility distribution minimizes the informational divergence to the target. We have proposed different target distributions defined on intensities or on a 2D space which considers the intensities and the gradient magnitudes. We provided two strategies, a global one that considers general features of the data and an importance-driven one that emphasizes intensity ranges or regions of interest. The different options

of the method have been evaluated on several data sets. It has been seen that the results with a global strategy give a good overall visualization of the volume data set, while the ones obtained with the importance-based strategy achieve focus+context visualizations. In addition we have evaluated the results obtained using the informational divergence of only one viewpoint. With this approach the computational cost is reduced considerably, but a recomputation is necessary when the viewpoint is changed. The computational cost of the proposed methods is low enough to make this technique feasible for real environments.

Future research will be done to define a new objective function that approximates the target distribution while maximizing the projected visibility without requiring a parameter such as β . We will also investigate the use of the average of the informational divergences for each viewpoint. In addition, we will analyze the performance of other distances, such as f -divergences or Jensen-Shannon divergence. Future work will be done to obtain a fully GPU implementation. To improve the computational cost, we will also explore new optimization methods.

Conclusions

Contents

9.1 Contributions	135
9.2 Future work	138

In last decades, volume visualization techniques have become fundamental in many scientific areas and specially in medical applications. These techniques aim to communicate information and assist interpretation of volumetric data sets in a clear and effective manner by means of graphical representations. For instance, in a medical environment realistic renderings are used for diagnosis and pre-operative planning, while non-photorealistic techniques are preferred in educational environments since they simplify data and produce clearer images than traditional photorealistic methods. It is not easy to fulfill these objectives because of the difficulty of algorithmically producing some types of renderings, and also because of the high computational cost of some of the techniques, such as global illumination. The challenge is how to exploit graphic capabilities to create visualizations suited to the user needs.

In this thesis, we have focused our interest on advanced volume illumination techniques. Our purpose has been to improve volume rendering techniques to produce high quality renderings, with both realistic and illustrative styles. We have proposed to approximate global illumination with computationally tractable techniques such as obscurances and ambient occlusion. We have also studied how important steps of the volume rendering pipeline, such as the transfer function definition, can be done in a completely automatic manner. Moreover, we have studied how volume exploration techniques can be improved by selecting the best viewpoints of the model or automatizing illustrative strategies such as exploded views. A common theme in our techniques is information theory.

Below we explain in more detail our contributions and the publications derived from them.

9.1 Contributions

The main objective of this thesis was to develop new strategies to enhance the volume rendering process and the way to explore data, with special interest on medical environments. This objective has been achieved with the following contributions:

- We have adapted obscurances to volume rendering.

Obscurances, from which ambient occlusion is a particular case, is a technology that produces natural-looking lighting effects in a faster way than global illumination. Its application in volume visualization is of special interest since it permits us to generate a high quality rendering at a low cost. We have proposed an obscurance-based framework that allows us to obtain realistic and illustrative volume visualizations in an interactive manner. Obscurances can include color bleeding effects without additional cost. Moreover, we obtain a saliency map from the gradient of obscurances and we show its application to enhance volume visualization and to select the most salient views.

This work, titled *Obscurance-based volume rendering framework*, has been published in Eurographics/IEEE VGTC on Volume and Point-Based Graphics, pages 113–120, 2008. [Ruiz 2008a]

- We have automatized the creation of exploded views of volume models and used entropy rate to select the most structured views.

Exploded views are often used in illustration to overcome the problem of occlusion when depicting complex structures. We have proposed a volume visualization technique inspired by exploded views that partitions the volume into a number of parallel slabs and shows them apart from each other. The thickness of slabs is driven by the similarity between partitions. We use an information-theoretic technique for the generation of exploded views. First, the algorithm identifies the viewpoint which gives the most structured view of the data. Then, the partition of the volume into the most informative slabs for exploding is obtained using two complementary similarity-based strategies. The number of slabs and the similarity parameter are freely adjustable by the user.

This work, titled *Similarity-based exploded views*, has been published in Smart Graphics, volume 5166 of *Lecture Notes in Computer Science*, pages 154–165. Springer Berlin / Heidelberg, 2008. [Ruiz 2008b]

- We have developed a filtering-based technique to produce illustrative renderings.

We have proposed a simple and interactive technique for volume illustration by using the difference between the original intensity values and a low-pass filtered copy. This difference, known as unsharped mask, provides us with a spatial importance map that captures salient and separability information about regions in the volume. We integrate this map in the visualization pipeline and use it to modulate the color and the opacity assigned by the transfer function to produce different illustrative effects. We also apply stipple rendering modulating the density of the dots with the spatial importance map. The core of our approach is the computation of a 3D Gaussian filter, which is equivalent to three consecutive 1D filters. This separability feature allows us to obtain interactive rates with a CUDA implementation. We show results of our approach for different data sets.

This work, titled *Interactive volume illustration using intensity filtering*, has been published in Workshop on Computational Aesthetics, pages 51–58, 2010. [Ruiz

2010a]

- We have used information theory to produce realistic and illustrative renderings and to select the best viewpoints.

We have developed a volume rendering framework based on the information channel constructed between the volumetric data set and a set of viewpoints. From this channel, the information associated to each voxel can be interpreted as an ambient occlusion value that allows to obtain illustrative volume visualizations. The use of the voxel information combined with the assignment of color to each viewpoint and non-photorealistic effects produces an enhanced visualization of the volume data set. Voxel information is also applied to modulate the transfer function and to select the most informative views.

This work, titled *Viewpoint information channel for illustrative volume rendering*, has been published in *Computers & Graphics*, vol. 34, no. 4, pages 351–360, 2010. [Ruiz 2010b]

- We have used separable filtering to compute volumetric ambient occlusion.

