D I S S E R T A T I O N

Importance-Driven Expressive Visualization

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

unter Anleitung von
Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Institut für Computergraphik und Algorithmen
der Technischen Universität Wien

eingereicht an der Technischen Universität Wien, Fakultät für Informatik, durch

Dipl.-Ing. Ivan Viola
Matrikelnummer: 9726070
Dietrichsteingasse 8/14
1090 Wien, Österreich
geboren am 25.06.1977

Wien, im Mai 2005

Importance-Driven Expressive Visualization

PhD Thesis

Ivan Viola
Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria
viola@cg.tuwien.ac.at

Mission Statement

Developing techniques for automatic computer generated expressive visualizations of complex volumetric data.

Abstract

In this thesis several expressive visualization techniques for volumetric data are presented. The key idea is to classify the underlying data according to its prominence on the resulting visualization by *importance* value. The importance property drives the visualization pipeline to emphasize the most prominent features and to suppress the less relevant ones. The suppression can be realized globally, so the whole object is suppressed, or locally. A local modulation generates cut-away and ghosted views because the suppression of less relevant features occurs only on the part where the occlusion of more important features appears.

Features within the volumetric data are classified according to a new dimension denoted as *object importance*. This property determines which structures should be readily discernible and which structures are less important. Next, for each feature various representations (*levels of sparseness*) from a dense to a sparse depiction are defined. Levels of sparseness define a spectrum of optical properties or rendering styles. The resulting image is generated by ray-casting and combining the intersected features proportional to their importance. An additional step to traditional volume rendering evaluates the areas of occlusion and assigns a particular level of sparseness. This step is denoted as *importance compositing*. Advanced schemes for importance compositing determine the resulting *visibility* of features and if the resulting visibility distribution does not correspond to the importance distribution different levels of sparseness are selected.

The applicability of importance-driven visualization is demonstrated on several examples from medical diagnostics scenarios, flow visualization, and interactive illustrative visualization.

Kurzfassung

In der vorliegenden Arbeit werden verschiede Verfahren zur expressiven Visualisierung von volumetrischen Daten vorgestellt. Der grundlegende Ansatz basiert auf einer Klassifikation der Daten entsprechend ihrer *Wichtigkeit*. Diese Eigenschaft wird in der Visualisierung verwendet um signifikante Merkmale in der Darstellung hervorzuheben, während weniger relevante Teile unterdrückt werden. Dies kann sowohl global, d.h. auf das gesamte Objekt bezogen, als auch lokal geschehen. Im Falle einer lokalen Modulation werden sogenannte *cut-away* und *ghosting* Ansichten generiert, da nur jene Regionen, die höherpriore Merkmale verdecken, vollständig oder teilweise unterdrückt werden.

Die Dimension der *Wichtigkeit* kann also verwendet werden um bestimmte Strukturen in Hinblick auf ihre Prominenz in der Darstellung zu kategorisieren. Zu diesem Zweck können ebenfalls verschiedene visuelle Repräsentation definiert werden. Diese sogenannten *Levels of Sparseness* verwenden unterschiedlichste Darstellungsformen und Stile, die sich in ihrer optischen Dichte unterscheiden. Das *Ray Casting* Verfahren wird eingesetzt um die definierten Strukturen gemäß ihrer Wichtigkeit zu kombinieren. Dabei werden in einem zusätzlichen Schritt, dem *Importance Compositing*, Überdeckungsbereiche identifiziert und einem *Level of Sparseness* zugeordnet. Verschiedene Verfahren des *Importance Compositing*, die sich in der Verteilung der *Levels of Sparseness* unterscheiden, wurden entwickelt.

Um die Relevanz des beschriebenen Verfahrens zu demonstrieren, werden Anwendungssmöglichkeiten dieser neuen Visualisierungstechnik an Beispielen aus der medizinischen Diagnostik, Strömungsvisualisierung und interaktiven illustrativen Visualisierung präsentiert.

Acknowledgments

This thesis would not be possible without supervision, support and motivation of the following people. At first I would like to thank for an excellent supervision done by Meister Eduard Gröller. His strange way of humor, brilliant ideas, and constant encouragement have significantly motivated me through all my PhD studies. Once more: "**Master**, thank you!"

I would like to express my thanks to my co-workers from the Visualization Group of the Institute of Computer Graphics and Algorithms, Vienna University of Technology, namely Matej Mlejnek, Sören Grimm, Stefan Bruckner, Armin Kanitsar, and Alexandra La Cruz for a very nice working environment, fruitful discussions, and excellent team-work. Additionally big thanks to Armin for the excellent management of the ADAPT research project I was funded by during my PhD studies.

Furthermore thank goes to my computer science students Martin Haidacher, Bernhard Pflugfelder, and Christopher Thurnher. Parts of their computer science projects have been used in this thesis.

I would like to express my BIG thanks to my mother and to my lovely chick, for the longstanding support and help.

The work presented in this thesis has been funded by the ADAPT project (FFF-804544). ADAPT is supported by *Tiani Medgraph*, Vienna (http://www.tiani.com), and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria.

The datasets are courtesy of Tiani Medgraph, University of the Veterinary Medicine Vienna, and National Center for Atmospheric Research in the United States. The illustrations are courtesy of Howell MediGraphics [30] and Kevin Hulsey Inc. [31].

Contents

1 Introduction **1**
- 1.1 Visualization of Liver Lesions 1
- 1.2 Lung Nodules Detection and Visualization 2
- 1.3 Inspection of Tubular Structures 3
- 1.4 Thesis Contribution 4
- 1.5 Thesis Outline 4

2 Background in Technical Illustration **7**
- 2.1 History of Illustration 7
- 2.2 Modern Illustration Techniques 13

3 State of the Art **21**
- 3.1 Feature Classification 21
- 3.2 Visual Representations 25
- 3.3 Focus+Context Visualization 38
- 3.4 Smart Visibility in Visualization 44
- 3.5 Automatic Visual Enhancements 50

4 Importance-Driven Visualization **55**
- 4.1 The Model 55
- 4.2 Importance Compositing 59
- 4.3 Levels of Sparseness 64
- 4.4 Cut-Away Generation 68
- 4.5 Results 72

5 Visibility Feasibility **77**
- 5.1 Visibility Evaluation 77
- 5.2 Visibility Preserving Visualization 80

6 Applications **87**
- 6.1 Lung Nodules Visualization 87
- 6.2 Breast Cancer Visualization 88

6.3 Interactive Importance-Driven Visualization 90

7 Summary and Conclusions **95**

Bibliography **96**

A Curriculum Vitae **105**

Chapter 1

Introduction

The relevance of volume visualization in medical, geological, and flow simulation applications has been increasing over the last years. Three-dimensional visualization is becoming an essential tool for all these application areas. The rapid development of high-precision scanning modalities, or steadily increasing simulation data leads to very large amounts of overall data often in contrast to a relatively small portion of relevant information contained within. Therefore the small, and most interesting features have to be visually emphasized. Examples are tumors in the kidneys, lesions inside the liver, and lung nodules from medical applications such as diagnosis and operation planning. In flow visualization of simulated or measured data vortex cores can be features of interest. Another example are small scale oil and gas resources in measured terabytes of geological and geographical data. All these examples clearly show the necessity of visual emphasis of the most relevant features. Focusing on medical applications it turns out that diagnostic examinations are complex tasks, where properties of the anatomical tissues have to be taken into account. In addition to the size and shape of pathologies also their spatial position and vicinity to other anatomical structures is of interest. Hence, from a computer science point of view, expressive visualization of above mentioned examples is a focus+context task. In the following several examples of medical scenarios are presented, where smart visual emphasis of important features significantly improves the diagnostic process. Another example where importance-driven feature emphasis can be effectively used is neck dissection planning [43].

1.1 Visualization of Liver Lesions

The detection of liver lesions illustrates the medical requirements on the applied visualization method. Medical experts need to see the tumor

from several directions in order to estimate the shape of the lesion. Furthermore the spatial position of arteries in close vicinity is very important in order to determine which liver segments must be removed in a possible subsequent surgical treatment. The visualization task is to display three different structures: the tumor, the vessel tree of the liver, and the liver parenchyma. However, displaying these structures simultaneously results in objects occluding each other. Traditional techniques classify objects within the dataset independently from the viewpoint. The global setting limits viewpoint positions and viewing angles to a range, where the important structures are not occluded by other objects. One possibility is to use clipping planes. But such an approach eliminates less important objects also in those viewing situations, where it would not be necessary. Different optical properties and rendering techniques (i.e., silhouette rendering) ease the problem only to a certain degree and fine-tuning of rendering parameters is a time consuming process not suitable for rapid clinical use.

Medical tasks such as visualizing liver lesions can be resolved by assigning an *importance* attribute explicitly to each feature. This attribute reflects the desired prominence of this feature in the resulting visualization. The tumor and the vascular tree in close vicinity are the most important features, the liver tissue and the surrounding anatomy (bones, aorta, skin) are of lower importance but still helpful for orientation purposes. Incorporating an importance factor interesting structures can be made clearly visible automatically and for each viewing angle. Occluding objects are rendered more sparsely or suppressed entirely.

1.2 Lung Nodules Detection and Visualization

Early stage lung cancer is hard to diagnose using traditional X-ray examinations. CT images of the chest give a much clearer view on the lungs and enable to detect lung nodules with just 3 mm in diameter. The process shall be done fully automatically starting with an automatic nodules segmentation and classification, followed by an expressive visualization to guarantee the visibility of important features. The visualization mapping of a particular feature highly depends on the probability that it is classified as a nodule. All suspicious regions are highlighted by assigning importance values proportional to the level of suspicion. The medical expert can switch between one or a group of suspicious regions. The occluding features will be represented sparsely, only the structures in close vicinity to suspicious regions, such as pulmonary arteries, will be

represented densely. The feature importance is defined by the nodule detection algorithm and according to the distance to the nodules. The occluding context information can be represented by suggestive contours or any other image-space saving technique. The expressive view is additionally linked to a slice viewer showing all three orthogonal directions. Therefore it is possible to validate the presence of the nodule on the slices, which is currently the most often used technique. But the exhausting process of manually traversing all slices is replaced and the diagnosis is expected to be faster.

1.3 Inspection of Tubular Structures

State-of-the-art techniques for inspecting tubular structures enable to see most of the relevant information. All prominent information has to be visible on the final image. Typical examples are virtual colon unfolding [80] for polyp detection, or curved planar reformation for vessel visualization [35]. The visualizations are using non-linear distortions of the original data such as straightening, reformation, or flattening. Therefore the spatial location of a particular feature (e.g., polyp or calcification) might be somewhat difficult to perceive. Although the visualizations are suitable for diagnosis, they are less usable for planning of potential surgical treatments.

The spatial arrangement can be conveyed to the medical doctor by linking the distorted view to views of the original volume where the spatial position of a particular feature is depicted more clearly. In this case expressive visualization is superior to other visualizations: Small, but important focus features are here the calcified vessel parts or suspicious polyps located on the colon walls respectively. Selecting a region in the distorted view, an inverse mapping enables to identify the corresponding region in the undistorted view. To give the medical doctor a clear view on the selected regions, the occluding features in front of the focus are automatically suppressed. Small features such as calcifications will be additionally enhanced by expressive effects such as halos or glows to immediately attract the expert's attention. Focusing the user's attention on the most relevant information can also be done by darkening or blurring of context information.

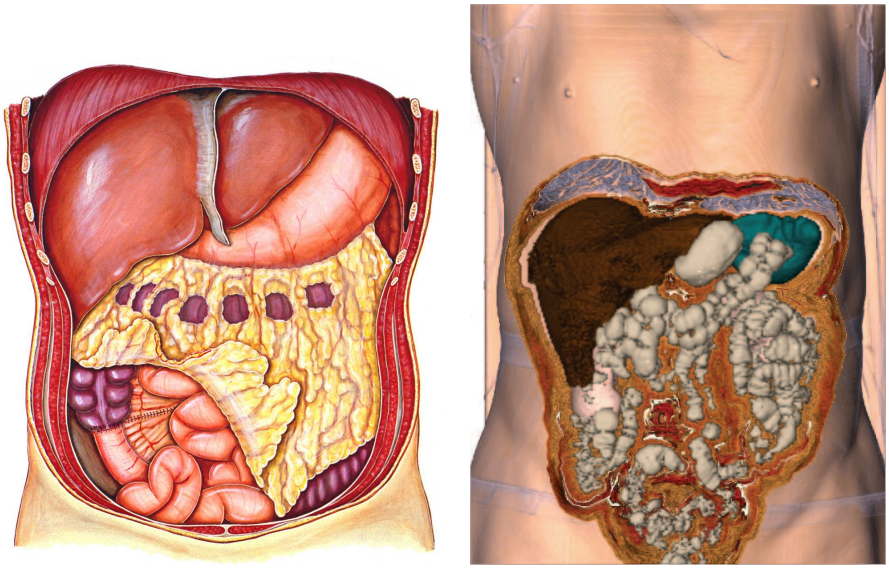
1.4 Thesis Contribution

The contribution of this thesis to scientific visualization is presenting importance-driven feature enhancement as an approach to automatic focus+context volume rendering. The proposed method overcomes the problem of occlusions within the volume, which typically happens when using a view-independent classification. The optical properties of the proposed technique may remain constant over an entire object, or they can vary when cut-away views or similar concepts are incorporated into the importance-driven model. Depending on the viewing situation, the estimated *level of sparseness*, i.e., the level of visual abstraction, is selected. In order to visually emphasize features with the highest importance, occluding objects between these features and the viewpoint are rendered sparsely. Interesting objects are represented more densely to see most of the details. If no occlusion occurs, even the less important features can be rendered densely. This enables an automatic generation of images with maximal visual information.

In Figure 1.1 an anatomical illustration of the human abdomen [30] and a result of importance-driven feature enhancement is illustrated. In this case the internal structures are classified with a high importance value so that structures between the viewpoint and the important features are simply cut away automatically.

1.5 Thesis Outline

The overall organization of the thesis is as follows: Chapter 2 gives an overview on artistic illustration techniques from the past to more recent approaches. This chapter is a prerequisite to the following chapters where the computer graphics and visualization techniques inspired by art and illustration are discussed. Chapter 3 follows with an overview on techniques where the visual representation is different to a photorealistic rendering. This includes different shading metaphors inspired by art, for surface and volumetric visual representations. Various visual representations are then presented with respect to particular visualization goals such as focus+context rendering. The chapter concludes with state-of-the-art techniques for expressive visualization and automatic scene enhancement. The thesis then explains the basic concepts of importance-driven visualization in Chapter 4. This chapter describes the key elements of importance-driven visualization in more detail. Several results are presented at the end of this chapter. Chapter 5 discusses possible



(*a*) (*b*)

Figure 1.1: *Comparison between (a) an artistic medical illustration of the abdomen [30] and (b) automatic view-dependent cut-away illustration.*

visibility validation schemes that can be integrated into the importance-driven model to guarantee a tight correspondence between the importance distribution and the visibility distribution. Chapter 6 presents applications of importance-driven visualization in various fields of visualization such as medical diagnosis, flow visualization and interactive illustrative visualization. The thesis draws conclusions in Chapter 7.

The presented work is based on the following publications: The background on illustrations and the state-of-the-art illustrative techniques in computer graphics and visualization (Chapters 2 and 3) are included in Eurographics 2005 tutorial on ***Illustrative Visualization*** [81]. The basis for Chapters 4 and 5 describing importance-driven visualization are publications on ***Importance-Driven Volume Rendering*** [82] and ***Importance-Driven Feature Enhancement in Volume Visualization*** [83].

Chapter 2

Background in Technical Illustration

An *Illustration* is a visualization that stresses subject more than form. Examples include drawings, paintings, or photographs. The aim of an Illustration is to elucidate or decorate a story, poem or piece of textual information (such as a newspaper article) by providing a visual representation of something described in the text. Illustrations can be used to display a wide range of subject matter such as:

» giving faces to characters in a story;
» displaying a number of examples of an item described in an academic textbook (e.g. typology);
» visualizing step-wise sets of instructions in a technical manual.

The term 'Illustration' can also be used in a non-visual sense i.e. "give an example to Illustrate the point you are trying to make".

Definition by *Wikipedia.org [87]*

2.1 History of Illustration

Illustration has always been an important visual communication medium among humans. The origin of illustration can be found in the Paleolithic period somewhere between 30,000 and 10,000 B.C. The cave paintings from this period display mostly large wild animals, such as bison, horses, aurochs, and deer, and tracings of human hands. Drawings of humans are rare and are usually schematic rather than the more naturalistic animal subjects. The paintings were drawn with red and yellow ochre, hematite, manganese oxide and charcoal. Sometimes the silhouette of

Figure 2.1: *The prehistoric paintings found in caves of Lascaux in France [46].*

the animal was incised in the rock first. In Figure 2.1 an example of prehistoric paintings is shown. It contains some of the earliest known art, dating back to somewhere between 13,000 and 15,000 B.C.

The painters of ancient Egypt (3200 B.C. to 30 B.C.) were among the first *professional* artists. Their visual language, hieroglyphs, depicted religious practices, political propaganda, scientific data, and daily life. The most important element in Egyptian paintings is the line. All paintings are bordered by black lines. It is generally accepted that Egyptians did not use perspective, they used hierarchic perspective in their early profile drawings. Through overlapping they tried to give the idea of depth. Some scenes with sets of people overlapping represent workers involved in the seeding of the fields. They convey the idea that there were several persons working next to each other. The base of these paintings is a tomb wall or a sarcophagus wood. Figure 2.2 shows hieroglyphs located on a tomb wall.

Like the hieroglyphs of the ancient Egyptians, the architectural drawings of the early Greeks (1100 B.C. to 100 B.C.) also lacked perspective. To imitate this kind of art in architecture, the ancient Greek architects even designed their buildings to visually counterbalance the viewer's intuitive

Figure 2.2: *Hieroglyphic paintings from ancient Egypt.*

understanding of perspective. A typical example of this is the Parthenon in Athens, which was situated at the top of the Acropolis compound. The Parthenon could only be approached from one access point. The rear of the structure is bigger and wider than the front, and the side columns increase in mass from front to rear. This construction technique gave the Parthenon an appearance through which it approximated the flat or axonometric views the Greeks were used to see in their art. Figure 2.3 shows example from typical ancient paintings painted on a vase.

The principle of perspective was defined in the year 1000 A.D. by the Arabian mathematician and philosopher Alhazen [60]. He explained in his work that light projects conically into the eye. A method for projecting a scene onto an image was developed approximately 300 years later during the Renaissance period. In this period the beginning of descriptive technical illustrations took place through the work of artists such as Leonardo da Vinci (1452-1519). In da Vinci's personality artistic abilities were combined with a scientific mind, which enabled a merging of visual art with invention. The creation of spatial illusions was another major

Figure 2.3: *Paintings from ancient Greece. The Greek visual art still lacked perspective in their works similar to ancient Egypt.*

achievement of this period. The evolution of what is called illusionistic perspective was taking place. This is illustrated in Figures 2.4 and 2.5 showing early drawings that are directly related to technical and medical illustrations.

The flourishing of technical illustrations was a direct implication of the industrial revolution. Mass production and outsourcing created the need to adopt conventions and standards in technical illustrations that were universally understood. By the mid 1900s both artistic and technical illustrators had a predictable methodology available for illustrating objects and environments more realistically. Illustrative techniques are often designed in a way that even a person with no technical understanding clearly understands the piece of art. The use of varying line widths to emphasize mass, proximity, and scale helped to make a simple line drawing more understandable to the lay person. Cross hatching, stippling, and other basic techniques gave greater depth and dimension to the subject matter. Technical illustration was further advanced during the photorealistic art direction around 1960. Photorealists were often

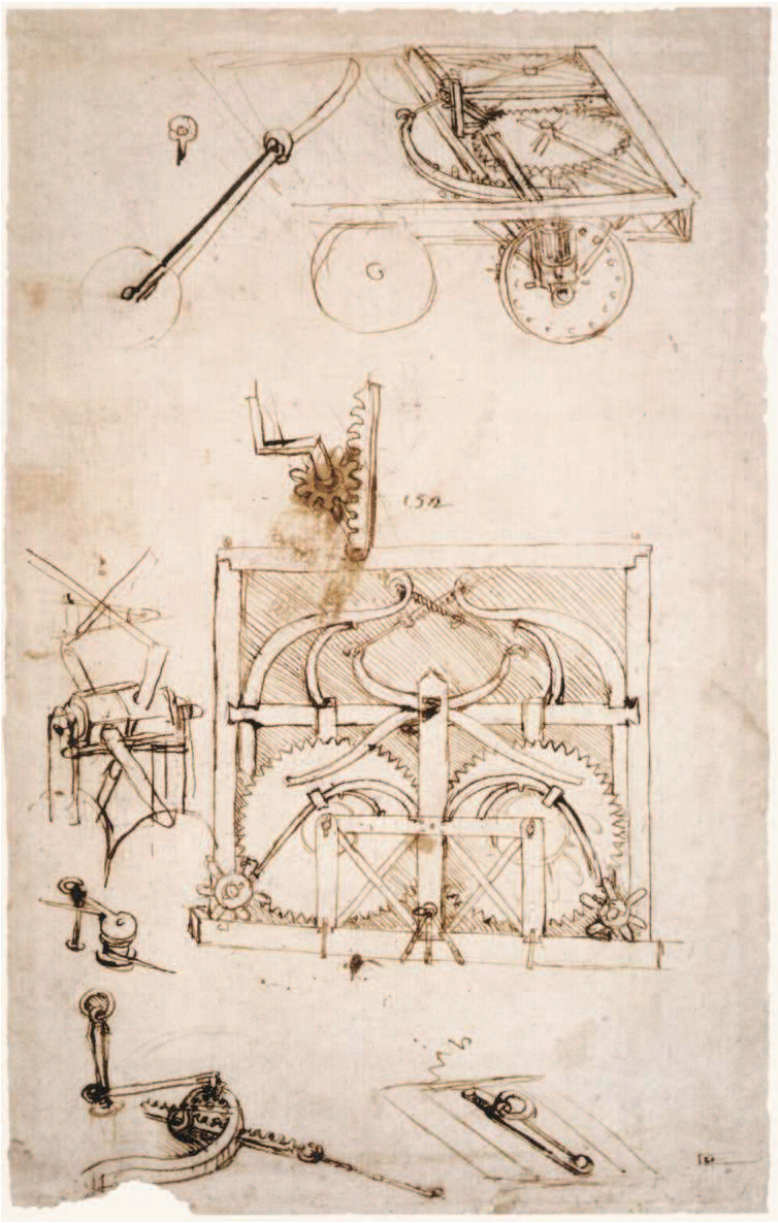


Figure 2.4: *Technical illustration of a vehicle created by Leonardo da Vinci [84].*

working with photographic slide projections onto canvases. The style is very accurate, with an emphasis on details and often simulates glossy effects such as reflections in specular surfaces. Typical paintings of this period also include so called pin-up paintings where beautiful women had been portrayed (Figure 2.6).

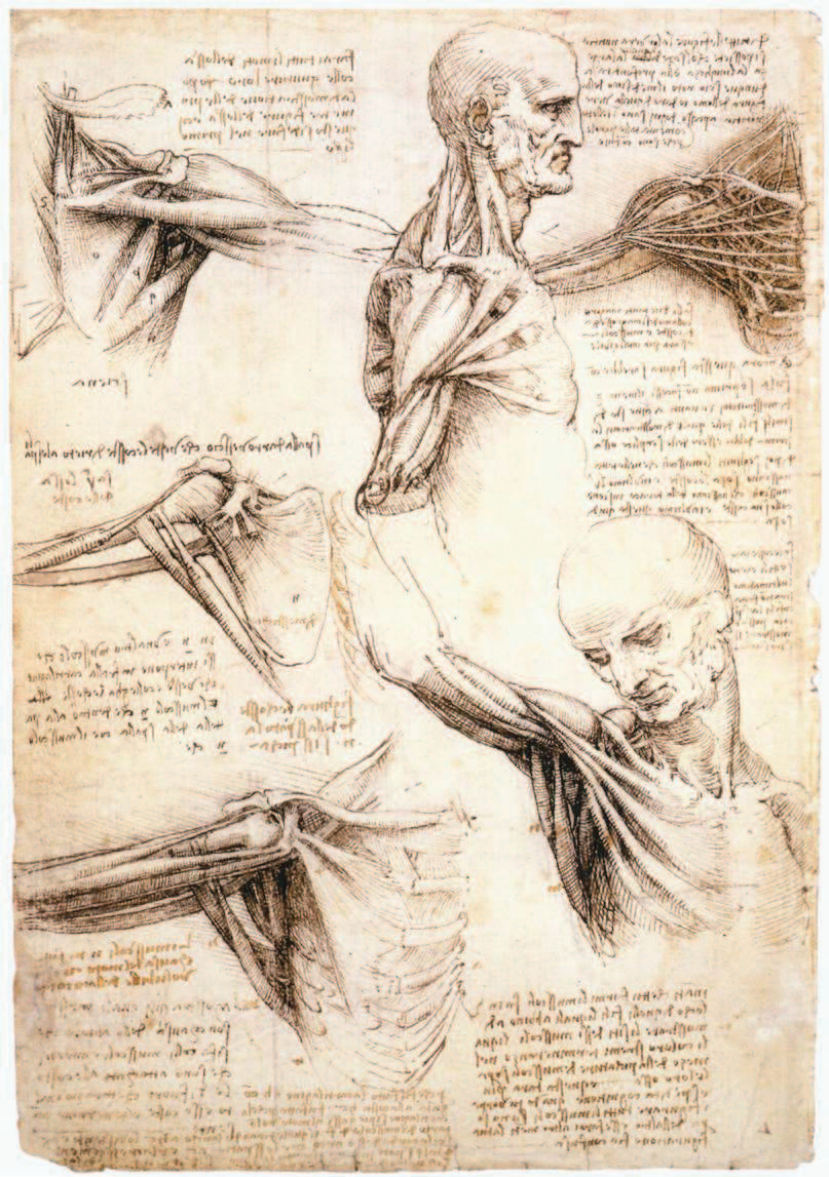


Figure 2.5: *Medical study of a human shoulder created by Leonardo da Vinci [84].*

By merging technical illustration and photorealism, the technical illustrator could now convey highly complex technical information to someone with little understanding of mechanics or drafting. To further increase the expressivity of the illustrations, various techniques have been established. In the following we will shortly review the state of the art techniques in modern illustration.

Figure 2.6: *A pin-up painting as an example of the photorealistic art direction.*

2.2 Modern Illustration Techniques

Technical, medical and biological illustrations have accommodated several types of rendering styles. The oldest technique is the line art technique (also often denoted as pen-and-ink technique), where the artist draws the shape contours and lets the human imagination complete the rest of the information. Big advantages of line art drawings are the high contrast so the features are clearly visible and the small covered image space. The disadvantage can be the sometimes bad spatial perception and the lack of shape information of feature parts facing directly towards the viewer. The spatial perception can be partially enhanced be encoding the spatial depth into the thickness of the lines. Shape information can be added to the line art using additional techniques like stippling, hatching, or charcoal shading. An example of the line art drawing is shown in Figure 2.7. The Figure shows the difference to photorealistic technical illustration in Figure 2.8. Although there is a clear understanding of the spatial arrangement among the features, the photorealistic drawing takes-up too much of the image space, which may cause a certain overload of the viewer. Therefore a combination of line art and photorealism turns out to be very useful, where some features are only outlined and other features are represented in a more realistic way. Also quite popular is a combination of a real photograph and line art.

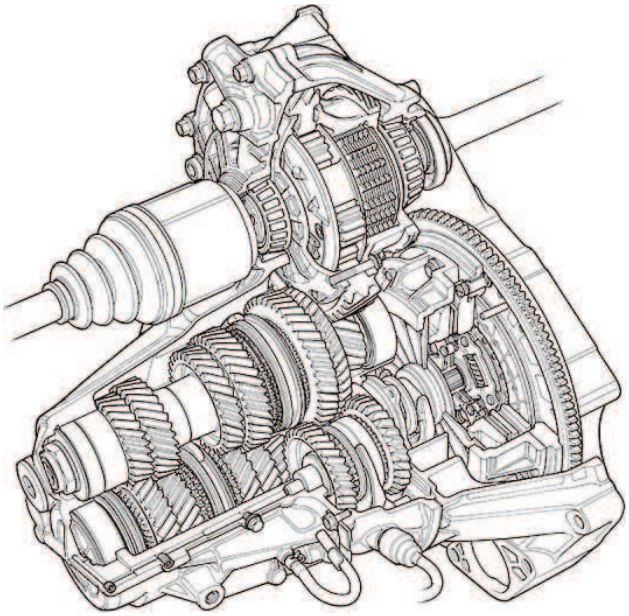


Figure 2.7: *Technical illustration using line-art drawing style [31].*

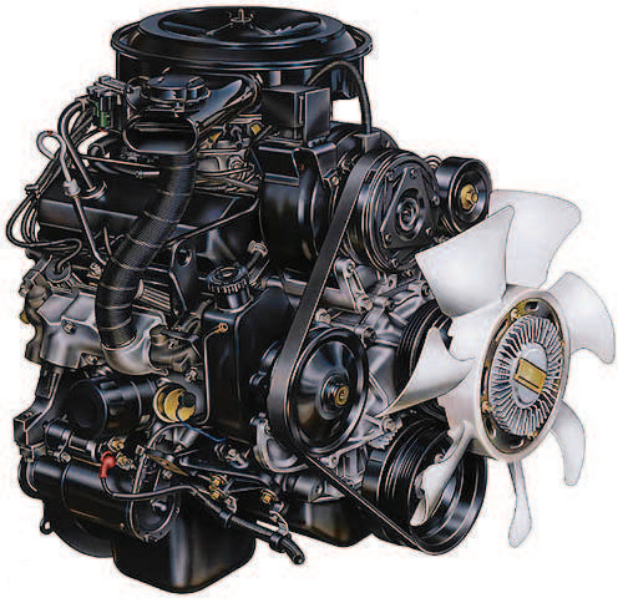
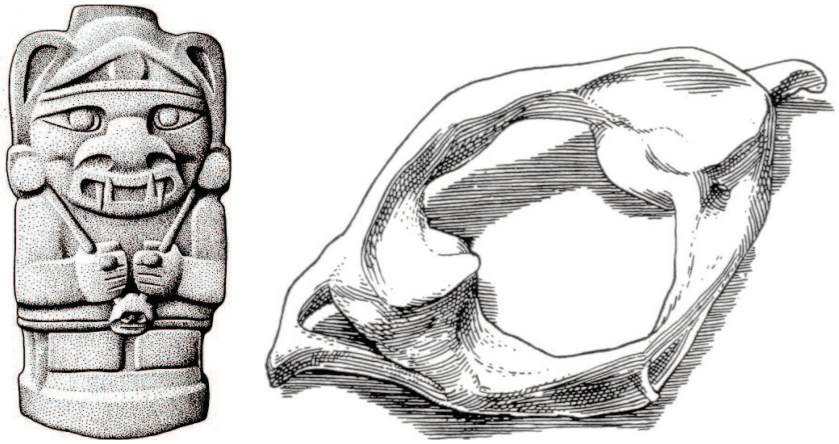


Figure 2.8: *Technical illustration using photorealistic drawing style [31].*

The above mentioned approaches like stippling, hatching, or charcoal shading allow to understand the front shape of features by simulating shading. Although shading simulation can be achieved by computer-



(*a*) (*b*)

Figure 2.9: *Hand-crafted line-art shading using (a) stippling and (b) hatching [29].*

generated halftoning, illustration artists often prefer to use the above mentioned types of shading due to their simplicity and handmade appearance. Such techniques are mostly used in biological illustrations, e.g., drawings of insects, animals, or vegetables.

Stippling is the technique of placing small dots to simulate shading. The dots are of a single color mostly black or brown, applied with a pen or brush. The denser the spacing of the dots, the darker is the apparent shade which is analogous to halftoning. Another line art technique for shading by hand is hatching, which uses lines instead of dots. The local density of the line strokes simulates the shading. Additionally the line direction on a surface is often aligned to principal curvatures to enhance shape understanding. A slightly modified version of hatching is called crosshatching, where the shade intensity is simulated by two line directions perpendicular to each other. Figure 2.9 shows examples of line art extended by shading through stippling and hatching.

Apart from different drawing techniques based on the type of drawing media, illustrators additionally use refined techniques to unveil most of the present information. Especially educative medical and technical illustrations try to show most the of information by incorporating artifical cuts and by removing some of the less relevant information. Often the problem is that the most interesting features are not easily perceivable, because they are occluded by less important features.

The most simple technique to resolve the visibility problem in the final illustration is by incorporating some clipping geometry, e.g., clipping

planes. A clipping plane defines two half-spaces. Information that is spatially located in one half-space is visible, while information in the other half-space is not displayed. This is easy and intuitive way to unveil the most important data (in the text further denoted as focus). However such an approach eliminates less important objects (i.e., contextual information) also in those viewing situations, where it would not be necessary. In the worst case the arrangement leads to a reduced spatial perception, because too much of the contextual information has to be removed. Applying different drawing styles and rendering techniques (e.g., contour rendering) in the suppressed half-space ease the problem to a certain degree.

Expressive illustration techniques such as section views, cut-away views, ghosted views, or exploded views and zooming distortions effectively uncover most important information by changing the level of visual abstraction or modifying the spatial arrangement of features. Generalized cut-away views are dealing with occlusion by removing certain parts of illustrated structures. These parts are either less important, or they are present in the final image to a sufficient degree so that the viewer can understand how the removed part would look like. A simple example are the section views that are similar to clipping planes. Instead of one half-space that has to be removed, a section is defined as the intersection of several half-spaces, e.g., aligned along a symmetry axis of the depicted object. An illustration using section views is shown in Figure 2.10 (a). The clipping geometry can be of course defined arbitrarily and for each structure individually as shown in Figure 2.10 (b). For contextual structures that are more complex in shape, it is often useful to convey shape in the removed areas by contour lines. Also the level of transparency can be adapted so the feature is not removed completely, only its representation is suppressed. The original representation is replaced by a see-through one, therefore such a kind of illustration is denoted as ghosting. Figure 2.10 (c) shows an example of ghosting to unveil the inner parts, and still convey the outer shape of the camera.

Exploded views and deformations handle the problem of occlusion in a different way as compared to cut-away views. Deformations and exploded views do not change the visual representation of features. Instead they transform features or change the spatial arrangement to increase their visibility. Deformations usually deal with magnification of particularly interesting parts in order to have a detailed view on them. Other parts (contextual information) of the illustrated structure or device may be then partly occluded by the magnification.

Another expressive illustration technique are exploded views. A typ-

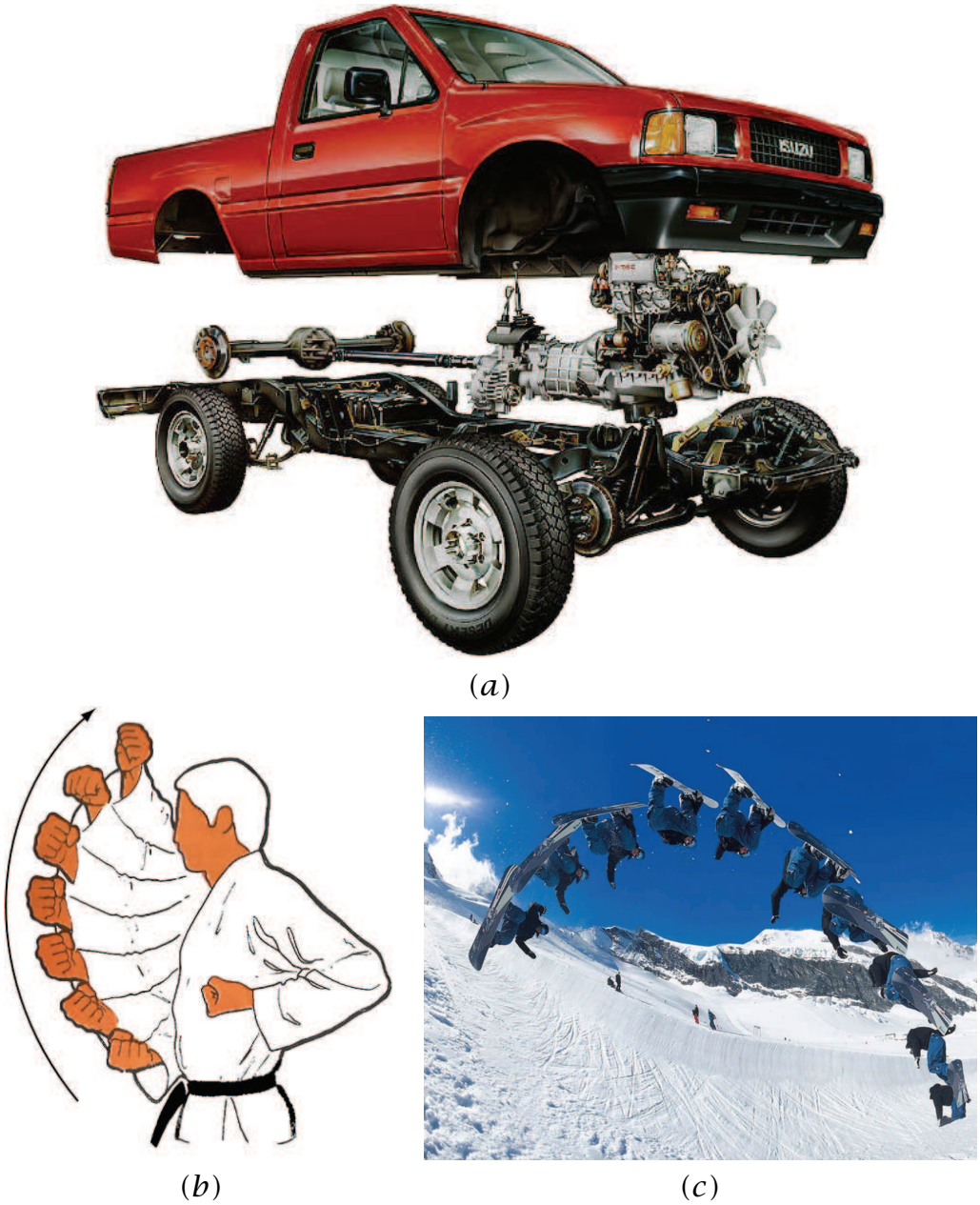
ical application area of exploded views are assembly instructions. They unobstructively show how individual parts parts have to be combined together to construct the final device. Assembly instructions use exploded views, where all components are visible because they are moved apart from each other. Symmetry properties of the object may suggest certain preferable explosion directions. The viewer processes the information by mentally putting all components together again. Another idea of exploded views is explosion in time. This means that a whole time series is being presented on one image. The idea of bringing time series is in photography realized through multiple exposures of moving scene. In image processing an automatic technique for combining multiple exposures into one image is called *computer enhanced multiple exposure numerical technique (CEMENT)* [56]. The technique is often used to illustrate crucial phases during the motion of an otherwise fast moving object. Figure 2.10 shows two different kinds of exploded views: (a) Spatial exploded view of a vehicle and a temporal exploded views in (b) illustration and (c) photograph.

Further examples of expressive illustration techniques can be found on the web-page of Kevin Hulsey Inc. [31].

(a) (b)

(c)

Figure 2.10: *Illustrations based on different cut-away principles: (a) section view of an oil filter, (b) cut-away view with arbitrary and individual clipping geometries for various structures, and (c) ghosting illustration of a digital camera [31].*



(*a*)

(*b*)

(*c*)

Figure 2.11: *Exploded views as illustrative technique: (a) a spatial exploded view [31], (b) a temporal exploded view illustration [69] and (c) a C.E.M.E.N.T photograph.*

Chapter 3

State of the Art

Scientific work related to expressive visualization and automatic feature enhancement can be divided into several categories. To emphasize a particular subset of the data, first data classification methods have to be defined. This chapter starts with an overview of visualization techniques dealing with feature definition. A feature can be specified according to a function value, spatial location, local properties, or, in the multi-variate case as a product of function value intervals for each modality. These approaches will be discussed in Section 3.1.

In Section 3.2 an overview of existing methods for visual representations of data is given. Many of these methods are inspired by artistic approaches that exploit the human visual system and provide certain level of abstraction to convey the most relevant information. These techniques are divided into surface-based methods and volumetric methods.

Section 3.1 on feature classification and Section 3.2 on visual representations provide the basic knowledge for understanding Sections 3.3 on focus+context visualization and Section 3.4 on smart visibility in visualization. Visualizations based on the focus and context (F+C) principle are discussed and advanced visualization techniques derived from illustration techniques presented in Section 2.2 are shown. Finally, early work on automatic visual enhancement based on human perception is discussed in Section 3.5.

3.1 Feature Classification

Segmentation

Segmentation is the the most typical and most often used feature classification approach in digital image processing and visualization. Segmentation refers to a (binary) partitioning of the data (raster image or

volumetric data) into multiple sets. Partitioning is done according to some predefined criterion. Thresholding is a most simple segmentation method that partitions the data according to a given threshold value or threshold interval. Such a segmentation is not very robust as soon as different features in the data include sets of pixels with the same or similar function value.

Therefore more sophisticated methods have been developed over the years. In the following we shortly discuss those techniques that gained popularity in the visualization area. Further methods can be found in the digital image processing literature [71, 73].

A typical segmentation algorithm is the region growing approach [71]. It is based on iterative merging set of elements such as pixels or voxels according to similarity constraints (e.g., homogeneity in gray levels or color). Initially an arbitrary seed point is selected and compared with neighboring elements. The region is grown from the seed point by adding neighboring elements that fulfill the homogeneity criterion, and the region size increases. This process is iteratively repeated until no neighboring elements are added according to given homogeneity criterion.

A similar concept to region growing is the concept of *watersheds*, adopted from topography [71]. The principle is explained on gray scale images. The gray scale is considered as a height field with gray levels encoding the height. Around each local minimum of such an image a *catchment basin* is defined, such that each of its points can be connected with the minimum by a descending path, denoted as downstream. Borders, between different catchment basins are called watersheds.

Another semi-automatic segmentation approach is based on an active contour model known as *snakes* [36]. The user defines an approximate region of interest where the feature is located. The snake algorithm starts with the initial approximate contour defined by the user which is then fitted to the segmented region. The snake is controlled by internal and external forces, which are usually defined in energy terms. The internal forces keep the snake smooth, while the external forces attract it to features, such as object boundaries. The initial approximation is given as a set of points defining a spline for example. The concept of an active contour model is also used in level set segmentation [61]. The difference between snakes and level set methods is in the contour representation. While snake contour is given explicitly for example as a spline, the level set is defined implicitly.

Incorporating local properties of a sample point, i.e., its first-order or second-order derivatives, enables to segment features according to their local shape. The shape characteristics in the close neighborhood

of the sample is defined by the eigenvalues computed from the Hessian matrix that consists of all second order derivatives. An example is the segmentation of tubular structures (e.g., vascular structures) or blobby structures (e.g., polyps in the colon or early stage lung cancer nodules). Segmentation based on local shape properties has been proposed by Sato et al. [68].

Transfer Functions

A very useful feature classification technique, particularly in scientific visualization, is transfer function specification. A transfer function refers to a function that maps the data values directly to optical properties. The classical transfer function is a one-dimensional function dealing with scalar data. The transfer function in this case defines for each data value R,G,B color channels and an opacity α value. During image synthesis using for example the direct volume rendering approach (DVR) [48] the sample colors and opacities are composited using the *over* operator. A more detailed discussion on DVR rendering techniques is given in Section 3.2.

One-dimensional transfer functions can be extended to more dimensions also for scalar input data. Other dimensions for example describe further local properties at a sample point. The gradient magnitude allows to assign high opacity values in interface areas, emphasizing boundary regions between different data regions [37]. Kniss et al. present a framework for interactive volumetric data exploration and transfer function design [39]. Their interaction tools allow to specify the transfer function according to the function value as in the case of one-dimensional transfer function and as well as gradient and curvature magnitude.

A class of transfer functions has been proposed that exploit the curvature information given by second-order derivatives [28, 38]. The optical properties vary according to the local shape characteristics. Additionally objects with the same general shape can be distinguished by different curvature magnitudes. The work on curvature-based transfer functions has been also applied to non-photorealistic contour rendering and is discussed in Section 3.2.

Multi-Modal Feature Enhancement

Previous feature definitions are primarily concerned with scalar data, although segmentation techniques and transfer function specification are

not limited to scalar data. These techniques can be also used to extract features from multi-modal datasets. However, the nature of multi-modal data, allows to define a feature more precisely. Typically a combination of value intervals from different modalities defines a feature. With a growing number of data properties the design of interaction tools becomes more complex.