We have presented new algorithms to compute ambient occlusion for volumetric data. Ambient occlusion is used in video-games and film animation to mimic the indirect lighting of global illumination. We extend a novel interpretation of ambient occlusion to volumetric models, that measures how big portion of the tangent sphere of a surface belongs to the set of occluded points, and propose statistically robust estimates for the ambient occlusion value. The data needed by this estimate can be obtained by separable filtering of the voxel array. As ambient occlusion is meant to obtain global illumination effects, it can provide decisive clues in interpreting the data. The new algorithms obtain smooth shading and can be computed at interactive rates, being thus appropriate for dynamic models exploration.

This work, titled *Volumetric ambient occlusion for volumetric models*, has been published in *The Visual Computer*, vol. 26, no. 6, pages 687–695, 2010. [Ruiz 2010c]

- We have developed a semi-automatic method to create opacity transfer functions.

We have presented a framework to define transfer functions from a target distribution provided by the user. A target distribution can reflect the data importance, or highly relevant data value interval, or spatial segmentation. Our approach is based on a communication channel between a set of viewpoints and a set of bins of a volume data set, and it supports 1D as well as 2D transfer functions including the gradient information. The transfer functions are obtained by minimizing the informational divergence or Kullback-Leibler distance between the visibility distribution captured by the viewpoints and a target distribution selected by the user. The use of the derivative of the informational divergence allows for a fast optimization process. Different target distributions for 1D and 2D transfer functions are analyzed together with importance-driven and view-based techniques.

This work, titled *Automatic transfer functions based on informational divergence*, has been published in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pages 1932–1941, 2011. [Ruiz 2011]

9.2 Future work

The work that has been done in this thesis can be extended in different directions.

In the context of exploded views, we plan to do a broader study with a more exhaustive analysis of behaviour and usability of the proposed approach for different image modalities and anatomical structures. We also plan to test and validate the method in a large cohort of patients to determine its clinical effectiveness. Another idea is to study how many slices should be presented and whether additional information about the similarity distances can help the user for an optimal understanding of the volume. Also, the trade-off between quality and cost for the different parameters involved in the determination of most structured views (distances between rays, subsequences of samples on those rays, sub-sampling of intensity values) can be investigated. Moreover, the technique can be extended to be applied in medical context as a method to enhance volume data exploration by automatically determining the damaged slabs of a model. Finally, as further improvement, we can also investigate if a neural network trained with physicians knowledge can be incorporated into our algorithm as a guide.

The obscurances algorithm should be implemented taking advantage of the GPU to achieve interactive obscurance computation.

The spatial importance map may be computed using other low-pass filters, such as the trilateral filter. Results obtained with different filters should be compared.

In the context of transfer function optimization, the main idea is to define a new objective function that approximates the target distribution while maximizing the projected visibility without requiring a parameter such as β . It would also be interesting to investigate the use of the average of the informational divergences for each viewpoint. Other distance measures, such as f -divergences or Jensen-Shannon divergence, can be considered and compared to Kullback-Leibler distance. The method has margin for faster implementations using the GPU in more parts of the algorithm. Finally, new optimization methods could reduce the computational cost even more.

Bibliography

- [Behrens 1998] Uwe Behrens and Ralf Ratering. *Adding shadows to a texture-based volume renderer*. In IEEE Symposium on Volume Visualization, 1998, pages 39–46, 1998. (Cited on page 20.)
- [Bordoloi 2005] Udeepa D. Bordoloi and Han-Wei Shen. *View selection for volume rendering*. In VIS 05. IEEE Visualization, 2005, pages 487–494, 2005. (Cited on pages 33 and 34.)
- [Bruckner 2005] Stefan Bruckner and Eduard Gröller. *VolumeShop: an interactive system for direct volume illustration*. In VIS 05. IEEE Visualization, 2005, pages 671–678, 2005. (Cited on pages 54 and 61.)
- [Bruckner 2006] Stefan Bruckner, Sören Grimm, Armin Kanitsar and Eduard Gröller. *Illustrative context-preserving exploration of volume data*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 6, pages 1559–1569, 2006. (Cited on pages 22, 23 and 38.)
- [Bruckner 2010] Stefan Bruckner and Torsten Möller. *Isosurface similarity maps*. Computer Graphics Forum, vol. 29, no. 3, pages 773–782, 2010. (Cited on pages 34 and 112.)
- [Burbea 1982] Jacob Burbea and C. Radhakrishna Rao. *On the convexity of some divergence measures based on entropy functions*. IEEE Transactions on Information Theory, vol. 28, no. 3, pages 489–495, 1982. (Cited on pages 30 and 59.)
- [Busking 2008] Stef Busking, Anna Vilanova and Jarke J. van Wijk. *Particle-based non-photorealistic volume visualization*. The Visual Computer, vol. 24, no. 5, pages 335–346, 2008. (Cited on page 11.)
- [Cabral 1994] Brian Cabral, Nancy Cam and Jim Foran. *Accelerated volume rendering and tomographic reconstruction using texture mapping hardware*. In Proceedings of the 1994 symposium on volume visualization, VVS '94, pages 91–98, 1994. (Cited on page 17.)
- [Castaño Moraga 2007] Carlos Alberto Castaño Moraga, Cristophe Lenglet, Rachid Deriche and Juan Ruiz Alzola. *A Riemannian approach to anisotropic filtering of tensor fields*. Signal Processing, vol. 87, no. 2, pages 263–276, 2007. (Cited on page 10.)
- [Chan 2006] Ming-Yuen Chan, Huamin Qu, Yingcai Wu and Hong Zhou. *Viewpoint selection for angiographic volume*. In Advances in Visual Computing, volume 4291 of *Lecture Notes in Computer Science*, pages 528–537. Springer Berlin / Heidelberg, 2006. (Cited on page 34.)