Doleisch et al. propose a framework of feature definition for multi-modal simulation data [15]. They use multiple linked scatterplots and histograms for feature specification. Smooth brushing is supported that classifies the data analogous to fuzzy sets. The set can be hierarchically described by brushing multiple dimensions. The feature definition is shown in the rendering view for an iterative adjustment of the feature set.

An interesting approach was presented by Hauser and Mlejnek for multi-dimensional 3D flow data using the above mentioned framework [25]. They denote the result obtained from the feature definition as the *degree-of-interest* (DOI) function. The DOI function is mapped to optical properties so for example the region with non-zero DOI is outlined by contours and the most relevant information (maximal DOI values) is shown as an iso-surface.

Focal-Point Feature Definition

An interesting idea is to classify a feature according to the distance to a focal point. The classification allows to smoothly or discretely vary the visual representation from a most *dense* to a more abstract one [90]. The optical properties are changing according to the distance to the focal point. This concept has also strong relevance to focus+context visualization as discussed in Section 3.3.

Another approach using a focal point is gaze-directed volume rendering [49]. This was an early approach in visualization, where the observer's viewing direction was taken into consideration. The pupils of the user were tracked to determine the focal point. The motivation in this case was to increase the rendering performance instead of enhancing the visual information. The volume dataset is rendered in different resolutions. According to the viewing direction only the focal region is represented in full resolution, and the other parts are rendered in lower resolution.

Classification via Painting

In this part special feature classification methods are discussed that are based on direct interaction with the data via painting. The first technique is based on painting directly on the three-dimensional visual representation of the underlying data [4, 5]. The painting classification evaluates the intersection of the *visible* parts of the data with the brush ray. The brush ray is represented as a three-dimensional Gaussian kernel cast from the viewport in the viewing direction. Such a segmentation works well for volumetric scalar data displayed with direct volume rendering (see Section 3.2) and using transfer functions for the visual mapping. The brush ray is cast through the volume data until it intersects a not fully transparent region. This is often denoted as the *first hit* with the visible data. According to the Gaussian kernel representation of the brush the segmentation is not a binary classification, but relates more to *fuzzy sets*. By iteratively selecting and deselecting (i.e., brushing and erasing) from different viewing directions an intuitive feature selection is achieved.

A second classification approach based on painting is related to an automatic transfer function definition. Through painting the user roughly identifies interesting structures on two-dimensional slices. A neural network generates a multi-dimensional transfer function that reflects and completes the user-supplied information. Such a definition is also an iterative approach during which the network *learns* the desired visual representation. This way of specifying of visual representation has been proposed recently by Tzeng et al. [77].

3.2 Visual Representations

The central role of visualization is to provide the user with a visual representation of underlying non-visual data. The goal is to convey properties of the data in an efficient and effective way. The graphics community has been inspired by visual artists to deviate from a pure photorealistic representation to use more abstract representations which exploit the human imagination. In the following techniques directly derived from art are discussed that have been adapted for computer graphics and visualization goals. First techniques that represent data as surfaces derived from the underlying polygonal or volumetric representation are depicted. Here techniques are presented that simulate contour painting and artistic shading techniques ranging from pen-and-ink shading to cartoon shading styles. The next category of rendering techniques is

intended for volumetric data, where the structures are not given as a set of surfaces, but as a set of samples distributed over a three-dimensional domain. In this area techniques from X-ray simulation to more elaborate representations using direct volume rendering (DVR) are discussed.

Surface-Based Representations

The display of *contours* is a popular method to visually represent digital data by a high-level of abstraction. Only parts of data with high gradient values and gradients approximately perpendicular to the viewing direction are shown. Contours give only a rough overview of the shape, but they take up only a small portion of image space. Contour rendering is very often used in focus+context visualization as we will discuss later in Section 3.3.

Another advantage of contours is that they can be computed very easily. To estimate whether a particular surface sample i is a part of a contour or not, it is only necessary to compute the angle between the viewing vector V and the surface normal N. The evaluation can be done for example by the following Equation 3.1:

$$C(i) = (1 - |N(i) \cdot V|)^n \tag{3.1}$$

The function C results in values in the range $[0, 1]$. The closer a value $C(i)$ is to 1 the more it belongs to the contour. The exponential factor n controls the thickness and sharpness of the contour.

A contour representation can be applied to polygonal data and to volumetric data as well. In case of volumetric data there are two approaches that should be distinguished. The first approach is showing the contour of a particular iso-surface, i.e., a surface within the data with a fixed scalar density value. In this case this is basically a surface based technique. Another contour rendering technique shows all contours within the volumetric data and therefore they are considered as volumetric contours. These will be discussed in the next part dealing with volumetric methods for visual representations.

Rendering contours using the previous method can lead to contours of varying thickness. Flat surface regions produce thick contour areas in image space, whereas high-curvature surface regions generate thin contour areas. To avoid this artifact, contours from polygonal data can be determined alternatively as edges between faces with different visibilities. A change in visibility is given by a sign change in $N(i) \cdot V$ [20]. Rendering contours from iso-surfaces of volumetric data requires to modify

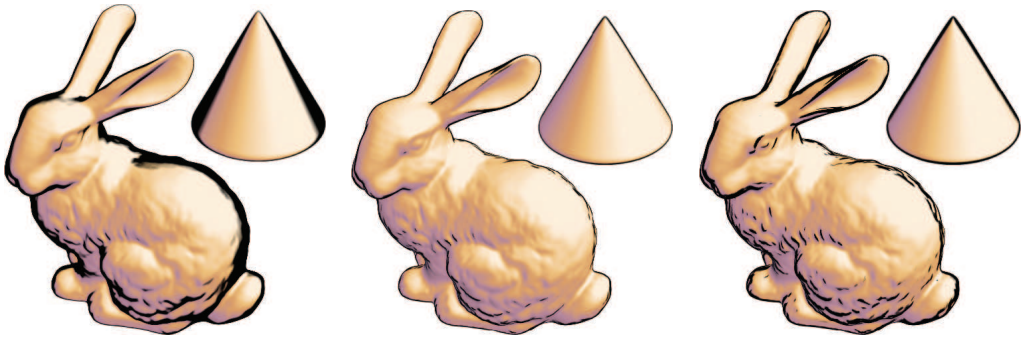


Figure 3.1: *Contour rendering (a) with a varying contour thickness and with (b) a thin or (c) a thick constant contour thickness using the curvature radius [38].*

Equation 3.1 to control contour thickness. This can be achieved by incorporating second-order derivative information, namely the curvature radius [38].

Figure 3.1 (a) shows contours of varying thickness due to a simple contour evaluation according to Equation 3.1. In Figure 3.1 (b) and (c) the curvature radius is taken into account to produce contours with constant thickness.

Suggestive contours [12] are an extension of traditional contours to convey the shape information of polygonal data more effectively. Suggestive contours combine contours rendered from a particular viewpoint with virtual contours from additional viewpoints close to the current view. This technique has been recently extended to achieve real-time performance with temporal coherence [11]. Figure 3.2 illustrates the difference between traditional contours evaluated from the given viewpoint only and suggestive contours where nearby viewpoint locations are also contributing to the final representation.

Similar as in the art domain, computer-generated contours can be extended by various shading techniques to better convey the shape information. First a brief review on methods simulating *pen-and-ink* shading techniques is given. Then other shading techniques based on modifications of the Phong illumination model [65] are discussed.

The pen-and-ink drawing style uses a distribution of points or lines as a replacement for photorealistic shading. An early attempt to algorithmically simulate pen-and-ink drawings has been presented by Winkenbach and Salesin [88]. They introduce stroke textures, which include the

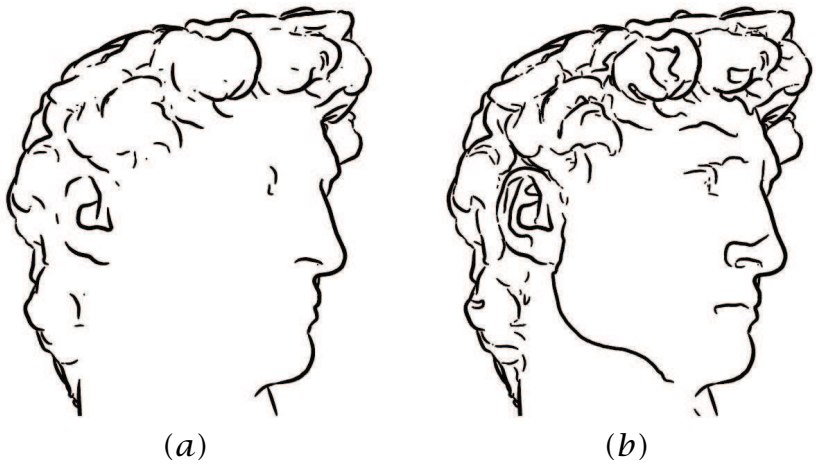
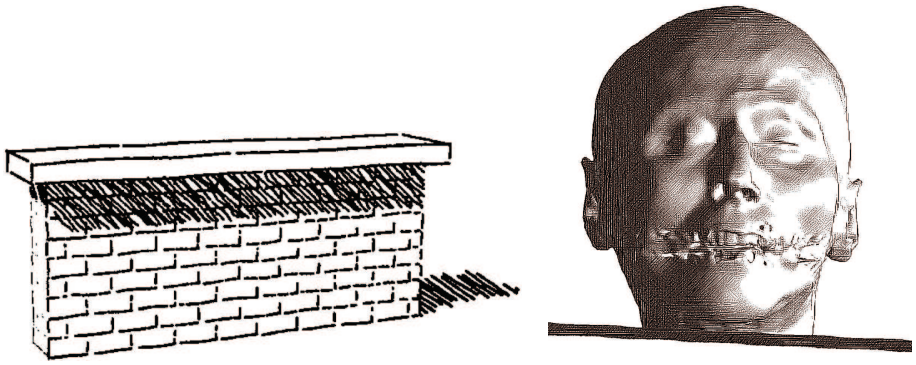


Figure 3.2: *(a) Traditional contour rendering evaluating the angle between the normal and the viewing vector and (b) a more expressive suggestive contours rendering [11].*

pen-and-ink texture of a particular surface, e.g., brick wall, and shading textures using hatching, cross-hatching or stippling [63] (see Section 2.2). Stroke textures also allow resolution-dependent rendering, to achieve an approximately constant thickness and density of strokes despite varying distances to the viewport. Stroke textures are illustrated in Figure 3.3 (a). Figure 3.3 (b) shows pen-and-ink rendering applied to the *skin* iso-surface of a volumetric dataset. Pen-and-ink rendering in volume visualization from the focus+context perspective is discussed in Section 3.3.

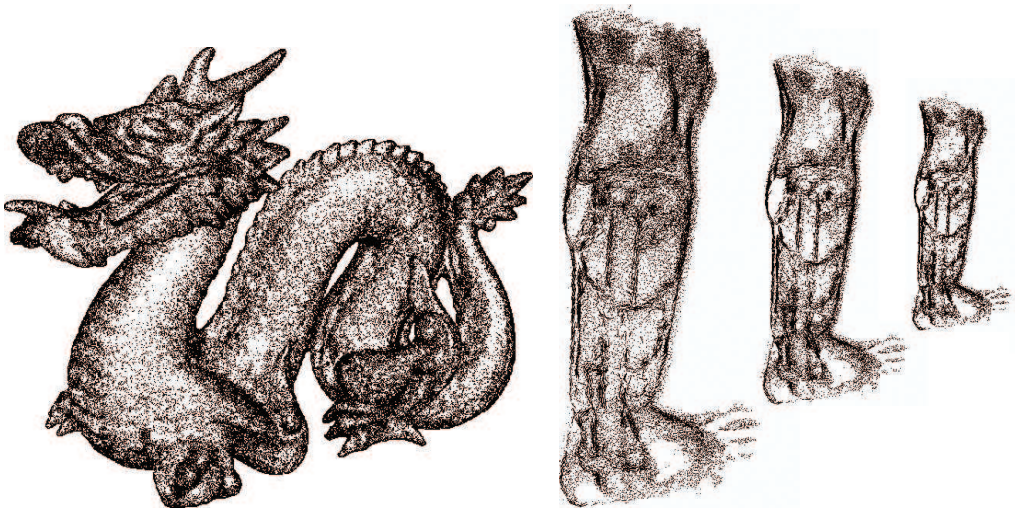
Stippling rendering of scientific data has been recently presented by Lu et al. [52]. They present an interactive technique for shading through stippling of surfaces and volumetric data. Their technique also includes resolution compensation that allows to see the stippled object with approximately a constant intensity of dots at varying distances. Figure 3.4 (a) shows stippling applied to a polygonal representation. Figure 3.4 (b) shows stippling applied to volume data of a leg. This example also illustrates the resolution enhancement which results in a nearly constant brightness with varying scales.

Apart from shading techniques realized through distributions of dots or lines, other shading models have been developed to simulate a particular illustration style. The most popular method that delivers a photo-realistic appearance is Phong shading. In this shading model a smooth intensity transition from regions facing towards the light source to re-



(*a*) (*b*)

Figure 3.3: *Examples of pen-and-ink rendering for (a) polygonal [88] and (b) volumetric data [76].*



(*a*) (*b*)

Figure 3.4: *(a) Surface-based stippling applied to polygonal data of a dragon and (b) volumetric stippling of a leg with resolution enhancement [52].*

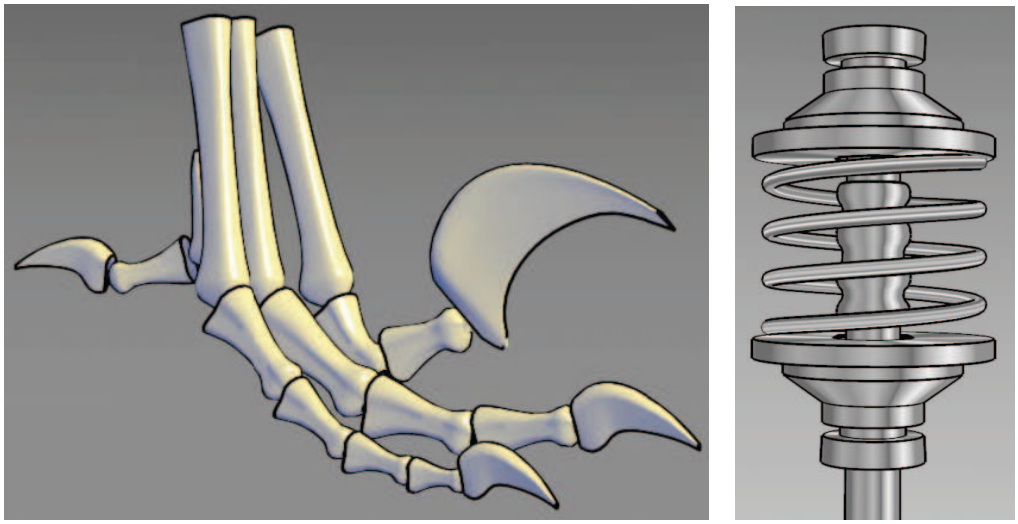
gions facing away from the light source is calculated. In the following slight modifications of the Phong illumination model are described which achieve shading effects similar to various different shading styles.

A popular illustration technique for animations is cartoon shading. In contrast to smooth intensity transitions achieved by Phong shading, car-

Figure 3.5: *Non-photorealistic cartoon shading [45].*

toon rendering uses only two or a few color intensities per material. The high intensity color is defined for regions facing almost directly towards the light source. The low intensity color with the same hue is defined for all other object regions. Cartoon shading can be achieved by replacing the illumination evaluation by a simple two-element lookup table with pre-calculated high and low intensity color values [45]. The input argument is the positive dot product between the normalized light source L and normal vector N ($max(0, L \cdot N)$). Color quantization is another possibility to achieve cartoon shading directly from the Phong illumination computation. The result of cartoon rendering is shown in Figure 3.5.

The Phong illumination model may result into very high dynamic changes of the highlights. This can reduce shape perception because of large areas that are either too dark or too bright. Another effective way of representing shading can be a shift in the hue channel instead of a shift in the intensity channel. Gooch et al. [19] propose a non-photorealistic lighting model for computer-assisted technical illustrations denoted as *cool-to-warm shading*. Instead of realizing shading by a transition from high to low luminance, a color transition from yellow to blue with approximately constant luminance is used. This encoding of shading is



(*a*) (*b*)

Figure 3.6: *Non-photorealistic shading techniques: (a) cool-to-warm color hue shading and (b) metal shading [19].*

mixed with the material color to estimate the final color value. Shading in the hue channel may convey the shape better, as compared to intensity shading. Figure 3.6 (a) is showing the hue shift based on cool-to-warm shading.

Metal shading is a modification to Phong shading which simulates anisotropic reflection of metallic objects. The technique applies a similar approach as illustrators do in traditional technical illustrations. In order to simulate the anisotropic reflection a set of lines with varying and rather low intensity is streaked along the parametric axis of maximal curvature. The intensity values between the lines are interpolated. Specular highlights in areas where the surface faces towards the light source are simulated by high intensity or white lines. A result can be seen in Figure 3.6 (b).

Volumetric Representations

This section focuses on techniques that visually represent volumetric data without extracting a particular part like an iso-surface. All surface-based visual representations mentioned above are applicable as volumetric visual representations without significant changes in the approach.

This is illustrated in the work on shading using stippling [52] where Lu et al. present stippling with respect surface-based representations as well as volume-based representations.

Scalar volumetric data can for example describe the density distribution over a particular volume. In order to visually interpret such data, we assign optical properties to the scalar values (e.g., densities). We denote the function that maps the original data values to a visual representation as *transfer function*. Advanced transfer function approaches have been already discussed in Section 3.1.

The display of volumetric data can be achieved by several volume rendering techniques [48, 44, 75, 86]. The volume rendering techniques are classified as image-order techniques, object-order techniques, and techniques based on the Fourier projection slice theorem. The most widely-used technique is direct volume rendering (DVR) [48], which is an image-order ray-casting technique. In DVR for each pixel in the image plane (viewport) a ray is cast through the volume data. At constant distances along the ray (denoted as *sampling distance*) a set of resampled values is computed from the discrete volumetric function. This set of samples is a discretized representation of the continuous ray. Instead of computing an analytical ray integral [57], the samples are combined via *compositing*. DVR typically includes the absorption and emission model [57]. DVR compositing is defined by Equation 3.2:

$$I[x,y] = \sum_{i=0}^{n} C(s_i)\alpha(s_i) \cdot \prod_{j=0}^{i-1} (1 - \alpha(s_j)) \tag{3.2}$$

Here, I is the color intensity at pixel $[x,y]$. s_i are the samples at position i along the ray which runs through pixel $[x,y]$. $C(s_i)$ are the local color values derived from the illumination model. $C()$ and $\alpha()$ are the color and opacity transfer functions. These functions assign a color and an opacity to each intensity value in the volume. Compositing can be also rewritten in a recursive form, which is more often used in practice. Its front-to-back variant is defined by Equation 3.3:

$$c_i = C(s_i)\alpha(s_i)(1 - \alpha_{i-1}) + c_{i-1} \tag{3.3}$$

$$\alpha_i = \alpha(s_i)(1 - \alpha_{i-1}) + \alpha_{i-1}$$

Here c_i and α_i are aggregated color and opacity values from the start of the ray until the sample i. Results of rendering various volumetric datasets using DVR are shown in Figure 3.7.

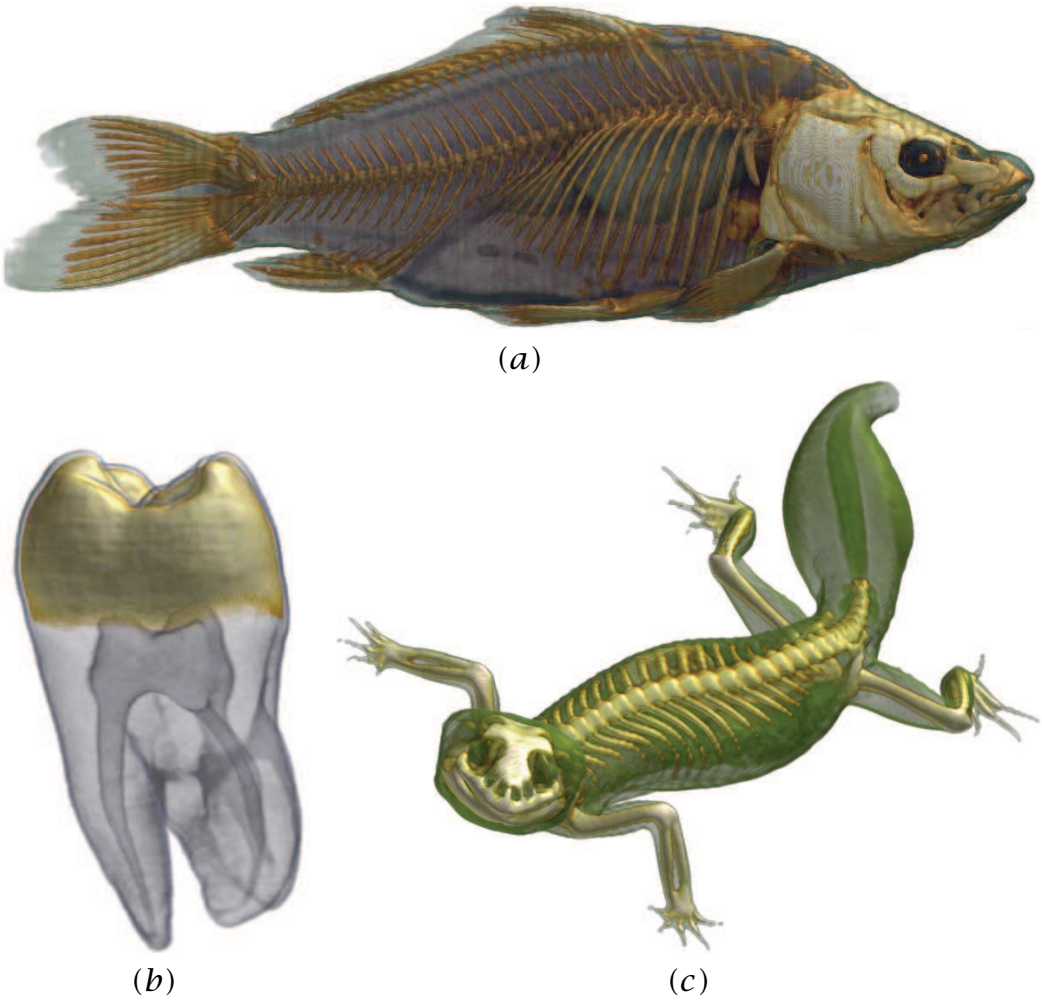


Figure 3.7: *Direct volume rendering of (a) carp, (b) tooth, and (c) Leopard gecko datasets using one-dimensional transfer functions.*

Other volumetric visual representations that are discussed below are also based on the ray-casting principle. The differences between the discussed volumetric visual representations is mostly given by a slight change in the compositing function.

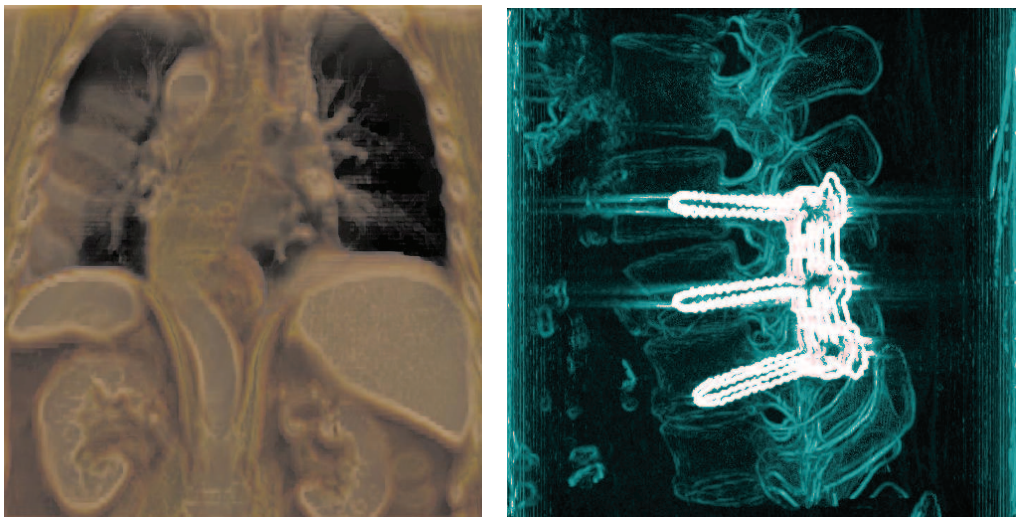
In the context of surface-based visual representations we have discussed contour representations as very useful because they roughly convey the shape and take-up just a small part of image space. Contour rendering is even more efficient for the visualization of complex volumetric data, because it enables to see outlines of volumetric regions with high

gradient magnitude. These outlines correspond to boundaries between volumetric features as the gradient magnitude is in these regions typically high. The compositing with contour enhancement is similar to DVR with an additional modulation of opacity according to local properties, i.e., gradient information [66]. This is expressed by Equation 3.4:

$$\alpha_i' = \alpha_i \left(k_{gc} + k_{gs}(||\nabla_{fn}||)^{k_{ge}} \right) \tag{3.4}$$

Here α_i' is the new opacity value. ∇_{fn} is the gradient vector at the sample location. The coefficient k_{gc} controls the influence of the original sample opacity α_i, k_{gs} controls the influence of local properties like gradient magnitude, and k_{ge} controls the thickness of the contour lines. Figure 3.8 shows two examples of volumetric contouring. The first image shows contour enhancement [66] in the rendering of a human torso and the second image shows non-photorealistic contour rendering of screws in the spinal cord [10]. The latter approach is using a similar concept as in Equation 3.4.Instead of the original density values only the gradient direction and magnitude are considered in the opacity modulation.

A typical visual representation of volumetric medical data is *summation*, which is analogous to X-ray imaging. Summation corresponds to a simple density projection, realized during the ray traversal by using *ad-*



(*a*) (*b*)

Figure 3.8: *(a) Contour enhancement of human torso [66], and (b) volumetric contour rendering of screws within a spinal cord [10].*

dition as the compositing function. Projections are visually represented with gray-scale intensities. Due to the analogy to X-ray imaging, medical users can easily understand such representations. Another advantage is that the ray is traversing the whole volume and every sample contributes to the resulting image. No information in the data is suppressed. Additionally, summation provides parameter-free rendering and an adjustment of visual settings like mapping the data to optical properties is not required. On the other hand, summation images do not effectively convey the shape of structures and their spatial arrangement is often unclear. An example of summation is depicted in Figure 3.9 (a).

A compositing function similar to summation is *maximum intensity projection* (MIP). Here the *maximum* operator is used instead of the *addition* operator. The pixel intensity is simply determined by the maximal density value along the corresponding ray. MIP also provides parameter-free rendering as in the case of summation. MIP also results in similar images as compared to summation, but MIP projections feature better contrast in general. MIP is often used in the visualization of computed tomography (CT) data of contrast-enhanced vascular structures. However this compositing function also does not provide good spatial perception, because the high density features appear to *stand-out* from their original spatial location. This artifact can be eased up by adding one parameter as threshold. This method has been introduced as *closest vessel projection* [91] and performs in the same way as MIP until a sample with an intensity higher than the adjustable threshold is reached. This sample value is the local maximum along the ray until a density value above the predefined threshold is encountered. The local maximum determines the pixel value irrespective from the fact that higher intensity values may appear *after* that sample. This method is also known as *local* MIP (LMIP) [67]. Visualizations using MIP and LMIP are shown in Figure 3.9 (b) and (c).

Another technique that provides almost parameter-free rendering is *bubble model* volume rendering [9]. In this case the gradient magnitude is considered instead of the density value at a particular sample position. The magnitude is proportionally mapped to an opacity value. The basic idea is that it highlights areas with high gradient magnitude, because these areas are usually boundaries between homogeneous tissues. An example showing the *bubble model* rendering is shown in Figure 3.9 (d). There is just one global scale parameter which determines the slope of the opacity ramp and thus the overall brightness in the result image.

Direct volume rendering as proposed by Levoy [48] is using the Phong illumination model [65]. This illumination model does not take into ac-

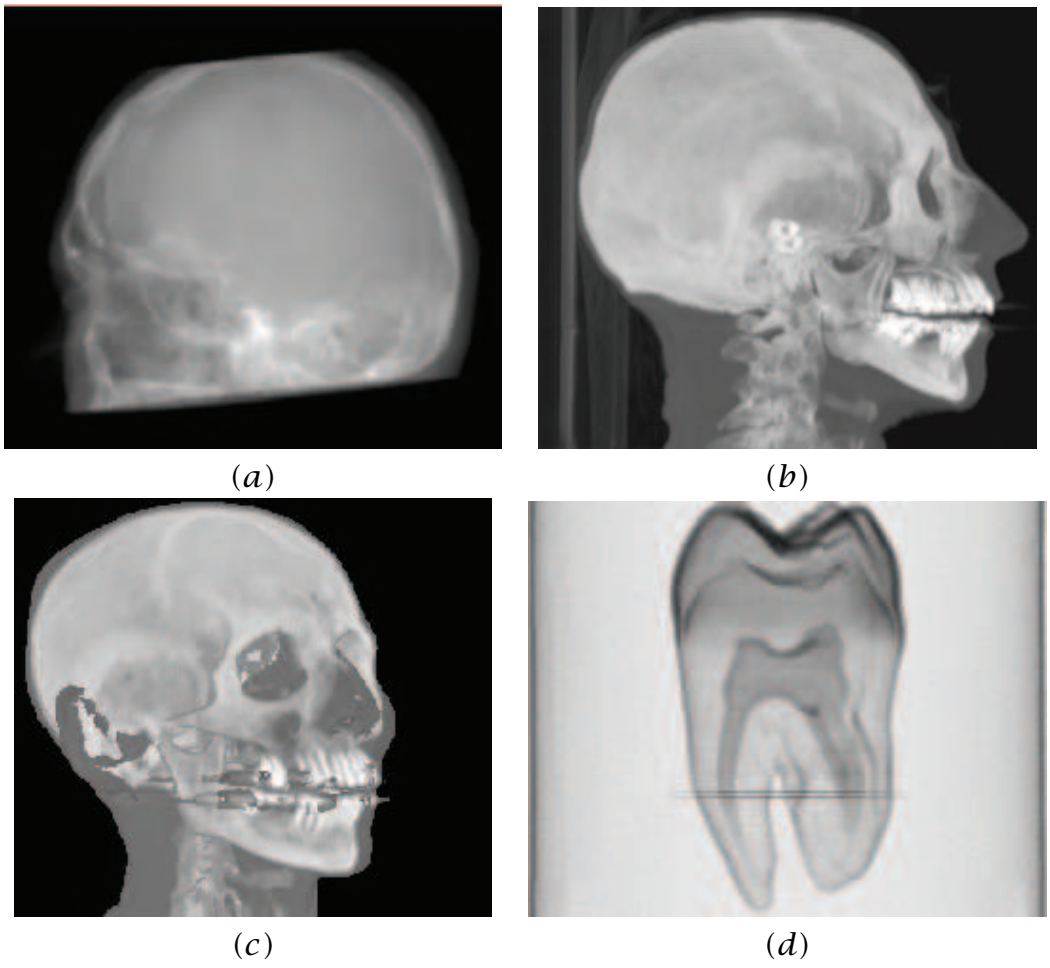


Figure 3.9: *Parameter-free rendering techniques: (a) summation, (b) maximum intensity projection [59], (c) local MIP [59], and (d) bubble model rendering [9].*

count scattering effects that lead to the appearance of translucency. More sophisticated models providing multiple scattering effects [57] are computationally very expensive. Recently an approximate model to increase the photorealism of rendered translucent volumetric data has been proposed by Kniss et al. [40]. The model takes into account forward scattering only, i.e., scattering only from the approximate direction of the light. As their implementation is completely mapped to GPU the rendering performance achieves interactive framerates. The result of rendering the volumetric dataset of legs is shown in Figure 3.10.

A different extension to direct volume rendering that focuses on in-

Figure 3.10: *Translucent direct volume rendering [40].*

creasing the visibility of features has been proposed by Bruckner et al. [3]. This technique is known as illustrative context-preserving volume rendering. The approach maps transparency to the strength of specular highlights. This allows to see *inside* the volume in the areas of highlights. The human perception can easily complete the shape of partially transparent parts and therefore additional information can be shown here. A further parameter tunes the ratio between specularity and transparency. A depth parameter determines how far one can look inside a volumetric object (fuzzy clipping). Certain data value ranges can be excluded from the transparency modulation to allow a clear view on specific (inner) structures. An example of illustrative context-preserving volume rendering is shown in Figure 3.11.

An extension of DVR to improve shape perception by moving of particles has been proposed by Lum et al. [54]. Additionally to direct volume rendering a set of moving particles is rendered on the surface shape. These particles are moving for example along the first principal curvature direction to optimally convey the shape. Figure 3.12 shows the PET scan of a mouse brain, where the shape becomes more clear with adding illuminated moving particles. The strength of this technique is

Figure 3.11: *Illustrative context-preserving volume rendering showing interior structures of a human hand [3].*

best demonstrated by animation sequences.

3.3 Focus+Context Visualization

Visualization tasks frequently emphasize a particular feature as opposed to the remaining context information. In medical visualization for example features interesting for the diagnosis are shown together with features in their close vicinity. Such visualization strategies are often denoted as *focus+context* visualization, where *focus* refers to the most interesting feature and *context* is the surrounding information to provide spatial or other referential relationships. In order to concentrate mostly on the focus information, the context often has to be represented in a sparse way that does not take-up too much of the viewport space. In the following several techniques are discussed that deal with visual representations for focus+context visualizations.

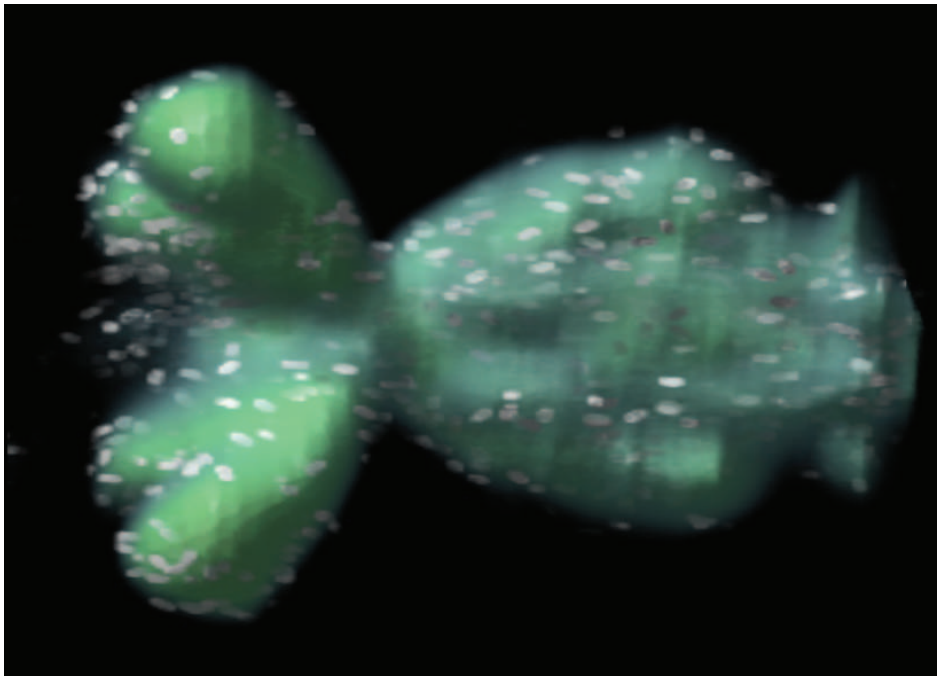


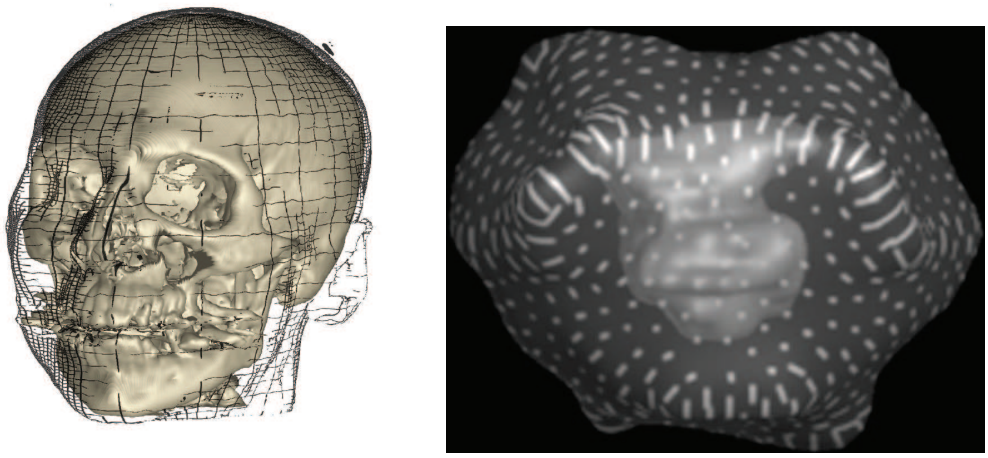
Figure 3.12: *Kinetic visualization of a mouse brain [54].*

Focus+Context in Scientific Visualization

In volume visualization sparse rendering techniques such as pen-and-ink illustrations were introduced to efficiently represent context information. Treavett et al. [76] use pen-and-ink styles in combination with direct volume rendering or surface shaded display. The sparse pen-and-ink representation is applied to outer iso-surfaces while an inner iso-surface is represented using surface shading. This is illustrated in Figure 3.13 (a).

Another approach for representing two surfaces, where the more interesting one is nested in the outer surface, incorporates curvature-directed strokes [32] for the outer structure. The inner surface is shown again as a shaded surface. Curvature-directed strokes are illuminated lines that effectively accentuate the shape via shading. Additionally the strokes are oriented in the direction of the principal curvature. The lines again do not take-up much of the image space. The outer shape is well presented while the inner surface structure remains clearly visible. An example of representing the context with curvature-directed lines is shown in Figure 3.13 (b).

An interesting approach for visualizing information from three-



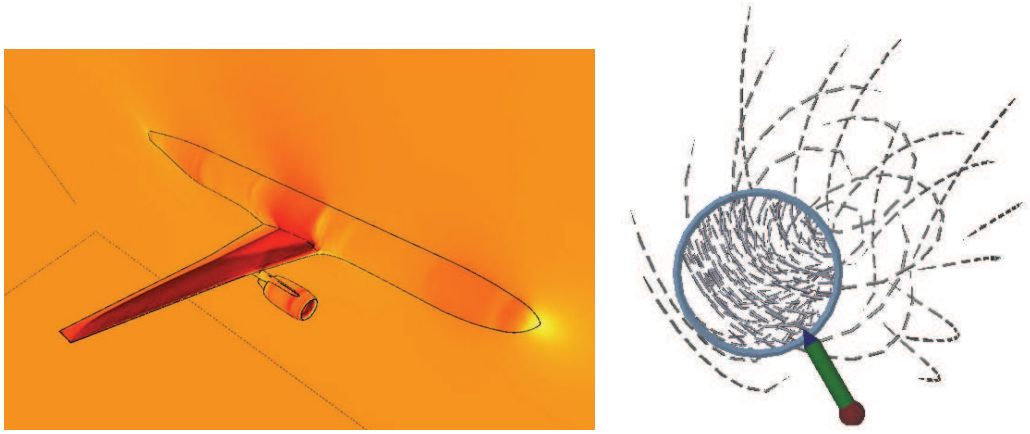
(*a*) (*b*)

Figure 3.13: *Focus+context visualization of nested objects: (a) a combination of a pen-and-ink technique with a surface shaded display of a human head [76], and (b) iso-intensity surfaces of radiation dose using illuminated curvature-directed strokes [32].*

dimensional unstructured grid data was presented by Ma and Interrante [55]. These data often come from simulations of particular flow properties, e.g., the aerodynamic properties of a particular aircraft. The visualization is more expressive when the aircraft outlines are present in the visualization. The work is presenting a technique for extracting feature lines which are contours and ridge-valley lines. Figure 3.14 (a) shows the visualization of flow properties including the aircraft which is represented by contour lines as the contextual information.

A combination of focus+context approaches with virtual reality environment for three-dimensional steady flow data has been presented by Fuhrmann and Gröller [17]. They represent the steady flow using dashtubes which are animated, opacity-mapped streamlines. The contextual information is represented by a relatively small number of dashtubes. These dashtubes give a rough sketch of the overall flow structure. To explore particular flow regions in a more detail, magic lens feature serves as a tool for focus area specification. The flow in this region is represented by a much higher number of dashtubes to enhance the visualization of local flow characteristics. The exploration of the steady three-dimensional flow using the magic lens is shown in Figure 3.14 (b).

A general approach to combine various visual representations for vol-



(*a*) (*b*)

Figure 3.14: *Focus+context flow visualization: (a) an airflow simulation over the right wing of transport aircraft enriched by feature lines of the aircraft [55], and (b) visualization of a three-dimensional steady flow data in virtual environments featuring magic lens exploration tool [17].*

ume rendering has been proposed by Hauser et al. [26]. The technique is known as two level volume rendering (2lVR). Well-known rendering techniques such as direct volume rendering (DVR), MIP, summation, or illustrative rendering with contour enhancement can be combined together. 2lVR renders each object within a volume with a different technique and composites the optical properties in a *local compositing step*. Each ray is partitioned by the intersecting objects into sub-rays. Local compositing is done for each sub-ray according to the rendering technique chosen for the respective object. The result of an entire ray is computed in a *global compositing step* which combines the results of the individual sub-rays. If the global compositing step uses DVR compositing, the formulas in Equation 3.3 are used. Basically the global compositing step can be based on any other compositing scheme as well. Figure 3.15 (a) shows a combination of different rendering styles using DVR as the global compositing step. An illustration of the 2lVR approach is also available in Figure 5.1 (a).

Tietjen et al. [74] presented a scenegraph based architecture that combines different rendering styles such as silhouette, surface, and volumetric styles. The advantage of a scenegraph is that it allows to combine several data representations such as volumetric data and convolution surfaces in one scene. The work furthermore describes the results of a

user study made with a group of medical experts. One conclusion is that silhouettes are an acceptable visual representation for contextual data in the vicinity of evaluated organs. Figure 3.15 (b) shows visualization of human abdominal organs using different rendering styles and data representations. The vascular structure inside the liver is represented by convolution surfaces the rest are volumetric data.

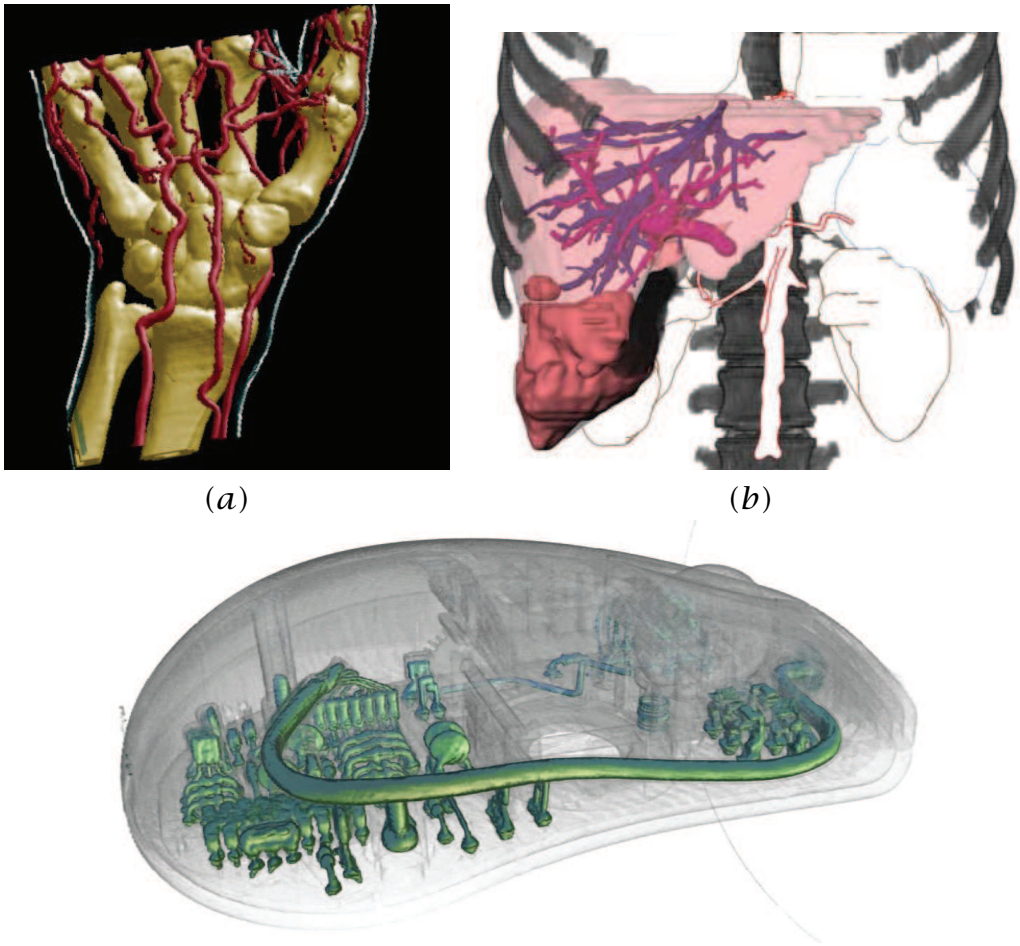
A pipeline that combines several illustrative effects to provide interactive visualization with emphasis on a focus object has been presented by Lum et al. [53]. Their implementation was mapped to the commodity graphics hardware of a computer cluster to guarantee interactive framerates for mid-sized volumetric datasets ($512 \times 512 \times 512$). An image of a rendering result combining DVR, contour enhancement, depth color cueing, and tone shading is shown in Figure 3.15 (c).