- [Chen 2010] Min Chen and Heike Jänicke. *An information-theoretic framework for visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6, pages 1206–1215, 2010. (Cited on page 33.)
- [Cignoni 2005] Paolo Cignoni, Roberto Scopigno and Marco Tarini. *A simple normal enhancement technique for interactive non-photorealistic renderings*. Computers & Graphics, vol. 29, no. 1, pages 125–133, 2005. (Cited on page 96.)
- [Cocosco 1997] Chris A. Cocosco, Vasken Kollokian, Remi K.-S. Kwan, G. Bruce Pike and Alan C. Evans. *Brainweb: online interface to a 3D MRI simulated brain database*. NeuroImage, vol. 5, no. 4, part 2, page S425, 1997. (Cited on page 126.)
- [Correa 2006] Carlos D. Correa, Deborah Silver and Min Chen. *Feature aligned volume manipulation for illustration and visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pages 1069–1076, 2006. (Cited on page 22.)
- [Correa 2009a] Carlos D. Correa and Kwan-Liu Ma. *The occlusion spectrum for volume classification and visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 6, pages 1465–1472, 2009. (Cited on page 18.)
- [Correa 2009b] Carlos D. Correa and Kwan-Liu Ma. *Visibility-driven transfer functions*. In PacificVis '09. IEEE Pacific Visualization Symposium, 2009, pages 177–184, 2009. (Cited on pages 18 and 112.)
- [Correa 2011] Carlos D. Correa and Kwan-Liu Ma. *Visibility histograms and visibility-driven transfer functions*. IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 2, pages 192–204, 2011. (Cited on pages 18, 112, 113 and 126.)
- [Cover 1991] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications. Wiley Online Library, 1991. (Cited on pages 24, 26, 27, 29, 30, 31, 32, 56, 57, 115, 120 and 128.)
- [Csébfalvi 2001] Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas H. König and Eduard Gröller. *Fast visualization of object contours by non-photorealistic volume rendering*. Computer Graphics Forum, vol. 20, no. 3, pages 452–460, 2001. (Cited on pages 11, 12 and 22.)
- [Csébfalvi 2003] Balázs Csébfalvi and László Szirmay-Kalos. *Monte Carlo volume rendering*. In VIS 2003. IEEE Visualization, 2003, pages 449–456, 2003. (Cited on page 12.)
- [Csiszár 2004] Imre Csiszár and Paul C. Shields. *Information theory and statistics: a tutorial*. Foundations and TrendsTM in Communications and Information Theory, vol. 1, no. 4, pages 417–528, 2004. (Cited on page 27.)

- [Desgranges 2007] Philippe Desgranges and Klaus Engel. *Fast ambient occlusion for direct volume rendering*, 2007. (Cited on page 20.)
- [Deussen 2000] Oliver Deussen, Stefan Hiller, Cornelius van Overveld and Thomas Strothotte. *Floating points: a method for computing stipple drawings*. Computer Graphics Forum, vol. 19, no. 3, pages 41–50, 2000. (Cited on page 107.)
- [Feixas 1999] Miquel Feixas, Esteve del Acebo, Philippe Bekaert and Mateu Sbert. *An information theory framework for the analysis of scene complexity*. Computer Graphics Forum, vol. 18, no. 3, pages 95–106, 1999. (Cited on page 33.)
- [Feixas 2009] Miquel Feixas, Mateu Sbert and Francisco González. *A unified information-theoretic framework for viewpoint selection and mesh saliency*. ACM Transactions on Applied Perception, vol. 6, no. 1, pages 1:1–1:23, 2009. (Cited on pages 33, 34, 36, 50 and 73.)
- [Feldman 1998] David P. Feldman and James Patrick Crutchfield. *Discovering noncritical organization: statistical mechanical, information theoretic, and computational views of patterns in one-dimensional spin systems*. Santa Fe Institute Working Paper 98-04-026, 1998. (Cited on page 31.)
- [Feldman 2002] David P. Feldman. *A brief introduction to: information theory, excess entropy and computational mechanics*, 2002. (Cited on page 31.)
- [Feldman 2003] David P. Feldman and James Patrick Crutchfield. *Structural information in two-dimensional patterns: entropy convergence and excess entropy*. Physical Review E, vol. 67, no. 5, page 051104, 2003. (Cited on pages 56 and 57.)
- [Gerig 1992] Guido Gerig, Olaf Kübler, Ron Kikinis and Ferenc A. Jolesz. *Nonlinear anisotropic filtering of MRI data*. IEEE Transactions on Medical Imaging, vol. 11, no. 2, pages 221–232, 1992. (Cited on page 10.)
- [González 2008] Francisco González, Mateu Sbert and Miquel Feixas. *Viewpoint-based ambient occlusion*. IEEE Computer Graphics and Applications, vol. 28, no. 2, pages 44–51, 2008. (Cited on pages 33, 34, 39 and 75.)
- [Gooch 1998] Amy Gooch, Bruce Gooch, Peter Shirley and Elaine Cohen. *A non-photorealistic lighting model for automatic technical illustration*. In Proceedings of the 25th annual conference on computer graphics and interactive techniques, SIGGRAPH '98, pages 447–452, New York, New York, USA, 1998. (Cited on page 47.)
- [Gruen 2008] Holger Gruen. *Approximate cumulative distribution function shadow mapping*. In ShaderX6: advanced rendering techniques, chapter 4. Course Technology, 2008. (Cited on page 85.)
- [Gumhold 2002] Stefan Gumhold. *Maximum entropy light source placement*. In VIS 2002. IEEE Visualization, 2002, pages 275–282, 2002. (Cited on page 33.)

- [Hadwiger 2009] Markus Hadwiger, Patric Ljung, Christof Rezk Salama and Timo Ropinski. *Advanced illumination techniques for GPU-based volume raycasting*. In ACM SIGGRAPH 2009 Courses, SIGGRAPH '09, pages 2:1–2:166, 2009. (Cited on pages 10 and 16.)
- [Haidacher 2008] Martin Haidacher, Stefan Bruckner, Armin Kanitsar and Eduard Gröller. *Information-based transfer functions for multimodal visualization*. In Eurographics Workshop on Visual Computing for Biomedicine, pages 101–108, 2008. (Cited on pages 33 and 112.)
- [Hauser 2001] Helwig Hauser, Lukas Mroz, Gian Italo Bischi and Eduard Gröller. *Two-level volume rendering*. IEEE Transactions on Visualization and Computer Graphics, vol. 7, no. 3, pages 242–252, 2001. (Cited on page 22.)
- [He 1996] Taosong He, Lichan Hong, Arie E. Kaufman and Hanspeter Pfister. *Generation of transfer functions with stochastic search techniques*. In Visualization '96. Proceedings, pages 227–234, 1996. (Cited on pages 18 and 112.)
- [Hernell 2007] Frida Hernell, Patric Ljung and Anders Ynnerman. *Efficient ambient and emissive tissue illumination using local occlusion in multiresolution volume rendering*. In Eurographics/IEEE VGTC Symposium on Volume Graphics, pages 1–8, 2007. (Cited on pages 20, 68 and 75.)
- [Hladůvka 2000] Jiří Hladůvka, Andreas H. König and Eduard Gröller. *Curvature-based transfer functions for direct volume rendering*. In Spring Conference on Computer Graphics, pages 58–65, 2000. (Cited on pages 17 and 112.)
- [Iones 2003] Andrey Iones, Anton Krupkin, Mateu Sbert and Sergey Zhukov. *Fast, realistic lighting for video games*. IEEE Computer Graphics and Applications, vol. 23, no. 3, pages 54–64, 2003. (Cited on pages 21, 44, 65, 70 and 81.)
- [Isenberg 2006] Tobias Isenberg and Angela Brennecke. *G-strokes: a concept for simplifying line stylization*. Computers & Graphics, vol. 30, no. 5, pages 754–766, 2006. (Cited on page 11.)
- [Itti 2001] Laurent Itti and Christof Koch. *Computational modelling of visual attention*. Nature Reviews Neuroscience, vol. 2, no. 3, pages 194–203, 2001. (Cited on pages 66, 73 and 95.)
- [Jainek 2008] Werner M. Jainek, Silvia Born, Dirk Bartz, Wolfgang Straßer and Jan Fischer. *Illustrative hybrid visualization and exploration of anatomical and functional brain data*. Computer Graphics Forum, vol. 27, no. 3, pages 855–862, 2008. (Cited on page 20.)
- [Ji 2006] Guangfeng Ji and Han-Wei Shen. *Dynamic view selection for time-varying volumes*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pages 1109–1116, 2006. (Cited on page 33.)

- [Kanda 2002] Ken-ichi Kanda, Shinobu Mizuta and Tetsuya Matsuda. *Volume visualization using relative distance among voxels*. In *Medical Imaging 2002: Visualization, Image-Guided Procedures, and Display*, volume 4681, pages 641–648, 2002. (Cited on page 18.)
- [Kaufman 1990] Arie E. Kaufman. *Volume visualization*. IEEE Computer Society Press, 1990. (Cited on page 7.)
- [Kim 2006] Youngmin Kim and Amitabh Varshney. *Saliency-guided enhancement for volume visualization*. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pages 925–932, 2006. (Cited on pages 73 and 95.)
- [Kindlmann 1998] Gordon Kindlmann and James W. Durkin. *Semi-automatic generation of transfer functions for direct volume rendering*. In *IEEE Symposium on Volume Visualization*, 1998, pages 79–86, 1998. (Cited on pages 17 and 112.)
- [Kindlmann 2003] Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen and Torsten Möller. *Curvature-based transfer functions for direct volume rendering: methods and applications*. In *VIS 2003*. *IEEE Visualization*, 2003, pages 513–520, 2003. (Cited on pages 17, 18, 22 and 112.)
- [Kniss 2002a] Joe Kniss, Gordon Kindlmann and Charles Hansen. *Multidimensional transfer functions for interactive volume rendering*. *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pages 270–285, 2002. (Cited on pages 18 and 95.)
- [Kniss 2002b] Joe Kniss, Simon Premože, Charles Hansen and David S. Ebert. *Interactive translucent volume rendering and procedural modeling*. In *VIS 2002*. *IEEE Visualization*, 2002, pages 109–116, 2002. (Cited on page 20.)
- [Kniss 2003] Joe Kniss, Simon Premože, Milan Ikits, Aaron Lefohn, Charles Hansen and Emil Praun. *Gaussian transfer functions for multi-field volume visualization*. In *VIS 2003*. *IEEE Visualization*, 2003, pages 497–504, 2003. (Cited on page 18.)
- [Kohlmann 2007] Peter Kohlmann, Stefan Bruckner, Armin Kanitsar and Eduard Gröller. *LiveSync: deformed viewing spheres for knowledge-based navigation*. *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pages 1544–1551, 2007. (Cited on page 132.)
- [König 2001] Andreas H. König and Eduard Gröller. *Mastering transfer function specification by using VolumePro technology*. In *Spring Conference on Computer Graphics*, pages 279–286, 2001. (Cited on page 18.)
- [Konrad-Verse 2004] Olaf Konrad-Verse, Bernhard Preim and Arne Littmann. *Virtual resection with a deformable cutting plane*. In *Simulation und Visualisierung 2004 (SimVis 2004)*, pages 203–214, 2004. (Cited on page 23.)