Focus+Context in Information Visualization

Information visualization is often concerned with the display of large, multi-dimensional, abstract data. In this area focus+context techniques are crucial to emphasize the small amount of relevant information among the typically very large overall data with multiple dimensions. In the following some approaches will be shortly discussed according to their relevance to this thesis.

One technique to focus the user attention to the most prominent information is called *semantic depth of field* (SDOF) [42]. Here the visual cue that targets the user's visual system is the level of sharpness of the presented objects. This is analogous to the depth of field (DOF) known from photography. The DOF function encodes the distance of an object from the focus plane of the camera. Objects close to the focus plane have a sharp appearance whereas objects farther away from the focus plane are blurred. SDOF simulates a semantic, i.e., relevance-based focus plane. Important objects are depicted sharply, while less important context information is blurred. The technique is illustrated by a chess example shown in Figure 3.16 (a). In semantic focus is the knight on $E3$ and those chessmen that cover it.

The second mentioned technique relates to the focus+context topic as well as to *smart visibility* using deformations which is discussed in Section 3.4. The technique performs viewpoint-dependent distortion of three-dimensional data to highlight data by dedicating more display space to it [7]. Distortions are applied to abstract graphs in order to clearly see interesting graph nodes. All nodes originally occluding the



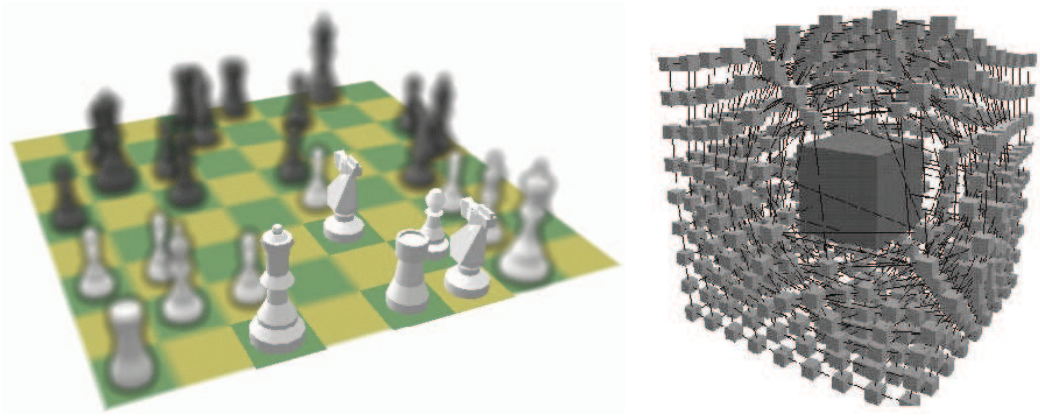
(*a*) (*b*)

(*c*)

Figure 3.15: *Combination of various rendering styles: (a) two-level volume rendering of the head dataset with contrast enhanced vessels [26], (b) combination of silhouette, surface and volumetric rendering of the human torso dataset [74], and (c) combination of various non-photorealistic effects for visualizing a computer mouse dataset [53].*

focus node are moved apart to uncover the most relevant information as shown in Figure 3.16 (b).

Almost every information visualization technique incorporates focus+context metaphor. A detailed discussion on broad spectrum of focus+context techniques is outside the scope of this thesis. Extensive overview of information visualization focus+context techniques can be



(*a*) (*b*)

Figure 3.16: *Information visualization examples using focus+context metaphor: (a) semantic depth of field [42] and (b) viewpoint-dependent distortions of three-dimensional graphs [7].*

found in the referenced literature [6].

3.4 Smart Visibility in Visualization

A typical problem in the visualization of three-dimensional or higher-dimensional data is that the most interesting features are not easily perceivable, because they are occluded by other, less important features. Traditional visualization techniques classify the visual representation of features independently from the viewpoint. The global setting limits viewpoint positions and viewing angles to a range, where the important structures are not occluded by other objects.

An interesting approach for the visualization of data obtained from multiple modalities has been proposed by König et al. [41]. Their viewport includes a three-dimensional view of the explored data plus three additional mirror-like projection views. These additional views can be arbitrarily customized to show different modality in different rendering styles. This technique has been proposed for exploring functional MRI data of the human brain.

An effective way to visualize three-dimensional data and resolve the occlusion of the most prominent information is to take approaches used in technical and medical illustrations [18, 29]. Illustration challenges are

very similar in this case. Illustration techniques such as cut-away views, ghosted views, or exploded views effectively uncover the most important information. This is achieved by changing the level of visual abstraction or modifying the spatial arrangement of features. In the following we describe illustrative visualizations that have been inspired by the illustration techniques from Section 2.2.

Cut-Away Views, Section Views, and Ghosted Views

The popularity of cut-away and ghosted views is demonstrated by the fact that they can be found in all books on technical or medical illustrations [18, 29]. An automatic generation of cut-away and ghosted views for polygonal data was introduced by Feiner and Seligmann [16]. They propose a family of algorithms that automatically identify potentially obscuring objects and display them in a ghosted or cut-away view. The proposed algorithms exploit z-buffer rendering, therefore they are suitable for real-time interaction achieved by hardware acceleration. Interactive semi-transparent views, section views, and cut-away views for polygonal data have been recently revised by Diepstraten et al. [13, 14]. Semi-transparent views unveil interesting objects obscured by other context information by increasing the transparency of the context. Diepstraten et al. propose to adhere to an effective set of rules for the automatic generation of the discussed illustrative techniques. For semi-transparent illustrative views the following three rules should be taken into consideration:

» faces of transparent objects never shine through
» objects occluded by two transparent objects do not shine through
» transparency falls-off close to the edges of transparent objects

For section views and cut-away views they propose to keep in mind seven other rules:

» inside and outside objects have to be distinguished from each other
» a section view is represented by the intersection of two half spaces
» the cut-out of a section view is aligned to the main axis of the outside object
» an optional jittering mechanism is useful for cut-outs
» a mechanism to make the walls visible is needed
» cut-outs consist of a single hole in the outside object
» interior objects should be visible from any given viewing angle

The mentioned algorithms and rules for cut-away views, section views, and ghosted views have been applied to polygonal data and are generally applicable in computer graphics. For an arbitrary clipping of volumetric data Weiskopf et al. [85] propose a number of effective techniques to increase performance and visual quality. The implementation of clipping operations is mapped to commodity graphics hardware to achieve interactive framerates. Additionally to clipping all rendering computations are performed on the graphics hardware. Per-fragment operations estimate on-the-fly the visibility according to the clipping geometry and adjust the shading in areas where clipping occurs.

Straka et al. [72] are applying a cut-away technique for CT-angiography of peripheral arteries in human legs. The goal is to have a clear view on the vessels, which are partially segmented by their centerline. For a clear understanding of the spatial arrangement it is necessary to visualize also bones and skin contours. To have an unobstructed view on the vessel for each viewpoint it is necessary to perform a cut in the bone. To avoid potential misinterpretations, the cut is clearly depicted as an artificial and sharp change in the data. This is illustrated in Figure 3.17 (a).

The previous application of cut-away views is viewpoint-dependent, i.e., the shape and location of the cut is directly dependent on the viewpoint information. Volume cutting is another medical visualization technique that is related to cut-away views, but the cut shape is not influenced by viewpoint settings. Pflesser et al. [64] present an interactive drill-like tool for surgical training, which is based on the multi-volume concept. Owada et al. [62] extend volume cutting by incorporating two-dimensional textures that are mapped on the cut surface. This enhances the visualization with additional information of the internal arrangement of bones or muscles. Such a concept can be very useful for anatomy education for example. Both volume cutting techniques are illustrated in Figure 3.17 (b) and (c).

Visualization of complex dynamical systems can be also enhanced by incorporating cuts into stream surfaces. Streamarrows proposed by Löffelmann et al. [51] exploit cutting for enhancing the visual information. They use arrows as a basic element for cutting away part of the stream surface. This allows to see through the surface and perceive other surfaces or structures behind. Animating streamarrows along the stream surface enables to see beyond the front stream surfaces and perceive the flow direction. Streamarrows belong to the category of view-point independent cut-away techniques and are shown in Figure 3.17 (d).

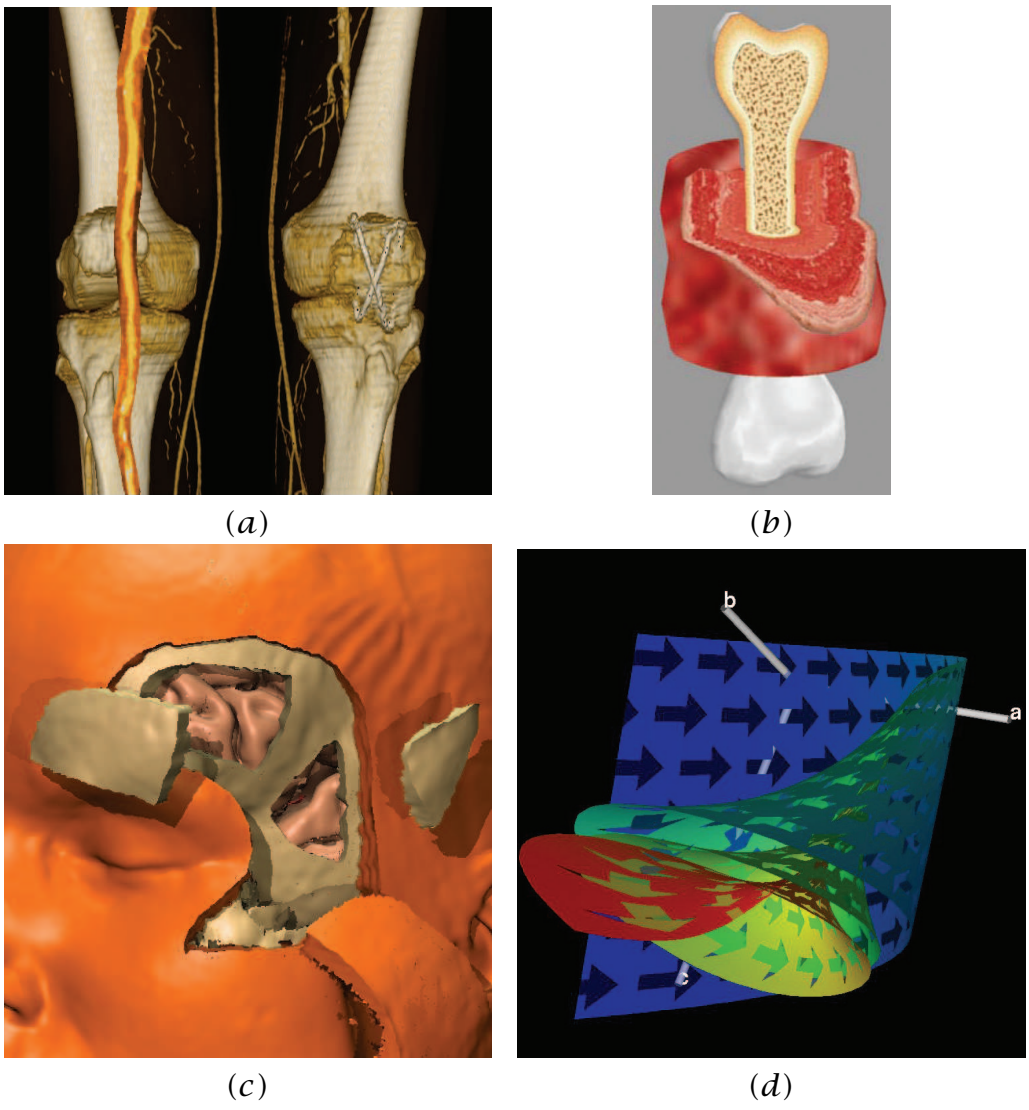


Figure 3.17: *Cut-away techniques: (a) peripheral vessels visualization [72], (b) volume cutting featuring two-dimensional textures for anatomy education [62], (c) volume cutting with a drill-like tool for surgical education [64], and (d) a streamsurface of a complex dynamical system with arrows as cutting element to enable to see structures behind the surface closest to the viewport [51].*

Exploded Views and Deformations

Exploded views and deformations modify the spatial arrangement of features to uncover the most prominent ones. It is also a very effective way

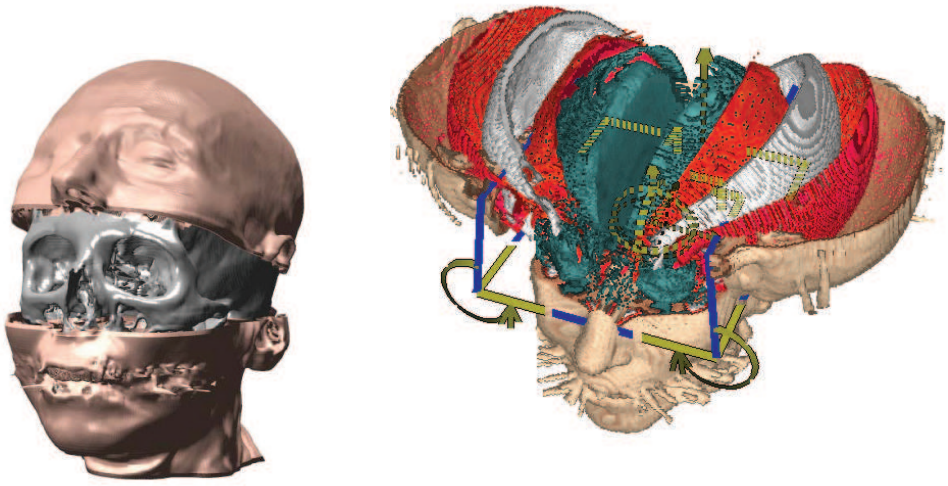
to present assembly instructions. Exploded views enable a clear view on individual features and convey the information about the original spatial location by helpers such as lines or arrows. Agrawala et al. [1] proposed design principles for creating effective assembly instructions based on exploded views. They additionally present a system for the automatic design of assembly instructions and a system that semi-automatically generates exploded views from two-dimensional images [50]. The rules for assembly instructions are based on cognitive psychology and experiments:

- » assembling is decomposed into a hierarchy of operations and parts
- » parts in the same hierarchy (e.g., legs of a chair) have to be added at the same time-step, or in sequence one after another
- » step-by-step instructions are better understandable than a single diagram showing all the operations
- » diagrams presenting the final assembly are necessary to understand the step-by-step action diagrams
- » parts added in the current assembly step must be clearly visible
- » objects have to be presented in their clearest orientation

Smart-visibility visualizations are using some of the above mentioned rules for other tasks than assembly instructions. In the following visualization approaches are presented that have been inspired by the exploded views concept. They use some of the rules for assembly instructions implicitly.

One of the visualization techniques that is closely related to exploded views is volume splitting [33, 22]. This technique is intended for displaying multiple enclosed iso-surfaces within the volumetric data. Each iso-surface, except the innermost one, is split into two parts and moved apart. Such splitting is denoted as logical splitting. Another type is geometrical splitting which moves apart the two halves of the entire volume. Logical splitting is illustrated in Figure 3.18 (a).

McGuffin et al. [58] propose an elaborate framework featuring a set of advanced deformations for an understandable visual presentation of complex three-dimensional information. The operation for investigating the interior of a volume is browsing. The browsing is realized on pre-segmented data decomposed into several semantic layers (e.g., skin, muscle, skull, brain). The user can cut into and open up, spread apart, or peel away parts of the volume in real time. This makes the interior visible while still retaining surrounding context. Additionally they present a set of interaction techniques based on various metaphors. Interaction



(*a*) (*b*)

Figure 3.18: *Deformations and spatial rearrangements of features: (a) volumetric splitting of the skin iso-surface [33], and (b) browsing of features through leafer deformation [58].*

techniques are controlled by pop-up menus and three-dimensional widgets. The interaction technique using leafing deformation is shown in Figure 3.18 (b).

Another interesting visualization technique inspired by exploded views is called fanning in time [22]. This technique is different from previously mentioned techniques, because it is a temporal exploded view analogous to temporal exploded views and multiple exposure photographs (Figures 2.11 (b) and (c)). It is useful for the visualization of time-series with a relatively small number of time-steps. The main goal is to show all time-steps in one image. Figure 3.19 (a) illustrates the idea of fanning in time.

The idea of time-series visualization can be very effectively applied for time-dependent flow visualization [89]. In contrast to fanning in time where all time-sequences are visible, this technique integrates the time-dependent information to a single image. This can be denoted as temporal *implosion* as all time steps are *imploded* into one spatial position. Such visualizations effectively indicate how values at particular sample position have changed over the time. This is illustrated in Figure 3.19

where (b) delta wing and (c) vortex datasets, are shown. Blue color coding indicates that the value fell over time within the time-series, red shows that the value rose over time, and green shows that the value rose and then fell in the time sequence.

3.5 Automatic Visual Enhancements

The last section of the state-of-the-art chapter is dealing with systems that evaluate particular characteristics of the scene in order to automatically improve the image quality. In the following two different scene optimizations are discussed, i.e., optimal viewpoint placement and optimal lighting settings.

Optimal Viewpoint

Techniques for optimal viewpoint placement have been investigated primarily for polygonal data. The most intensive research has been done by Vázquez et al. using the so called *viewpoint entropy* metrics [78]. This metrics incorporates both the projected area and the number of polygons, and can be understood as the amount of information captured by the viewpoint. The Shannon entropy of a discrete random variable X with values in the set $\{a_1, a_2, \ldots, a_n\}$ is defined as

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{3.5}$$

where p_i is the probability distribution of X. $0 \log_2 0$ is equal to 0 for continuity. The term $-\log_2 p_i$ represents the information associated with the result a_i, and therefore the entropy gives the average information of X. To define the viewpoint entropy for polygonal data the probability distribution is the relative area of the projected faces over the sphere of directions centered in the viewpoint. Thus, viewpoint entropy I is defined as:

$$I(p) = -\sum_{i=0}^{N_f} \frac{A_i}{A_t} \log_2 \frac{A_i}{A_t} \tag{3.6}$$

where N_f is the number of faces of the scene, A_i is the projected area of face i onto a sphere with center at point p, A_0 represents the projected area of the background in open scenes, and A_t is the total area of the sphere. The optimal viewpoint can be estimated by evaluating the

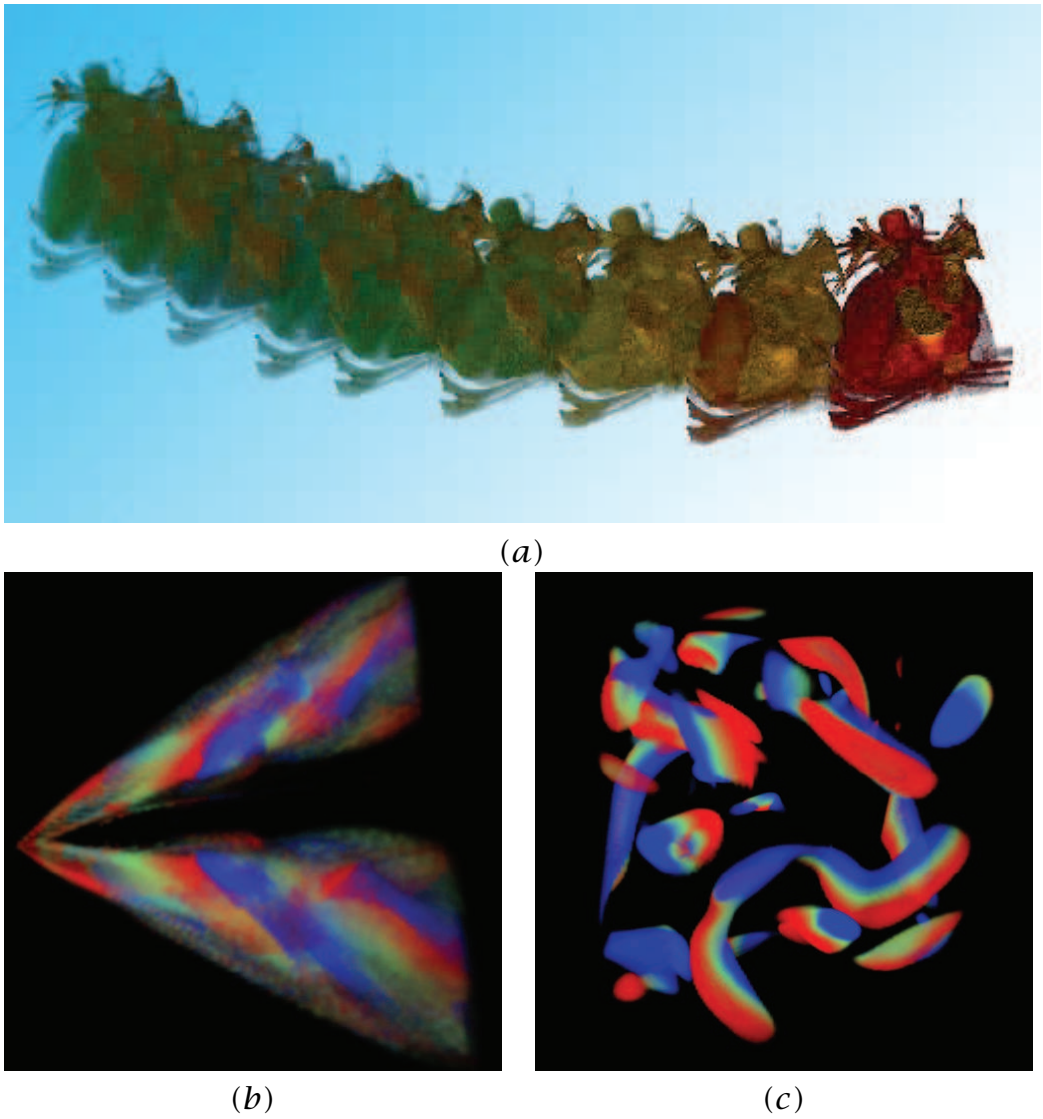


Figure 3.19: *(a) Fanning in time showing all time steps of the time-varying dataset of a beating heart [22] as an example for temporal explosion. Images below show temporal implosion of time-dependent datasets of (b) a delta wing dataset and (c) a vortex dataset showing the value change over the time-series by color coding.*

viewpoint entropy for a set of viewpoint settings, or some heuristics can be incorporated for speed-up. For example the viewpoint entropy can be first evaluated for a very small set of viewpoints (e.g. one for each orthog-

onal direction and orientation) located on a surface of a virtual sphere to keep the distance from the scene constant. Then the viewpoint with the highest viewpoint entropy is connected to all other viewpoints with known viewpoint entropy. In the middle between each two viewpoints (and projected onto the virtual sphere surface), viewpoint entropy is *predicted* by a heuristic prediction function. The approach is repeated until no viewpoint entropy prediction is higher than the existing highest viewpoint entropy.

Optimal viewpoint selection for the visualization of complex molecular structures shows that the viewpoint optimality criterion is a property highly dependent on the visualization task itself. In molecular visualization for example a combination of two specific views is interesting in order to efficiently convey information on the molecular structure [79]:

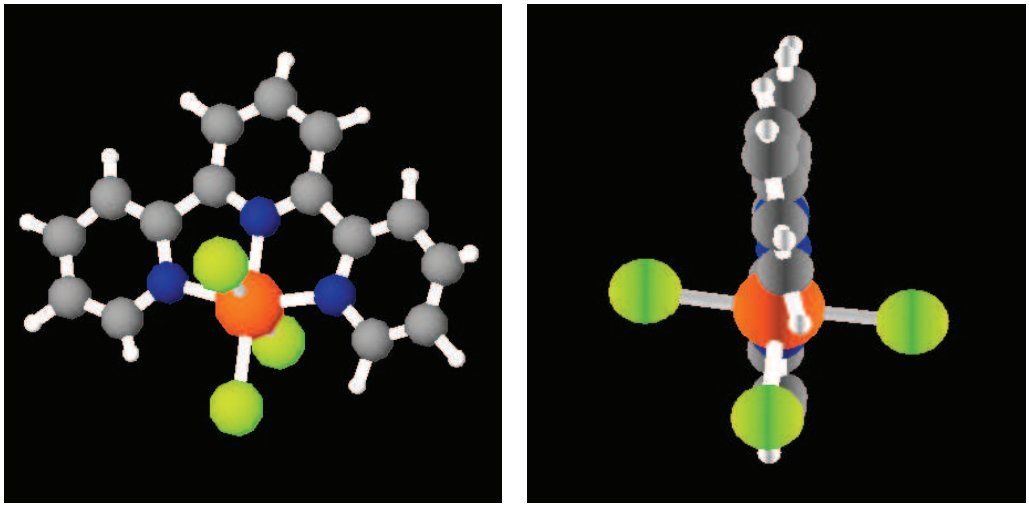
» A high entropy of single molecules gives the most informations of the molecule atoms as well as the distances and angles of the bonds.
» A low entropy of the molecule conveys the ordering and arrangement in space which determines physical properties of the molecules.

Figure 3.20 shows two different views of the same molecule. The combination of these two images with the highest and the lowest viewpoint entropy value conveys the information on the molecular structure in a very good way.

Lighting Design

Another scene optimization strategy as compared to viewpoint placement is the design of light placement. A light position is defined to be optimal, when the resulting illumination gives more information about the scene than the illuminations from all other light positions. This particularly maximizes the information that is added to the image through illumination.

Similar to the approach developed for viewpoint placement, Gumhold defines *illumination entropy* metrics [24]. In this approach only intensity values are considered, color and saturation are neglected. Therefore in the lighting design there is no degree of freedom in the choice of the light source color. First, a histogram of pixel intensities is computed under the current lighting setting. The illumination entropy is computed from Equation 3.5, where the probability of a particular intensity i is proportional to the fraction of the number of pixels with intensity i to the total number of viewport pixels. This concept was enhanced by results



(*a*) (*b*)

Figure 3.20: *Images of the same molecule from two different viewpoints: (a) viewpoint with the highest viewpoint entropy and (b) viewpoint with lowest viewpoint entropy. The combination of these two images gives a better information about the molecular structure [79].*

from user studies. An objective function for perceptual entropy has been developed that includes a weighting based on surface curvature.

Another lighting design for effective visualization has been introduced by Lee et al. [47]. This design also reflects principles of the human visual perception which have been exploited previously by artists and illustrators to enhance the perception of features through lighting. For different regions in space different lighting situations might be the best. The key idea is to use a lighting model that is locally consistent but globally inconsistent. The locally consistent lighting conveys the shape information of a particular object region by highlights, shadows, and silhouettes. Object regions, denoted as patches, are estimated according to their similarity in shape curvature. The mesh partitioning proceeds then analogous to the watershed segmentation discussed in Section 3.1 to create patches with larger areas. For each of these patches optimal lighting conditions are evaluated and applied. The illumination is blend between neighboring patches to avoid shape discontinuities caused by different lighting conditions. Figure 3.21 demonstrates the superiority of globally inconsistent lighting as compared to a globally consistent lighting design.

(*a*)

(*b*)

Figure 3.21: *Comparison of globally consistent to globally inconsistent lighting design. Four lights are placed to optimally convey local shape information: (a) globally consistent lighting design and (b) globally inconsistent lighting design [47].*

Chapter 4

Importance-Driven Visualization

Volume visualization is often dealing with the problem that interesting structures are partly or completely occluded by surrounding objects. This is hard to resolve by traditional view-independent approaches, such as transfer function specification. This thesis proposes a viewpoint-dependent model that suppresses unwanted occlusions automatically and maximizes the information content in the final image [82, 83].

4.1 The Model

Interesting structures within the volumetric data are denoted as *features* respectively *objects*. The specification can be done in many different ways, also depending on the type of the data. Some feature specification approaches have already been discussed in Section 3.1. In medical visualization features often correspond to particular organs. Such objects are defined by a segmentation process. Another way of feature classification can be done through defining a spatial relationship within the volume or it is given through the location with respect to another feature. In such a way it is for example possible to classify the vortex core of a hurricane. In the case of multi-dimensional volumetric data, features can be defined by specifying interesting ranges of values for each data dimension. There are many other ways to determine features.

Traditionally features within the volume dataset are classified by optical properties such as color and opacity. Importance-driven visualization additionally assigns another dimension to features, which describes their *importance*. Importance encodes which features are most interesting and have the highest priority to be clearly visible. Each feature is therefore weighted by a positive scalar value called ***object importance***. During the rendering stage, the model evaluates the visibility of each feature according to its importance. If less important structures are oc-

cluding features that are more interesting, the less important ones are rendered more sparsely, e.g., more transparently. If the same object does not cause any unwanted occlusions in other regions of the image, it is rendered more densely, e.g., opaque, in order to see it more clearly. All interesting structures are visible irrespective if they are surrounded by other structures or not, and the less important parts are still visible as much as possible.

Instead of using constant optical characteristics, which are independent from the viewpoint, several ***levels of sparseness*** are assigned to each object. Thus, instead of a single optical characteristic, the object has several characteristics with smooth transitions inbetween. These multiple levels of sparseness allow the object to continuously change its visual appearance from a very dense representation to a very sparse one. Which level of sparseness will be chosen is dependent on the importance of the particular object and the importance of objects behind it. The level of sparseness may remain constant for a particular object or may continuously vary within a single object. Cut-away views are an example where the level of sparseness changes discontinuously. Also depending on the viewpoint and the viewing situation the same part of an object may be represented with different levels of sparseness.

To determine the sparseness level for each object or parts thereof, the rendering pipeline requires an additional step, denoted as ***importance compositing***. This step evaluates occlusions, takes the importance factor of each object into account and assigns to all objects particular *levels of sparseness*. The final synthesis results in images with maximal visual information with respect to the predefined object importance.

For the sake of clarity the defining components of importance-driven visualization are now briefly summarized:

» **Object importance** describes the visibility priority of each object within the volume dataset. It is for example a positive scalar value, which is constant for the whole object.

» **Levels of sparseness** are various different representations of a particular object from the most dense to the most sparse one. Sparseness is defined in terms of how much the display of an object takes up screen estate. For an iso-surface a point-cloud or a wire-frame display is a very sparse representation. The display of filled opaque polygons is a very dense representation of the same iso-surface. In case of volumetric data each sample within the volume is classified to have a color and opacity. In direct volume rendering (DVR) these values are composited together to synthesize the final image. The

sparseness can in this case modulate the opacity of object samples so that the most interesting samples are opaque while other samples are more transparent. This is a simple and straightforward example of levels of sparseness. More elaborate schemes are discussed in Section 4.3.

» **Importance Compositing** is an additional step in the rendering pipeline that assigns to each part of an object a specific *level of sparseness*. A very simple importance compositing scheme is maximum importance projection (MImP). For each image area only the object with highest importance is visible. The remaining objects are shown with the highest level of sparseness, i.e., fully transparent. As another example, importance compositing can be performed in a similar way as compositing of color and opacity in the traditional DVR approach. Instead of compositing optical properties, however, the *object importance* values are accumulated. Different compositing modes are discussed in Section 4.2.

The relationship between the above mentioned components is depicted in Figure 4.1. *Importance compositing* is done similar to the direct volume rendering (DVR) approach. For each ray the compositing step evaluates object occlusions and assigns to each object the corresponding *level of sparseness*. Object importance is preserved in the sense that it is mapped to object visibility in the resulting image. This causes different rendering settings for the context object (importance value 0.1 in Figure 4.1) in the area of the image which is covered by the focus object (importance 0.7).

Importance-driven visualization is an automatic focus+context approach that can be applied to globally or locally suppress occluding context features and thus emphasize the focus information. The difference between global feature suppression and local suppression inspired by cut-away views is shown in Figure 4.2. The goal is to emphasize the inner organ as focus object. In the global suppression it is necessary to reduce the opacity of the entire occluding body. The cut-away metaphor suppresses only the context part which occludes the focus object while the non-occluding context information is represented densely.

The following sections discuss the elements of the importance-driven visualization pipeline in the context of cut-away views. However, the discussed concepts of *importance compositing* techniques and schemes for *levels of sparseness* can also be applied for global suppression in a straightforward way.

The first element that provides information on the prominence of a

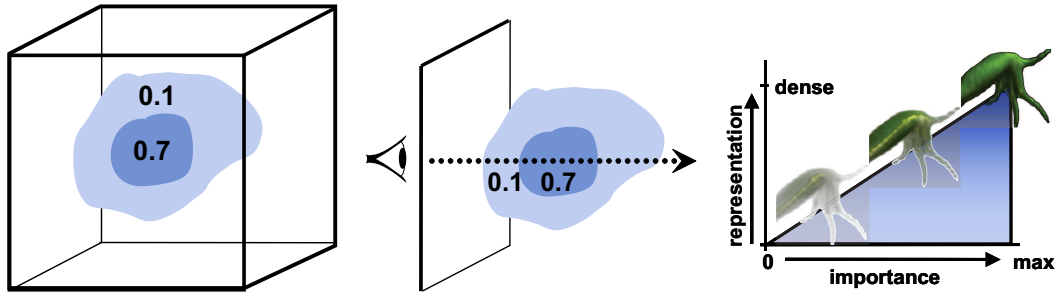


Figure 4.1: *Basic elements of importance-driven volume rendering: Volumetric features are classified by (a)* importance *values. The volume is traversed in the (b)* importance compositing *step. Then (c)* levels of sparseness *are chosen to enhance or suppress particular parts of the volume.*

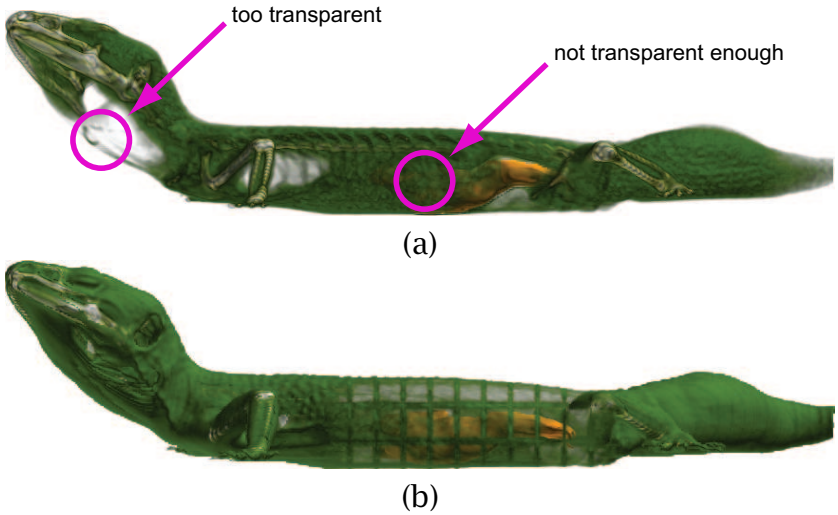


Figure 4.2: *Comparison between (a) a global suppression of a visual representation and (b) a local suppression.*

particular feature is *object importance*. It represents a function that maps the degree of prominence to features in order to determine which features have to be emphasized and which have to be suppressed. In related work [25] such a function is also denoted as *degree-of-interest* function (DOI). Various possible ways for feature classification have been already sketched in Section 3.1. This chapter focuses more on the remaining two elements of importance-driven visualization. First various ways of *importance compositing* are explained and then possible schemes for continu-

ous or step-wise transitions in the visual abstraction denoted as *levels of sparseness* are presented.

4.2 Importance Compositing

Importance compositing is an additional pass added to the traditional volume rendering pipeline. It determines the level of sparseness for each object or a part thereof in order to preserve important features. There are many possibilities conceivable how to perform importance compositing. In the following two methods of importance compositing are discussed, which are inspired by traditional compositing of optical properties through ray casting of volume data.

Maximum Importance Projection

Maximum intensity projection (MIP) [59] is a simple and fast volume rendering approach. With MIP compositing reduces to selecting the highest intensity value along a ray. It is applicable for sparse data where important information has high intensity values such as contrast-media enhanced blood vessels. Intensities are encoded as gray values to produce the final image.

Analogous to MIP we perform maximum importance projection (MImP). For each ray the object with highest importance along the ray is determined. This object is displayed densely. All the remaining objects along the ray are displayed with the highest level of sparseness, i.e., fully transparent. With MImP structures are either rendered using the most dense representation or they are not rendered at all.

With MIP the spatial arrangement of structures is not readily apparent. MImP has a similar problem, which can be alleviated as follows: The image area, where the most important object is projected onto, is denoted as object *footprint*. With MImP the footprint is exactly the image region where only the focus object is visible. One can consider MImP as a cut-away view, where the space in front of the most important object is simply clipped. The clipping region is a translational sweep with the footprint as cross section (general cylinder). One can now modify this cylinder to obtain a clipping frustum. This is achieved by scaling up the footprint during the translational sweep towards the viewer. This produces a cut-out clipping geometry. Figure 4.3 illustrates the difference between the cylindrical and the conical MImP in 2D. The conical MImP is easily realized during ray traversal by changing the starting point for

those rays that intersect the side faces of the clipping frustum. Figure 4.4 shows images to compare both approaches. The cylindrical MImP does not clearly show the spatial relationship between focus and context objects, i.e., the focus object appears in front of the context object. Conical MImP corrects this visual artifact. A detailed description how to generate the cut-out is given in Section 4.4.

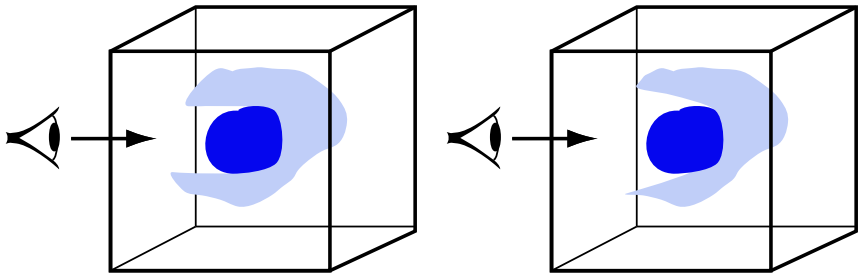


Figure 4.3: *Maximum Importance Projection. Illustration of (a) cylindrical and (b) conical cut-out.*

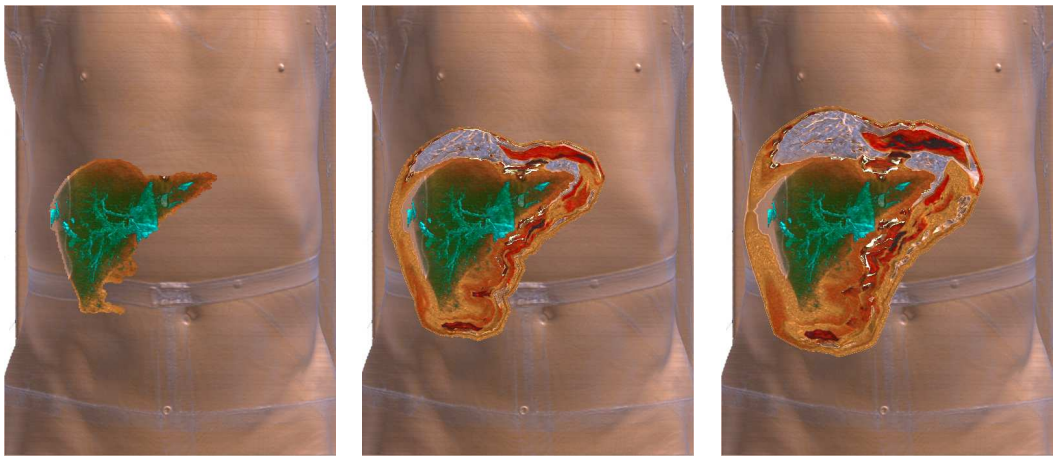


Figure 4.4: *Maximum importance projection (MImP): (a) cylindrical MImP and (b) (c) conical MImP with different slope factors .*

Average Importance Compositing

The second approach of importance compositing takes into account all the objects along a ray. The influence of an individual object is hereby independent of the number of ray samples within the object. An object o has an importance value I_o. Ray r intersects n_r objects. The level of sparseness S_o of a particular object o at ray r is equal to the fraction of its own importance and the sum of the importance of all the intersected objects:

$$S_o = \frac{I_o}{\sum_{i=1}^{n_r} I_i} \tag{4.1}$$

Average importance compositing (AImC) does not completely remove the less important objects as with MImP. The sparseness factors are estimated according to the given importance. This allows a very sparse representation of the occluding object to still see a rough outline of its shape and to clearly see the important object behind it. The importance compositing stage computes the levels of sparseness of all objects for every pixel in the final image. Levels of sparseness are computed using the object footprints. Object importance values of all objects that cover the current pixel position are summed up. The sparseness factor of each of these objects is estimated through division of their object importance by the evaluated sum (Equation 4.1).

The final image synthesis using AImC is an extension of traditional DVR. At each sample location during ray traversal the level of sparseness additionally modulates the visibility of the sample. Similar to cylindrical MImP, AImC generates images where the spatial arrangement of structures is negatively influenced by the discontinuous transitions between context and focus image areas. In order to improve spatial perception two methods are proposed to perform final importance-driven image synthesis with AImC, i.e., an image-space and an object-space approach.

Image-Space AImC: The object footprints introduce sharp transitions in levels of sparseness, which might reduce spatial perception. To improve on this smooth transitions between different levels of sparseness are generated. Before the levels of sparseness are computed for each object, morphological image operators are applied to every footprint, i.e., dilation and averaging. As pixels in the footprint have a weight of one, the weights in the generated transition zone are smoothly decreasing to zero. The levels of sparseness estimation is analogous to Equation 4.1. For each pixel the footprint-weighted importance sum of all contributing objects is computed. The object importance is in this case always multiplied by the footprint value in the range of $[0, 1]$. Footprint values below

one are part of the *transition* area between different levels of sparseness.

The image-space approach does not evaluate whether the sample of a suppressed context object is in front or behind the important object. The level of sparseness is constant for all samples of an object along a particular ray. This means that also part of the context object *behind* the focus object is unnecessarily suppressed (see Figure 4.5).

Object-space AImC: To avoid suppression of context behind the focus object, a more costly object-space approach is proposed. Using this approach only those samples of the context object are suppressed that are *in front* of the focus. In this case the level of sparseness is not constant for an object along a particular ray. The difference between the image-space and the object space approach is illustrated in Figure 4.5. The Figure shows that image-space AImC suppresses all context samples along a ray. The object space approach suppresses only the part of the context object that occludes the focus object.