- [Kronander 2011] Joel Kronander, Daniel Jönsson, Joakim Löw, Patric Ljung, Anders Ynnerman and Jonas Unger. *Efficient visibility encoding for dynamic illumination in direct volume rendering*. IEEE Transactions on Visualization and Computer Graphics, 2011. (Cited on page 20.)
- [Krüger 2003] Jens Krüger and Rüdiger Westermann. *Acceleration techniques for GPU-based volume rendering*. In VIS 2003. IEEE Visualization, 2003, pages 287–292, 2003. (Cited on page 16.)
- [Kullback 1951] Solomon Kullback and Richard A. Leibler. *On information and sufficiency*. The Annals of Mathematical Statistics, vol. 22, no. 1, pages 79–86, 1951. (Cited on page 27.)
- [Lacroute 1994] Philippe Lacroute and Marc Levoy. *Fast volume rendering using a shear-warp factorization of the viewing transformation*. In Proceedings of the 21st annual conference on computer graphics and interactive techniques, SIGGRAPH '94, pages 451–458, 1994. (Cited on page 16.)
- [Landis 2002] Hayden Landis. *Production-ready global illumination*. In SIGGRAPH 2002 Course 16, pages 87–101, 2002. (Cited on pages 20, 21, 22, 37, 38, 42, 44, 67 and 81.)
- [Lee 2005] Chang Ha Lee, Amitabh Varshney and David W. Jacobs. *Mesh saliency*. ACM Transactions on Graphics, vol. 24, no. 3, pages 659–666, 2005. (Cited on pages 73, 75 and 77.)
- [Lee 2011] Teng-Yok Lee, Oleg Mishchenko, Han-Wei Shen and Roger Crawfis. *View point evaluation and streamline filtering for flow visualization*. In 2011 IEEE Pacific Visualization Symposium (PacificVis), pages 83–90, 2011. (Cited on page 33.)
- [Levoy 1988] Marc Levoy. *Display of surfaces from volume data*. IEEE Computer Graphics and Applications, vol. 8, no. 3, pages 29–37, 1988. (Cited on pages 15, 17, 41, 112 and 114.)
- [Lorensen 1987] William E. Lorensen and Harvey E. Cline. *Marching cubes: a high resolution 3D surface construction algorithm*. SIGGRAPH Computer Graphics, vol. 21, no. 4, pages 163–169, 1987. (Cited on page 11.)
- [Lu 2002] Aidong Lu, Christopher J. Morris, David S. Ebert, Penny Rheingans and Charles Hansen. *Non-photorealistic volume rendering using stippling techniques*. In VIS 2002. IEEE Visualization, 2002, pages 211–218, 2002. (Cited on page 22.)
- [Luft 2006] Thomas Luft, Carsten Colditz and Oliver Deussen. *Image enhancement by unsharp masking the depth buffer*. ACM Transactions on Graphics, vol. 25, no. 3, pages 1206–1213, 2006. (Cited on pages 96 and 97.)

- [Lum 2004] Eric B. Lum and Kwan-Liu Ma. *Lighting transfer functions using gradient aligned sampling*. In IEEE Visualization, 2004, pages 289–296, 2004. (Cited on page 18.)
- [Lundström 2006] Claes Lundström, Patric Ljung and Anders Ynnerman. *Local histograms for design of transfer functions in direct volume rendering*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 6, pages 1570–1579, 2006. (Cited on page 18.)
- [Maciejewski 2009] Ross Maciejewski, Insoo Woo, Wei Chen and David S. Ebert. *Structuring feature space: a non-parametric method for volumetric transfer function generation*. IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 6, pages 1473–1480, 2009. (Cited on page 18.)
- [Maes 1997] Frederik Maes, André Collignon, Dirk Vandermeulen, Guy Marchal and Paul Suetens. *Multimodality image registration by maximization of mutual information*. IEEE Transactions on Medical Imaging, vol. 16, no. 2, pages 187–198, 1997. (Cited on page 60.)
- [Malzbender 1993] Tom Malzbender. *Fourier volume rendering*. ACM Transactions on Graphics, vol. 12, no. 3, pages 233–250, 1993. (Cited on page 12.)
- [Marks 1997] Joe William Marks, Brad Andalman, Paul A. Beardsley, William T. Freeman, Sarah F. Frisken Gibson, Jessica Kate Hodgins, Tae Ik Kang, Brian Vincent Mirtich, Hanspeter Pfister, Wheeler Ruml, Kathy Ryall, Joshua E. Seims and Stuart M. Shieber. *Design Galleries: a general approach to setting parameters for computer graphics and animation*. In Proceedings of the 24th annual conference on computer graphics and interactive techniques, SIGGRAPH '97, pages 389–400, 1997. (Cited on pages 18 and 112.)
- [Mavridis 2010] Pavlos Mavridis, Athanasios Gaitatzes and Georgios Papaioannou. *Volume-based diffuse global illumination*. In Proceedings of the IADIS International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, pages 31–38, 2010. (Cited on page 20.)
- [Max 1995] Nelson Max. *Optical models for direct volume rendering*. IEEE Transactions on Visualization and Computer Graphics, vol. 1, no. 2, pages 99–108, 1995. (Cited on pages 12 and 65.)
- [McGuffin 2003] Michael J. McGuffin, Liviu Tancu and Ravin Balakrishnan. *Using deformations for browsing volumetric data*. In VIS 2003. IEEE Visualization, 2003, pages 401–408, 2003. (Cited on page 23.)
- [Meißner 2000] Michael Meißner, Jian Huang, Dirk Bartz, Klaus Mueller and Roger Crawfis. *A practical evaluation of popular volume rendering algorithms*. In Proceedings of the 2000 IEEE symposium on volume visualization, VVS '00, pages 81–90, 2000. (Cited on page 14.)

- [Méndez 2003a] Àlex Méndez, Mateu Sbert and Jordi Catà. *Real-time obscurances with color bleeding*. In Proceedings of the 19th spring conference on computer graphics, SCCG '03, pages 171–176, 2003. (Cited on pages 66, 67 and 70.)
- [Méndez 2003b] Àlex Méndez, Mateu Sbert and László Neumann. *Obscurances for ray-tracing*. In Proceedings of Eurographics 2003 (short papers), 2003. (Cited on page 21.)
- [Méndez 2009] Àlex Méndez and Mateu Sbert. *From obscurances to ambient occlusion: a survey*. The Visual Computer, vol. 25, no. 2, pages 181–196, 2009. (Cited on page 21.)
- [Mueller 1999] Klaus Mueller, Naeem Shareef, Jian Huang and Roger Crawfis. *High-quality splatting on rectilinear grids with efficient culling of occluded voxels*. IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 2, pages 116–134, 1999. (Cited on page 16.)
- [Neophytou 2005] Neophytos Neophytou and Klaus Mueller. *GPU accelerated image aligned splatting*. In Fourth International Workshop on Volume Graphics, 2005, pages 197–242, 2005. (Cited on page 16.)
- [Neophytou 2006] Neophytos Neophytou, Klaus Mueller, Kevin T. McDonnell, Wei Hong, Xin Guan, Hong Qin and Arie E. Kaufman. *GPU-accelerated volume splatting with elliptical RBFs*. In EUROVIS - Eurographics/IEEE VGTC Symposium on Visualization, pages 13–20, 2006. (Cited on page 16.)
- [Neumann 2000] László Neumann, Balázs Csébfalvi, Andreas H. König and Eduard Gröller. *Gradient estimation in volume data using 4D linear regression*. Computer Graphics Forum, vol. 19, no. 3, pages 351–358, 2000. (Cited on pages 75 and 90.)
- [Newman 2006] Timothy S. Newman and Hong Yi. *A survey of the marching cubes algorithm*. Computers & Graphics, vol. 30, no. 5, pages 854–879, 2006. (Cited on page 11.)
- [Pavlík 2009] Ivo Pavlík. *Advanced volume ray casting on GPU*. PhD thesis, Charles University in Prague, 2009. (Cited on page 16.)
- [Penner 2008] Eric Penner and Ross Mitchell. *Isosurface ambient occlusion and soft shadows with filterable occlusion maps*. In Eurographics/IEEE VGTC on Volume and Point-Based Graphics, pages 57–64, 2008. (Cited on pages 81, 83 and 87.)
- [Pfister 2001] Hanspeter Pfister, William E. Lorensen, Chandrajit Bajaj, Gordon Kindlmann, Will Schroeder, Lisa Sobierajski Avila, Ken Martin, Raghu Machiraju and Jinho Lee. *The transfer function bake-off*. IEEE Computer Graphics and Applications, vol. 21, no. 3, pages 16–22, 2001. (Cited on pages 17 and 112.)

- [Pluim 2003] Josien P. W. Pluim, J. B. Antoine Maintz and Max A. Viergever. *Mutual information based registration of medical images: a survey*. IEEE Transactions on Medical Imaging, vol. 22, no. 8, pages 986–1004, 2003. (Cited on page 10.)
- [Press 1992] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, second edition, 1992. (Cited on page 121.)
- [Rheingans 2001] Penny Rheingans and David S. Ebert. *Volume illustration: nonphotorealistic rendering of volume models*. IEEE Transactions on Visualization and Computer Graphics, vol. 7, no. 3, pages 253–264, 2001. (Cited on pages 22, 54 and 65.)
- [Rigau 2003] Jaume Rigau, Miquel Feixas and Mateu Sbert. *Refinement criteria based on f -divergences*. In EGSR03 14th Eurographics Symposium on Rendering, pages 260–269, 2003. (Cited on page 33.)
- [Ritschel 2008] Tobias Ritschel, Kaleigh Smith, Matthias Ihrke, Thorsten Grosch, Karol Myszkowski and Hans-Peter Seidel. *3D unsharp masking for scene coherent enhancement*. ACM Transactions on Graphics, vol. 27, no. 3, pages 90:1–90:8, 2008. (Cited on page 96.)
- [Roettger 2005] Stefan Roettger, Michael Bauer and Marc Stamminger. *Spatialized transfer functions*. In EUROVIS 2005: Eurographics/IEEE VGTC Symposium on Visualization, pages 271–278, 2005. (Cited on page 18.)
- [Ropinski 2008] Timo Ropinski, Jennis Meyer-Spradow, Stefan Diepenbrock, Jörg Mensmann and Klaus Hinrichs. *Interactive volume rendering with dynamic ambient occlusion and color bleeding*. Computer Graphics Forum, vol. 27, no. 2, pages 567–576, 2008. (Cited on pages 20, 68 and 75.)
- [Ruijters 2008] Daniel Ruijters, Bart M. ter Haar Romeny and Paul Suetens. *Efficient GPU-based texture interpolation using uniform B-splines*. journal of graphics, gpu, and game tools, vol. 13, no. 4, pages 61–69, 2008. (Cited on page 10.)
- [Ruiz 2008a] Marc Ruiz, Imma Boada, Ivan Viola, Stefan Bruckner, Miquel Feixas and Mateu Sbert. *Obscurance-based volume rendering framework*. In Eurographics/IEEE VGTC on Volume and Point-Based Graphics, pages 113–120, 2008. (Cited on page 136.)
- [Ruiz 2008b] Marc Ruiz, Ivan Viola, Imma Boada, Stefan Bruckner, Miquel Feixas and Mateu Sbert. *Similarity-based exploded views*. In Smart Graphics, volume 5166 of *Lecture Notes in Computer Science*, pages 154–165. Springer Berlin / Heidelberg, 2008. (Cited on page 136.)
- [Ruiz 2010a] Marc Ruiz, Imma Boada, Miquel Feixas and Mateu Sbert. *Interactive volume illustration using intensity filtering*. In Workshop on Computational Aesthetics, pages 51–58, 2010. (Cited on page 136.)