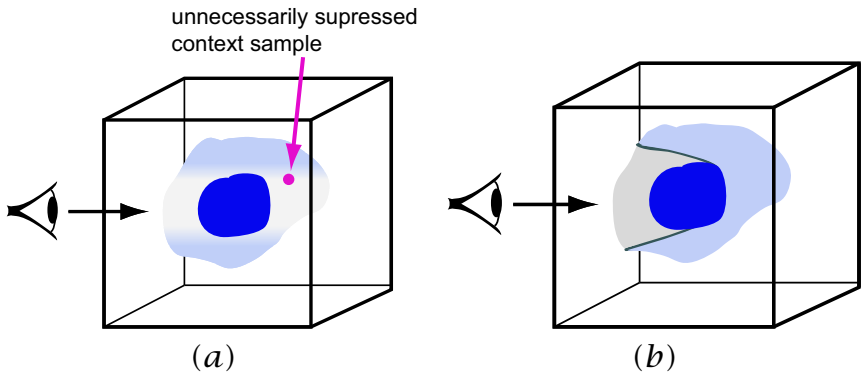


Figure 4.5: *Average importance compositing (AImC). The illustration depicts the difference between the (a) image-space and (b) the object space approach.*

The image synthesis of the object space approach is analogous to the conical MImP. In the case of the conical MImP the cut-out geometry is used to estimate the starting position of the ray in order to perform the cut-out. In object-space AImC the cut-out geometry defines the border between different levels of sparseness. The starting position of the ray is not changed. During the ray traversal in the final image synthesis, each sample location is evaluated whether it *belongs* to the cut-out region or not. The context outside the cut-out is depicted with a more dense representation and inside a more sparse representation is chosen.

The image-space AImC approach is shown in Figure 4.6. Figure 4.7

depicts the object-space AImC. The images show the same dataset under different viewing angles.

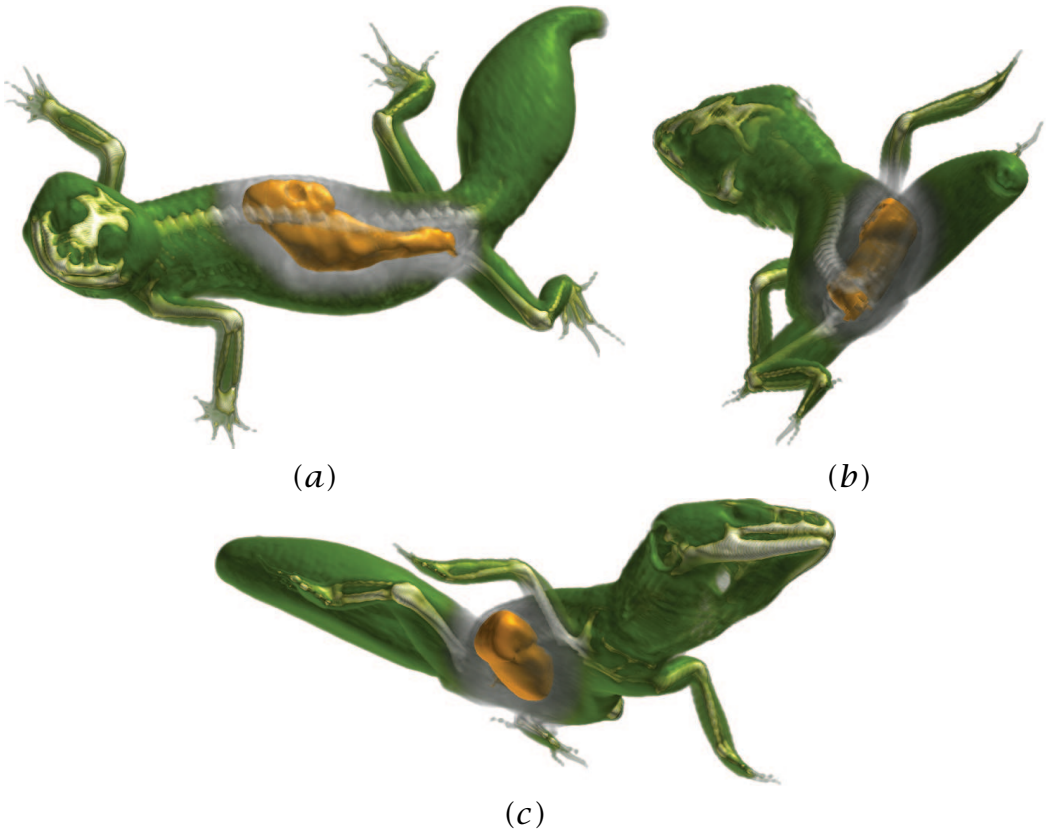


Figure 4.6: *Image-space average importance compositing is shown under different viewpoint settings (a), (b), and (c) in combination with modulating optical properties.*

The AImC approach preserves the *thickness* of the occluding part of the context object. This leads to different visibilities of the focus object under different viewing conditions. If the occluding context area is too thick, the focus object is not visible enough. In order to see the focus object *properly* the level of sparseness function has to be changed for the context object or the importance of the focus object has to be increased. Thus the user has to adjust these properties because an automatic *visibility evaluation* is not considered here. In Chapter 5 a simple technique for an automatic *visibility evaluation* is introduced. Additionally techniques to automatically overcome the problem of a varying thickness of

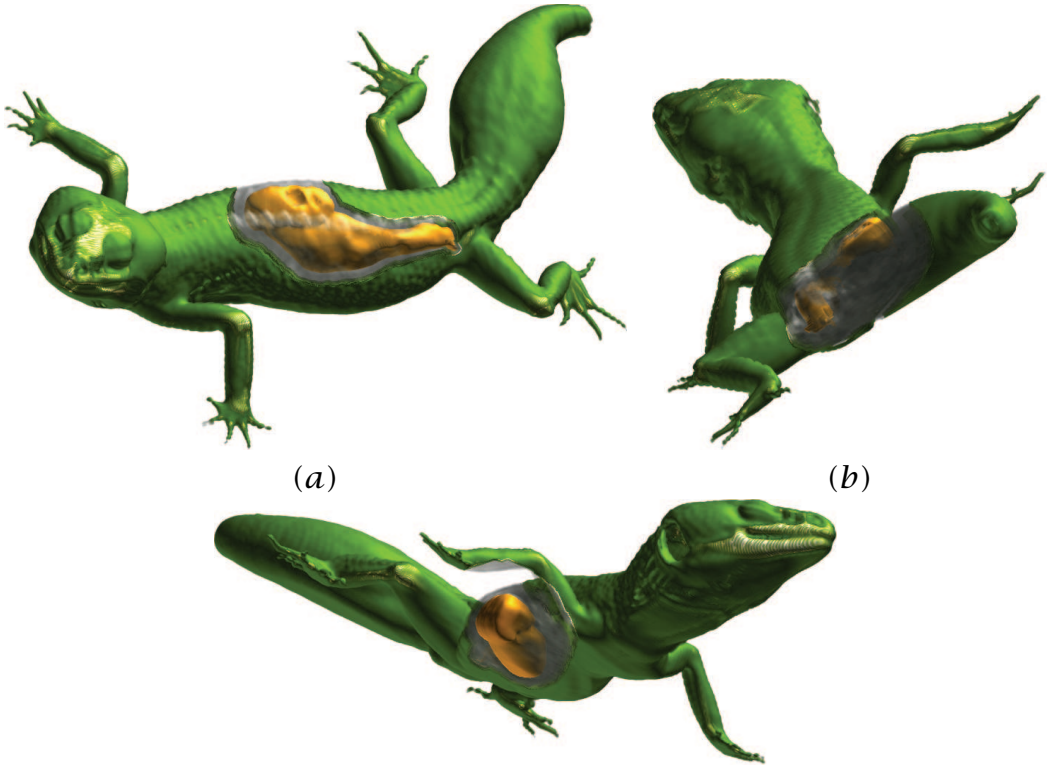


Figure 4.7: *Object-space average importance compositing is shown with the same viewpoint settings (a), (b), and (c) as Figure 4.6.*

the occluding object are presented to preserve an approximately constant visibility of the focus object.

4.3 Levels of Sparseness

Importance compositing determines for each part of an object its visibility in the rendered image. This is achieved by computing the appropriate level of sparseness. In the following four types of levels of sparseness are described. Three of those are depicted in Figures 4.8, 4.9, and 4.10. The series of images illustrates how the context area in front of the focus object smoothly varies from a dense to a sparse representation.

Color and Opacity Modulation

A direct control of optical properties is the first approach to modify the visual prominence of a particular feature. With increasing sparseness the object becomes more transparent in order to show the more important underlying data. This approach is widely used in transfer function specification.

Interesting results can be achieved by controlling color saturation through levels of sparseness. In general color is a very important visual cue in visualization. Highly saturated colors attract the observer's attention more than colors close to gray. The level of sparseness can therefore be expressed also in the saturation of the color. Changing only the saturation, however, does not increase the visibility of occluded objects. It is necessary to change the color and opacity values at the same time. Different visual appearances within the same object can cause misinterpretations. Therefore smooth transitions between different levels of sparseness have to be applied. A smooth modulation of the optical properties is shown in Figure 4.8.

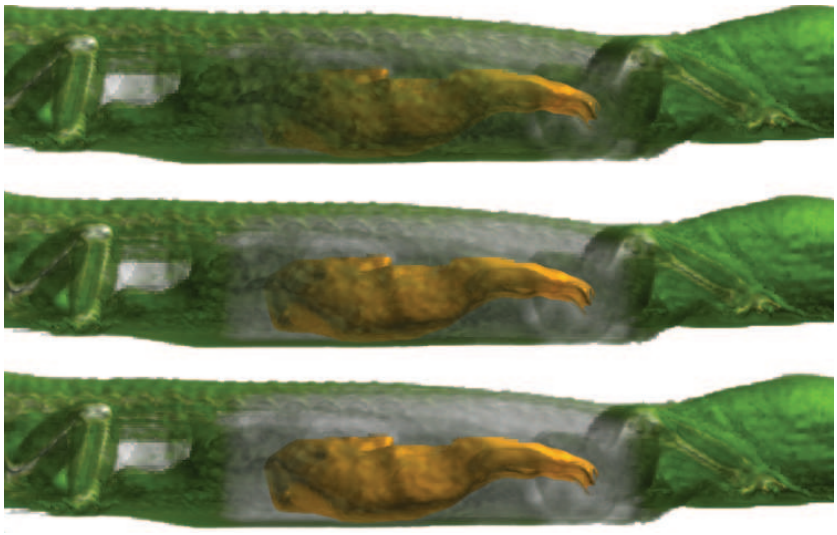


Figure 4.8: *Levels of sparseness using opacity modulation and color saturation modulation.*

Screen-Door Transparency

Screen-door transparency is a well-known strategy to simulate transparency. The appearance of an object behind another semi-transparent

object is simulated with a screen-door as follows: A screen-door consists of a wire-mesh and holes inbetween. The wires of the mesh depict the first object, while the second object is visible through the holes. From a certain distance holes and wires blend together to produce a semi-transparent impression. An analogous idea can be used to define levels of sparseness. The volumetric dataset consists of voxels. The level of sparseness determines which voxels should be rendered and which not. The arrangement of visible voxels is uniform and is forming a 3D wireframe structure. The impact of increasing sparseness is shown in Figure 4.9.

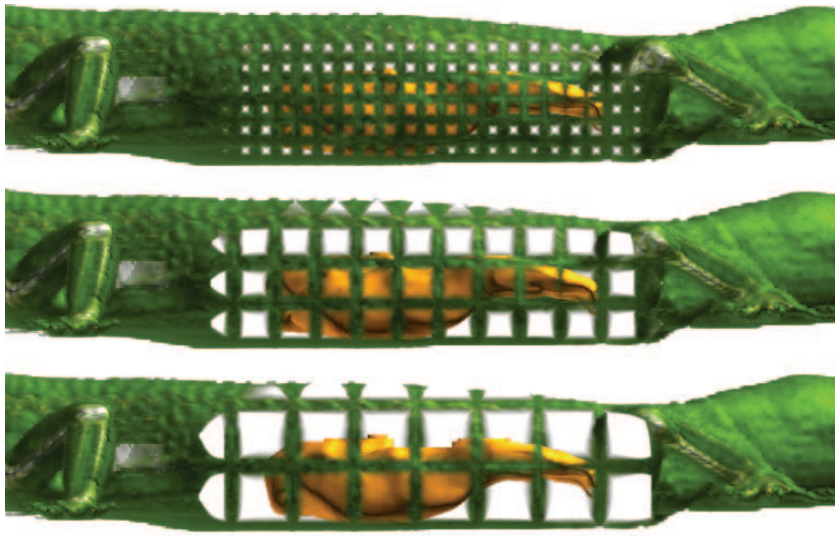


Figure 4.9: *Levels of sparseness using screen-door transparency.*

Volume Thinning

Volume thinning proceeds as follows: Voxels of an object are sorted according to two sorting keys. The first sorting key is gradient magnitude, the second sorting key is curvature magnitude of the iso-surface through the voxel. Increasing the sparseness according to gradient magnitude has the effect that the volume is continuously reduced to fewer and fewer strong iso-surfaces. As soon as only few iso-surfaces remain the reduction proceeds according to curvature magnitude. This has the effect that the iso-surfaces gradually dissolve and in the end (most sparse representation) only few high curvature areas on strong iso-surfaces remain. Figure 4.10 illustrates visibility reduction through volume thinning.

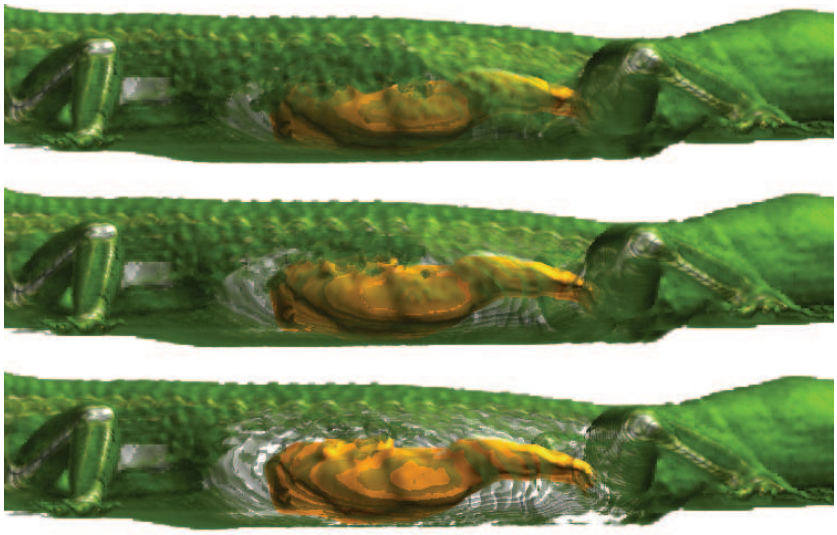


Figure 4.10: *Levels of sparseness using volume thinning.*

Sparse and Dense Rendering Styles

The previous levels of sparseness techniques describe how to enhance/suppress the visual representation of a particular object. The sparseness function smoothly varies from the most dense to the most sparse representation.

Another approach is to assign different *rendering techniques* as different levels of sparseness. For example the dense representation is achieved by direct volume rendering and the sparse one by illustrative contour rendering. In the case of object representations, combination is done via compositing. The combination of different rendering techniques is achieved through two level volume rendering [26] (see Section 3.3). The difference between levels of sparseness based on object representations and based on rendering styles is shown in Figure 4.11.

Results of using different rendering techniques as levels of sparseness are shown in Figure 4.12. Where no occlusion occurs the context information is rendered using traditional DVR. In the case of occlusion of the inner structure a different sparse rendering technique is applied. The images show the application of summation, contour enhancement, and maximum intensity projection for the local compositing. Global compositing is done by using again DVR.

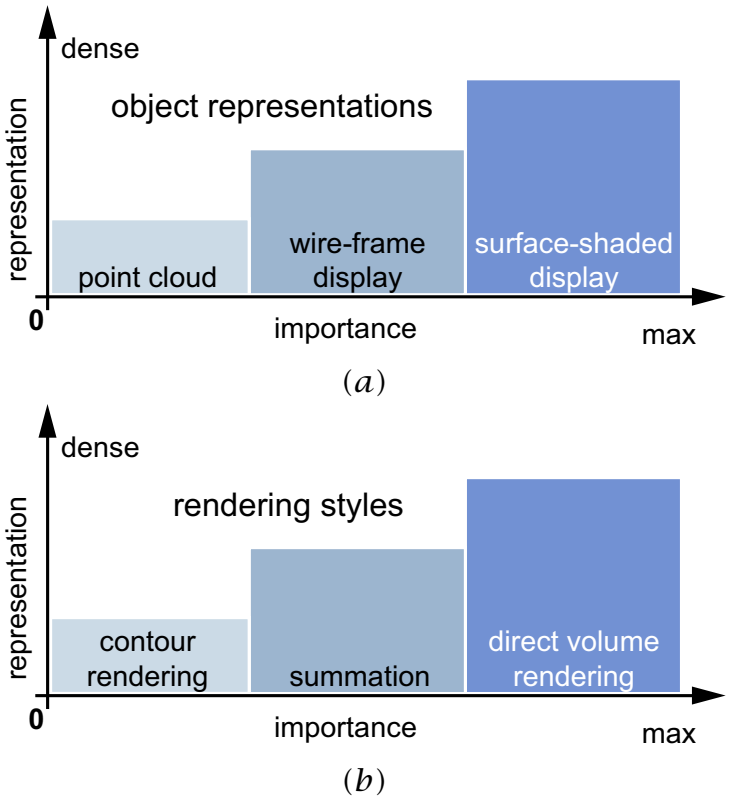
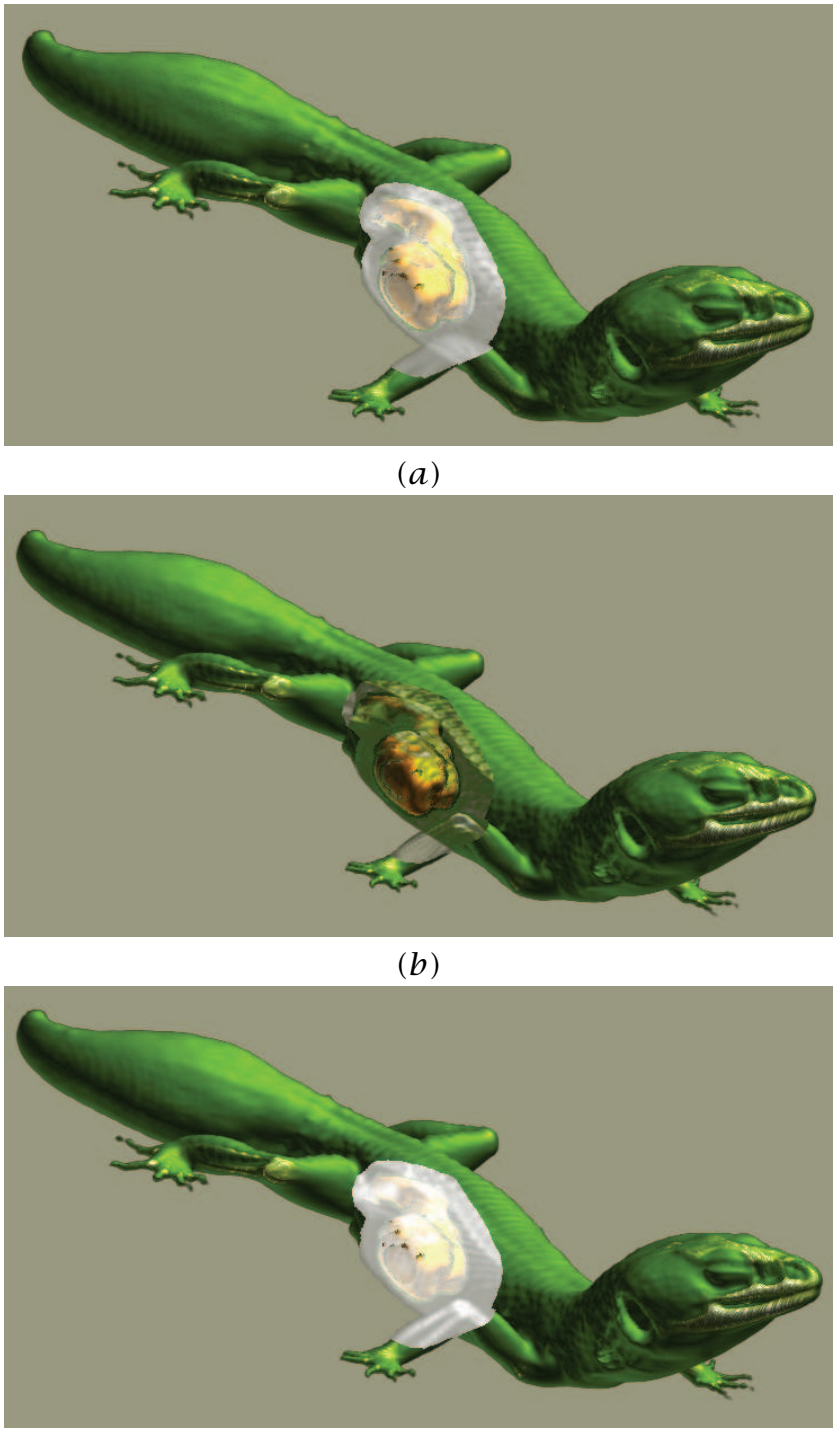


Figure 4.11: *Examples of two types of* levels of sparseness*: (a) based on object representations and (b) based on different rendering techniques.*

4.4 Cut-Away Generation

Previous sections were discussing importance compositing techniques and the levels of sparseness schemes. Both components of importance-driven visualization have been demonstrated on local feature enhancement resulting in cut-away views, but are applicable for the global feature enhancement as well. This section is relevant for the local feature enhancement, because it describes the cut generation used in the local feature enhancement.

The generation of the cut-away effect can be achieved in several ways. One possibility is to generate a geometry representation of the cut-out region and perform a test whether the current ray position is inside or outside of the cut-out geometry. This can be effectively implemented by introducing a *cut-out depth buffer* which contains the depth information



(*a*)

(*b*)

(*c*)

Figure 4.12: *Sparse and dense rendering styles: The occluding context object is rendered using (a) summation, (b) illustrative contour enhancement, and (c) maximum intensity projection.*

of the cut-out geometry. During the ray traversal a cut-out test is integrated to estimate what level of sparseness should be applied to a particular sample position.