- [Ruiz 2010b] Marc Ruiz, Imma Boada, Miquel Feixas and Mateu Sbert. *Viewpoint information channel for illustrative volume rendering*. *Computers & Graphics*, vol. 34, no. 4, pages 351–360, 2010. (Cited on page 137.)
- [Ruiz 2010c] Marc Ruiz, László Szirmay-Kalos, Tamás Umenhoffer, Imma Boada, Miquel Feixas and Mateu Sbert. *Volumetric ambient occlusion for volumetric models*. *The Visual Computer*, vol. 26, no. 6, pages 687–695, 2010. (Cited on page 137.)
- [Ruiz 2011] Marc Ruiz, Anton Bardera, Imma Boada, Ivan Viola, Miquel Feixas and Mateu Sbert. *Automatic transfer functions based on informational divergence*. *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pages 1932–1941, 2011. (Cited on page 138.)
- [Rusinkiewicz 2000] Szymon Rusinkiewicz and Marc Levoy. *QSplat: a multiresolution point rendering system for large meshes*. In *Proceedings of the 27th annual conference on computer graphics and interactive techniques, SIGGRAPH '00*, pages 343–352, 2000. (Cited on page 12.)
- [Salama 2006] Christof Rezk Salama, Maik Keller and Peter Kohlmann. *High-level user interfaces for transfer function design with semantics*. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pages 1021–1028, 2006. (Cited on page 18.)
- [Sbert 2005] Mateu Sbert, Dimitri Plemenos, Miquel Feixas and Francisco González. *Viewpoint quality: measures and applications*. In *Workshop on Computational Aesthetics*, pages 185–192, 2005. (Cited on page 33.)
- [Sbert 2009] Mateu Sbert, Miquel Feixas, Jaume Rigau, Miguel Chover and Ivan Viola. *Information theory tools for computer graphics*, volume 12 of *Synthesis Lectures on Computer Graphics and Animation*. Morgan & Claypool, 2009. (Cited on page 33.)
- [Schlechtweg 2005] Stefan Schlechtweg, Tobias Germer and Thomas Strothotte. *RenderBots—multi-agent systems for direct image generation*. *Computer Graphics Forum*, vol. 24, no. 2, pages 137–148, 2005. (Cited on page 108.)
- [Secord 2002] Adrian Secord. *Weighted Voronoi stippling*. In *Proceedings of the 2nd international symposium on non-photorealistic animation and rendering, NPAR '02*, pages 37–43, 2002. (Cited on page 107.)
- [Šereda 2006a] Petr Šereda, Anna Vilanova and Frans A. Gerritsen. *Automating transfer function design for volume rendering using hierarchical clustering of material boundaries*. In *EUROVIS - Eurographics/IEEE VGTC Symposium on Visualization*, pages 243–250, 2006. (Cited on page 19.)
- [Šereda 2006b] Petr Šereda, Anna Vilanova, Iwo W. O. Serlie and Frans A. Gerritsen. *Visualization of boundaries in volumetric data sets using LH histograms*. *IEEE*

- Transactions on Visualization and Computer Graphics, vol. 12, no. 2, pages 208–218, 2006. (Cited on page 18.)
- [Shannon 1948] Claude Elwood Shannon. *A mathematical theory of communication*. Bell System Technical Journal, vol. 27, no. 3 and 4, pages 379–423 and 623–656, 1948. (Cited on pages 24 and 25.)
- [Sharma 2010] Neeraj Sharma and Lalit M. Aggarwal. *Automated medical image segmentation techniques*. Journal of Medical Physics, vol. 35, no. 1, pages 3–14, 2010. (Cited on page 11.)
- [Slonim 1999] Noam Slonim and Naftali Tishby. *Agglomerative information bottleneck*. In Advances in Neural Information Processing Systems (NIPS), volume 12, pages 617–623, 1999. (Cited on pages 32, 55 and 59.)
- [Slonim 2000] Noam Slonim and Naftali Tishby. *Document clustering using word clusters via the information bottleneck method*. In Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '00, pages 208–215, 2000. (Cited on page 30.)
- [Sokolov 2006] Dmitry Sokolov, Dimitri Plemenos and Karim Tamine. *Methods and data structures for virtual world exploration*. The Visual Computer, vol. 22, no. 7, pages 506–516, 2006. (Cited on page 34.)
- [Šoltészová 2010] Veronika Šoltészová, Daniel Patel, Stefan Bruckner and Ivan Viola. *A multidirectional occlusion shading model for direct volume rendering*. Computer Graphics Forum, vol. 29, no. 3, pages 883–891, 2010. (Cited on page 20.)
- [Sousa 2003] Mario Costa Sousa, Kevin Foster, Brian Wyvill and Faramarz Samavati. *Precise ink drawing of 3D models*. Computer Graphics Forum, vol. 22, no. 3, pages 369–379, 2003. (Cited on page 108.)
- [Stewart 2003] A. James Stewart. *Vicinity shading for enhanced perception of volumetric data*. In VIS 2003. IEEE Visualization, 2003, pages 355–362, 2003. (Cited on pages 20, 38, 42, 44, 66, 67, 68, 81 and 87.)
- [Studholme 1997] Colin Studholme. *Measures of 3D medical image alignment*. PhD thesis, London University, 1997. (Cited on page 60.)
- [Szirmay-Kalos 2010] László Szirmay-Kalos, Tamás Umenhoffer, Balázs Tóth, László Szécsi and Mateu Sbert. *Volumetric ambient occlusion for real-time rendering and games*. IEEE Computer Graphics and Applications, vol. 30, no. 1, pages 70–79, 2010. (Cited on pages 81, 82 and 83.)
- [Takahashi 2005] Shigeo Takahashi, Issei Fujishiro, Yuriko Takeshima and Tomoyuki Nishita. *A feature-driven approach to locating optimal viewpoints for volume visualization*. In VIS 05. IEEE Visualization, 2005, pages 495–502, 2005. (Cited on pages 33 and 34.)

- [Tao 2009] Yubo Tao, Hai Lin, Hujun Bao, Feng Dong and Gordon Clapworthy. *Feature enhancement by volumetric unsharp masking*. The Visual Computer, vol. 25, no. 5, pages 581–588, 2009. (Cited on pages 96 and 97.)
- [Tarini 2006] Marco Tarini, Paolo Cignoni and Claudio Montani. *Ambient occlusion and edge cueing to enhance real time molecular visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pages 1237–1244, 2006. (Cited on page 20.)
- [Tiede 1990] Ulf Tiede, Karl Heinz Hoehne, Michael Bomans, Andreas Pommert, Martin Riemer and Gunnar Wiebecke. *Investigation of medical 3D-rendering algorithms*. IEEE Computer Graphics and Applications, vol. 10, no. 2, pages 41–53, 1990. (Cited on page 12.)
- [Tishby 1999] Naftali Tishby, Fernando C. Pereira and William Bialek. *The information bottleneck method*. In Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing, pages 368–377, 1999. (Cited on pages 32, 55 and 59.)
- [Tsotsos 2005] John K. Tsotsos, Laurent Itti and Geraint Rees. *A brief and selective history of attention*. In Neurobiology of attention, pages xxiii–xxxii. Elsevier Academic Press, 2005. (Cited on pages 66 and 75.)
- [van de Weijer 2006] Joost van de Weijer, Theo Gevers and Andrew D. Bagdanov. *Boosting color saliency in image feature detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pages 150–156, 2006. (Cited on pages 66 and 75.)
- [van Pelt 2010] Roy van Pelt, Anna Vilanova and Huub van de Wetering. *Illustrative volume visualization using GPU-based particle systems*. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 4, pages 571–582, 2010. (Cited on page 11.)
- [Vázquez 2001] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert and Wolfgang Heidrich. *Viewpoint selection using viewpoint entropy*. In Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV-01), pages 273–280, 2001. (Cited on pages 33, 34 and 115.)
- [Vázquez 2003] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert and Wolfgang Heidrich. *Automatic view selection using viewpoint entropy and its application to image-based modelling*. Computer Graphics Forum, vol. 22, no. 4, pages 689–700, 2003. (Cited on page 34.)
- [Verdú 1998] Sergio Verdú. *Fifty years of Shannon theory*. IEEE Transactions on Information Theory, vol. 44, no. 6, pages 2057–2078, 1998. (Cited on page 24.)
- [Viola 1995] Paul A. Viola. *Alignment by maximization of mutual information*. PhD thesis, Massachusetts Institute of Technology, 1995. (Cited on page 60.)

- [Viola 2004] Ivan Viola, Armin Kanitsar and Eduard Gröller. *Importance-driven volume rendering*. In IEEE Visualization, 2004, pages 139–145, 2004. (Cited on page 22.)
- [Viola 2005] Ivan Viola, Armin Kanitsar and Eduard Gröller. *Importance-driven feature enhancement in volume visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 4, pages 408–418, 2005. (Cited on page 23.)
- [Viola 2006] Ivan Viola, Miquel Feixas, Mateu Sbert and Eduard Gröller. *Importance-driven focus of attention*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pages 933–940, 2006. (Cited on pages 33, 34, 35, 36, 37, 38, 51, 113 and 115.)
- [Wang 1995] Sidney W. Wang and Arie E. Kaufman. *Volume sculpting*. In Proceedings of the 1995 symposium on Interactive 3D graphics, I3D '95, pages 151–156 and 214, 1995. (Cited on page 23.)
- [Wang 2006] Chaoli Wang and Han-Wei Shen. *LOD map - a visual interface for navigating multiresolution volume visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 5, pages 1029–1036, 2006. (Cited on page 33.)
- [Wang 2008] Chaoli Wang, Hongfeng Yu and Kwan-Liu Ma. *Importance-driven time-varying data visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 14, no. 6, pages 1547–1554, 2008. (Cited on page 33.)
- [Wang 2010] Yunhai Wang, Wei Chen, Guihua Shan, Tingxin Dong and Xuebin Chi. *Volume exploration using ellipsoidal Gaussian transfer functions*. In 2010 IEEE Pacific Visualization Symposium (PacificVis), pages 25–32, 2010. (Cited on page 19.)
- [Wang 2011] Chaoli Wang and Han-Wei Shen. *Information theory in scientific visualization*. Entropy, vol. 13, no. 1, pages 254–273, 2011. (Cited on pages 33 and 115.)
- [Weiskopf 2003] Daniel Weiskopf, Klaus Engel and Thomas Ertl. *Interactive clipping techniques for texture-based volume visualization and volume shading*. IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 3, pages 298–312, 2003. (Cited on page 23.)
- [Westover 1992] Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. PhD thesis, University of North Carolina at Chapel Hill, 1992. (Cited on page 16.)
- [Winitzki 2008] Sergei Winitzki. *A handy approximation for the error function and its inverse*, 2008. (Cited on page 86.)

- [Wirjadi 2007] Oliver Wirjadi. *Survey of 3D image segmentation methods*. Technical report 123, Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM, 2007. (Cited on page 11.)
- [Withey 2007] Daniel J. Withey and Zoltan Joseph Koles. *Medical image segmentation: methods and software*. In NFSI-ICFBI 2007. Joint Meeting of the 6th International Symposium on Noninvasive Functional Source Imaging of the Brain and Heart and the International Conference on Functional Biomedical Imaging, 2007, pages 140–143, October 2007. (Cited on page 11.)
- [Wu 2007] Yingcai Wu and Huamin Qu. *Interactive transfer function design based on editing direct volume rendered images*. IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 5, pages 1027–1040, 2007. (Cited on pages 18 and 112.)
- [Wyman 2006] Chris Wyman, Steven Parker, Peter Shirley and Charles Hansen. *Interactive display of isosurfaces with global illumination*. IEEE Transactions on Visualization and Computer Graphics, vol. 12, no. 2, pages 186–196, 2006. (Cited on page 20.)
- [Xu 2010] Lijie Xu, Teng-Yok Lee and Han-Wei Shen. *An information-theoretic framework for flow visualization*. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6, pages 1216–1224, 2010. (Cited on page 33.)
- [Yeung 2008] Raymond W. Yeung. *Information theory and network coding*. Springer, 2008. (Cited on pages 24, 27, 28, 30, 31, 115, 120 and 128.)
- [Zhang 2003] Caixia Zhang and Roger Crawfis. *Shadows and soft shadows with participating media using splatting*. IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 2, pages 139–149, 2003. (Cited on page 20.)
- [Zhou 2009] Jianlong Zhou and Masahiro Takatsuka. *Automatic transfer function generation using contour tree controlled residue flow model and color harmonics*. IEEE Transactions on Visualization and Computer Graphics, vol. 15, no. 6, pages 1481–1488, 2009. (Cited on page 19.)
- [Zhukov 1998] Sergey Zhukov, Andrey Iones and Grigorij Kronin. *An ambient light illumination model*. In Rendering Techniques '98, Proceedings of the Eurographics Workshop, 1998, pages 45–56, 1998. (Cited on pages 20, 21, 37, 65, 67 and 81.)