The *cut-out depth buffer* is computed from the *footprint* of the focus object, i.e., those viewport pixels where the focus object projects to. The footprint contains depth information of the focus object's *last hit* for each ray along the viewing direction. The cut-out depth buffer is initialized with this value which gives the farthest distance of the focus object to the viewer (e_{max}). To realize a conical clipping geometry the footprint in the cut-out depth buffer is enlarged to represent the conical side-walls of the cut-out geometry. This is done by applying a 2D chamfer distance transform to the cut-out depth buffer [2]. First e_{max} the maximal depth value of the cut-out depth buffer is calculated. The depth values of the conical part of the cut-out e_i are calculated from the maximal depth value e_{max}, the slope factor s_c of the cut-out, and the image space distance from pixel i to the footprint d_i as shown in Equation 4.2:

$$e_i = e_{max} - \frac{d_i}{s_c} \tag{4.2}$$

This generation is also illustrated in Figure 4.13, where e_{max} is represented by the darkest blue color, d_i is the distance of a pixel from the footprint and slope s_c is illustrated as a color difference. Figure 4.13 (a) shows a conical cut-away effect and Figure 4.13 (b) illustrates the corresponding cut-out depth buffer.

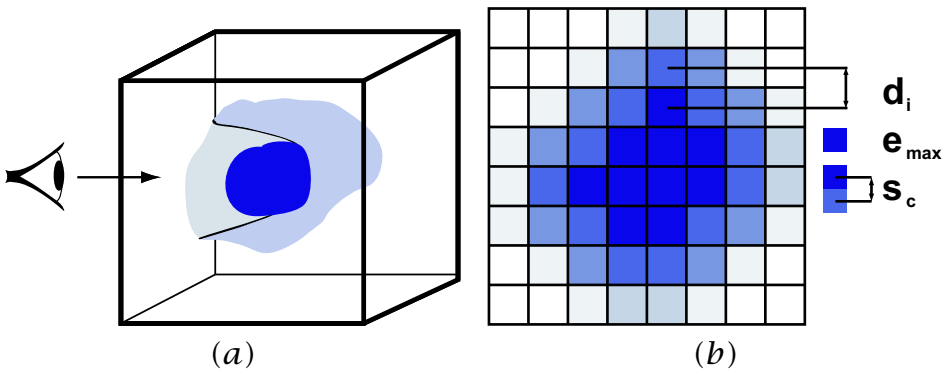


Figure 4.13: *(a) Generated local feature enhancement via cut-away view and (b) the corresponding cut-out depth buffer generated from the focus object.*

For hierarchically represented volumetric data such as given by an octree structure, there is also another fast and approximate way to generate

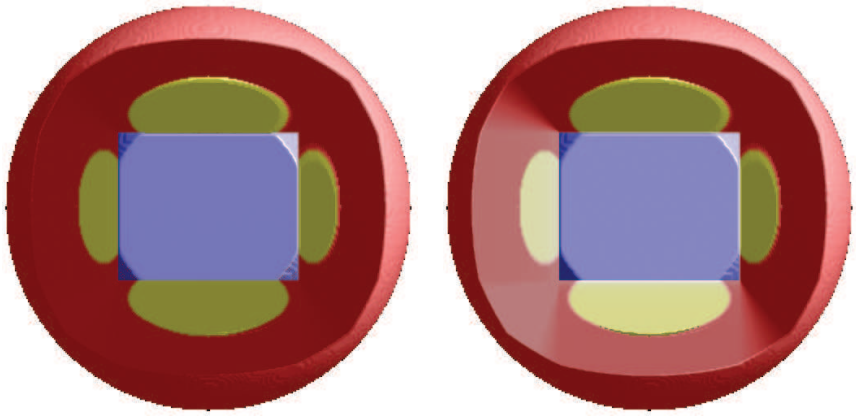
the cut-out. Instead of generating the cut-out depth buffer from the footprint of the focus object, represented by a set of voxels, it is possible to generate the buffer from the footprint of focus object's bounding boxes. These bounding boxes are readily available in the octree hierarchy. The last hit e_{max} is stored in the cut-out depth buffer as in the previous case. To avoid rectangularly shaped cut-outs the depth buffer is processed by a smoothing image processing operator. The cut-out depth buffer in this case as compared to the distance transform approach is generated faster, as it is built from higher hierarchy levels.

Generating the cut-outs provides results with a correct spatial arrangement of features, i.e., the focus features do not appear in front of the context ones like in case of cylindrical MImP (Section 4.2). For the realization of MImP ray samples that belong to the context object are simply skipped, until the ray reaches depth e_i stored in the cut-out depth buffer. For object-space AImC, samples along the ray with a depth smaller than e_i are not skipped, but they are represented sparsely.

The side-walls need to be shaded according to the cut-out geometry. This requires proper computation of the surface normal at the cut-out region. In the case of volumetric data the normal vector (or gradient) at particular sample position is estimated from samples from its local neighborhood. However, at the cut-out walls this normal vector will result into incorrect shading. Correct shading requires the a vector perpendicular to the side-wall geometry. In case of very opaque representation of the volumetric data along the cut-out this will result in proper shading. However in the case of semi-transparent samples at the side-wall the correct shading will be not sufficiently achieved. Therefore Weiskopf et al. [85] propose to continuously modify the normal vector values in a small region *behind* the cut-out side-walls. The normals from this region are computed as a linear combination of the cut-out normal and the normal (i.e., gradient vector) estimated from the local neighborhood of the processed sample.

In the importance-driven local feature enhancement the surface normals of the cut-out side walls are calculated from the cut-out depth buffer and the cut-out slope s_c. For example if the viewing direction parallel to the z axis is considered, the z component of the gradient is constant and corresponds to the *slope* factor s_c. The x and y components are computed from the the distance vector to the footprint in the cut-out depth buffer. The distance vector is defined as a vector between the current pixel position in the cut-out depth buffer and the closest footprint pixel. The approach is analogous to the approach of Shin and Shin [70]. The difference in the rendering result is illustrated in Figure 4.14. The dataset

consists of two nested spheres which enclose an innermost cube. Figure 4.14 (a) illustrates shading with volumetric gradients. Figure 4.14 (b) illustrates shading with surface normals of the conical side-walls. Shading with the surface normals produces more pleasing results.



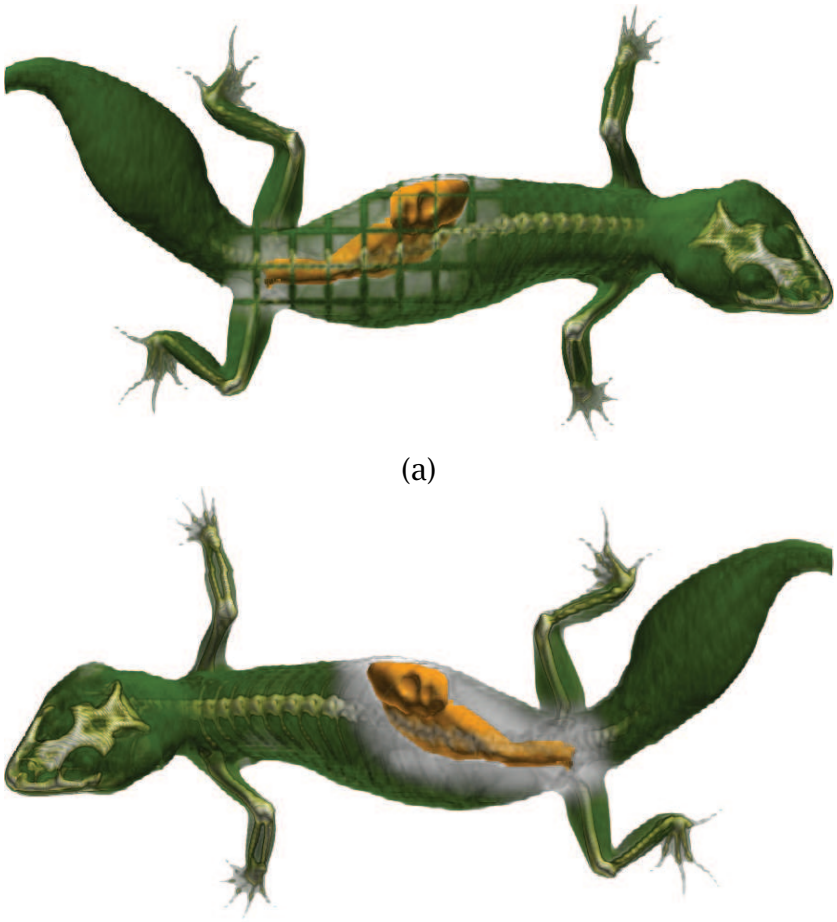
(*a*) (*b*)

Figure 4.14: *Sample points at the cut-out geometry shaded using (a) volumetric gradients and (b) surface normals of the conical side walls.*

4.5 Results

In the following we demonstrate importance-driven visualization on three real-world datasets. The *Leopard Gecko* dataset is of resolution $512 \times 512 \times 87$. The rendering results of this dataset have been already shown in the previous sections of this chapter. The *Monster Study* is of resolution $256 \times 256 \times 610$. Both datasets are using pre-segmented objects. The third dataset is a time-varying simulation of the hurricane Isabel. One time-step is of resolution $250 \times 250 \times 50$. This three-dimensional flow dataset features multiple simulated properties including cloud moisture, precipitation, pressure, and temperature. This dataset is not pre-segmented, only the position of the hurricane eye in object space is predefined. The simulation consists of 48 time steps.

Figure 4.15 (a) demonstrates combination of image-space AImC with screen-door transparency as levels of sparseness scheme. Figure 4.15 (b) demonstrates the combination of image-space AImC with color and opacity modulation.



(a) (b)

Figure 4.15: *Image-space AImC of the Leopard Gecko dataset: (a) screen-door transparency and (b) color and opacity modulation.*

Figure 4.16 shows the conical MImP of multiple abdomen organs of the Monster Study from different viewpoints. The liver, spleen, kidneys, and intestine have the same importance value. The tumor (in yellow), located between kidney and liver, has the highest importance value. The rest of the body is of lower importance than any of the mentioned objects. As the organs of the abdomen have the same importance they do not *cut away* each other. The highest importance value is assigned to the tumor and therefore everything in front of the tumor is cut away. MImP allows to visualize the most important information, i.e., the tumor, its shape and spatial position in relation to other objects. In contrast to traditional approaches the occlusion problem is solved automatically.

Another example of a conical MImP is shown in Figure 4.17. It shows 6

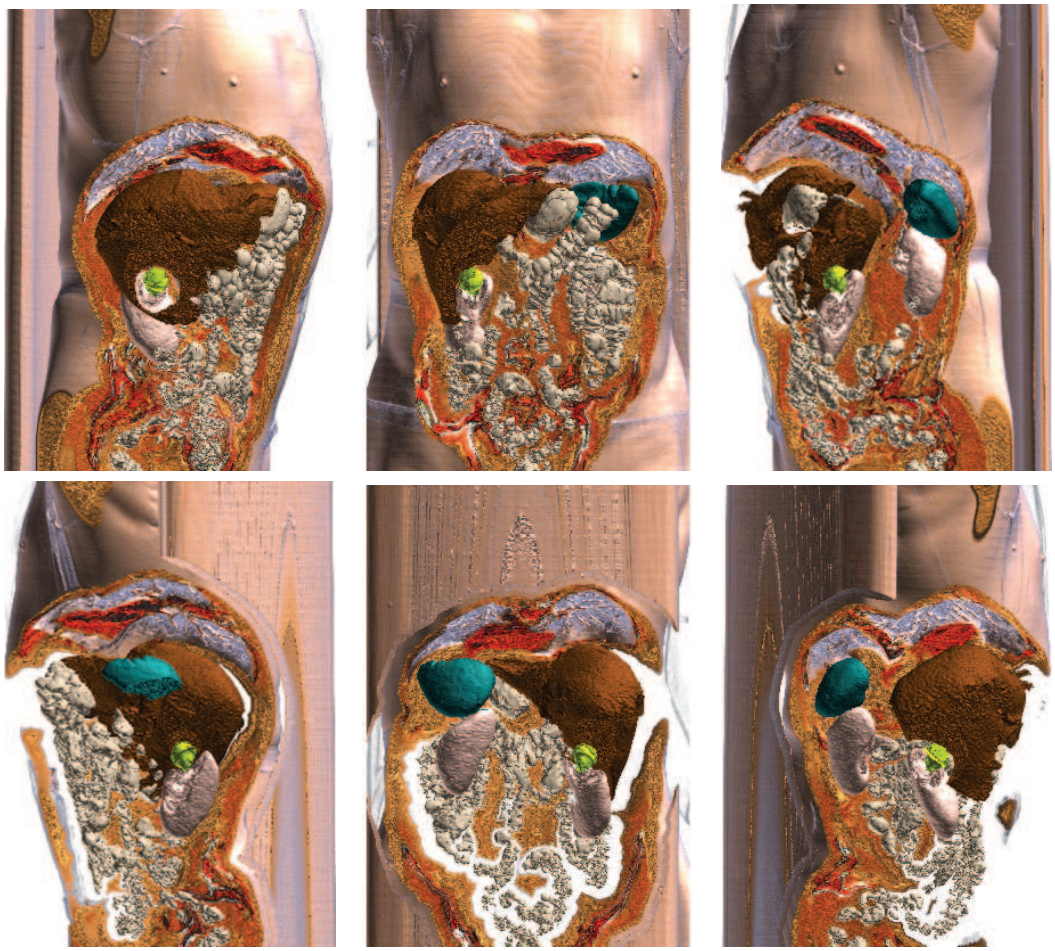


Figure 4.16: *The Monster Study dataset rendered using a conical MImP. The highest importance is assigned to the tumor (yellow object). The organs of the abdomen are assigned a medium importance. The rest of the dataset has the lowest importance.*

out of 48 time steps of the hurricane Isabel data. In this case a different way of feature classification was chosen. The important feature is the position of the eye of the hurricane. At the eye position a proxy cylinder is placed and everything inside the cylinder has higher importance than the rest of the data. The cylinder footprint is the basis for the cut-out geometry.

This example also shows how to combine multiple scalar volumes using importance-driven visualization. The focus object is defined as the

group of voxels inside the cylinder around the hurricane eye. Inside the cylinder the total precipitation mixing ratio is visualized. Due to the cut-away view it is possible to have a clear view at this property close to the eye of the hurricane. Outside the cylinder is the context area where the total cloud moisture is visualized. This time-dependent dataset also shows that the important feature can change its position and shape over time. Importance-driven visualization guarantees to emphasize the important feature irrespective of viewpoint and feature position and shape.

The prototype implementation has been integrated in the J-Vision medical workstation [34]. The implementation is written in java and is not using a performance-optimized ray-caster. Thus the performance is not interactive. The rendering takes approximately from 5 to 40 minutes per-frame depending on the screen resolution, data resolution, and importance-driven settings.

To achieve interactive rendering performance, importance-driven components have to be combined with a very fast CPU-based ray-caster system [23, 21] also partly integrated in the J-Vision workstation. Due to implementation constraints the integration of a real-time ray-caster has not been done yet. This is planned for first diagnostic prototypes based on importance-driven visualization in the near future. An interactive volume illustration system integrating importance-driven visualization has been proposed by Bruckner et al. [5, 4]. The interactive framerate is in this case achieved by using texture-mapping graphics hardware for volume rendering.

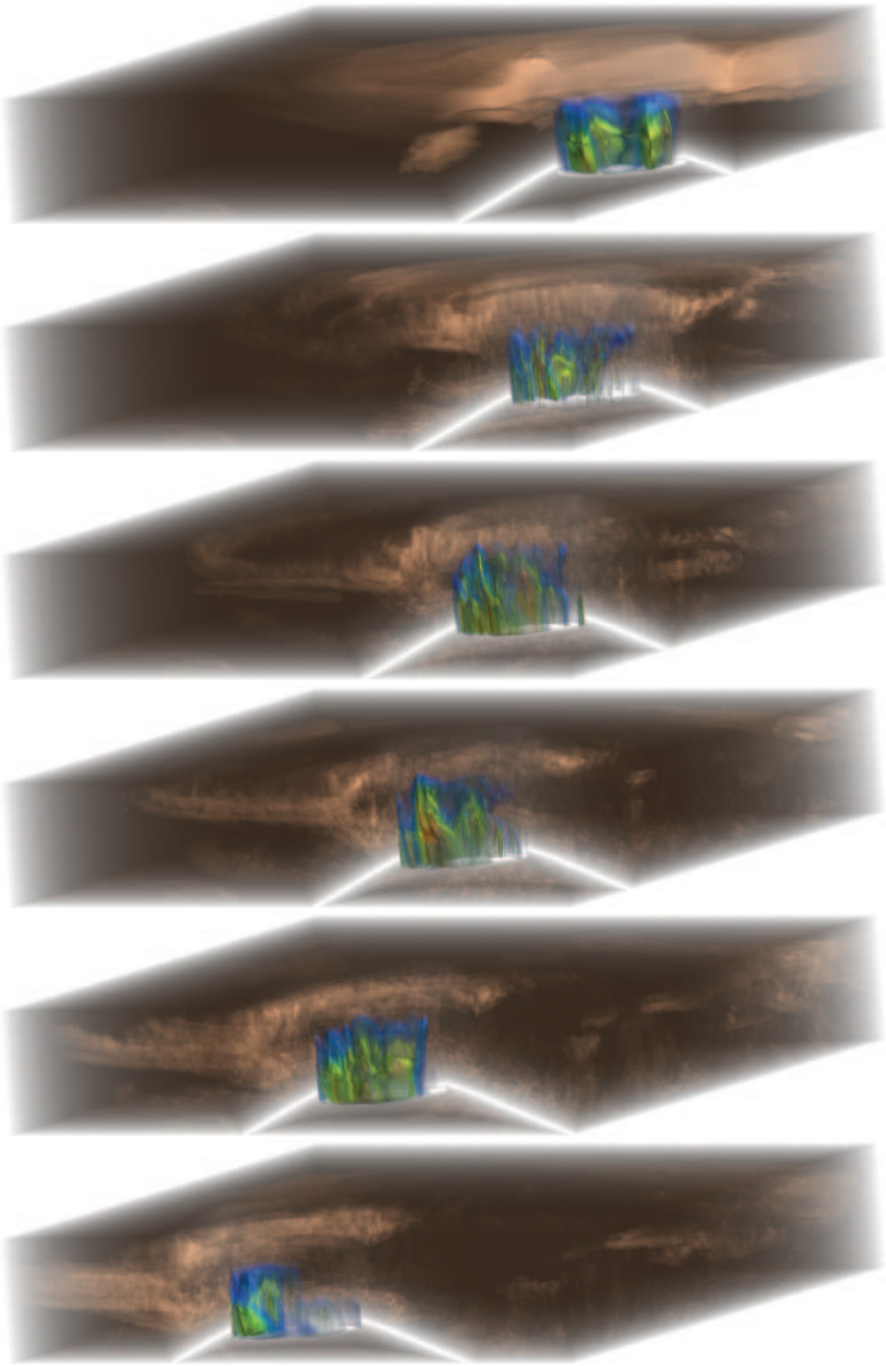


Figure 4.17: *Simulation data of hurricane Isabel rendered using a conical MImP. Two different properties are visualized: total cloud and total precipitation mixing ratios. The interesting feature is the precipitation close to the eye of the hurricane. The context feature is the cloud mixing ratio. Images are showing 6 out of 48 time steps.*

Chapter 5

Visibility Feasibility

In the previous chapter the model for importance-driven visualization has been presented. Two different importance compositing schemes have been proposed: maximum importance projection (MImP) as a compositing scheme for cut-away views and average importance compositing (AImC) for ghosted views. In the case of ghosted views the context information is not removed completely. The context is just suppressed to improve the visibility of the more prominent structures. As already discussed at the end of the Section 4.2 AImC preserves the information about the thickness of the context information in front of the focus, which can compromise the clear visibility of the focus object. The importance-driven visualization model as it was presented in the previous chapter does not evaluate the resulting visibility of the objects, and it does not incorporate explicit visibility-preserving importance compositing schemes.

In this chapter the AImC scheme, is extended by the evaluation of the object visibility. The motivation is to enable the validation whether the result image reflects the correspondence between the object visibility and the assigned importance value. In the following simple visibility evaluations of volumetric data are presented. The visibility evaluations are applied in visibility-preserving importance compositing (VPImC). This is as a variant of AImC for visibility-preserving ghosted views. Afterwards an iterative global VPImC approach is discussed.

5.1 Visibility Evaluation

Visibility of a volumetric object is defined as the amount of information present in the final rendered image. The most simple visibility estimation scheme evaluates the visibility according to the accumulated opacity values stored in the pixels. The visibility value is evaluated on a per-pixel

basis and it directly corresponds to the accumulated opacity values from volume rendering. In case multiple volumetric objects contribute to a pixel, the visibility distribution corresponds to the accumulated opacities of the respective objects. The exact evaluation of per-object visibilities is illustrated by an example in the text below.

To compute the object visibility, objects are first rendered using the local compositing step of two-level volume rendering [26] (see Section 3.3). The compositing scheme is using DVR compositing defined by Equations 3.2 and 3.3. For a single pixel the maximal opacity of a particular object is 1.0, which corresponds to the maximal per-pixel visibility. In the global compositing step the total visibility of a particular object along a ray is computed as the sum of all accumulated opacities computed in the local compositing step. This is explained on a simple example as shown in Figure 5.1. Accumulated colors and opacities computed by the local compositing step are illustrated in different colors of the ray segments in Figure 5.1 (b). For the given pixel, two accumulated opacities (in cyan and green) contribute to the visibility of the outer object o_0 and one accumulated opacity (in red) contributes to the visibility of the inner object o_1. Analogous to the computation of opacity as defined by Equation 3.3 we compute the per-pixel visibility of the inner object o_1 as:

$$\nu(o_1, r) = (1 - \alpha(o_0, r_0)) \cdot \alpha(o_1, r_1))$$

Here ν is a visibility function with object o_x and r_y as parameters. α is the accumulated opacity of a particular object o_x along a ray segment r_y. The per-pixel visibility of the outer object o_0 is computed as the sum of two contributing object parts:

$$\nu(o_0, r) = \alpha(o_0, r_0) + \alpha(o_0, r_2) \cdot (1 - (\nu(o_1, r) + \alpha(o_0, r_0)))$$

All per-pixel object visibilities are summed-up to determine the object visibility over the entire image.

The visibility computation abstracts from the type of visual representation of a volumetric object. An iso-surface for example shows very clearly a narrow region of the volumetric feature, but does not give any information about areas away from the iso-surface. With this representation visibility is just concentrated on the iso-surface region. On the other hand a ramp-like transfer function produces a semi-transparent representation where object locations more evenly contribute to the overall visibility. Vastly varying visual representations may thus provide the same overall visibility. In this visibility estimation scheme the visibility value does not directly reflect whether all information along the ray is

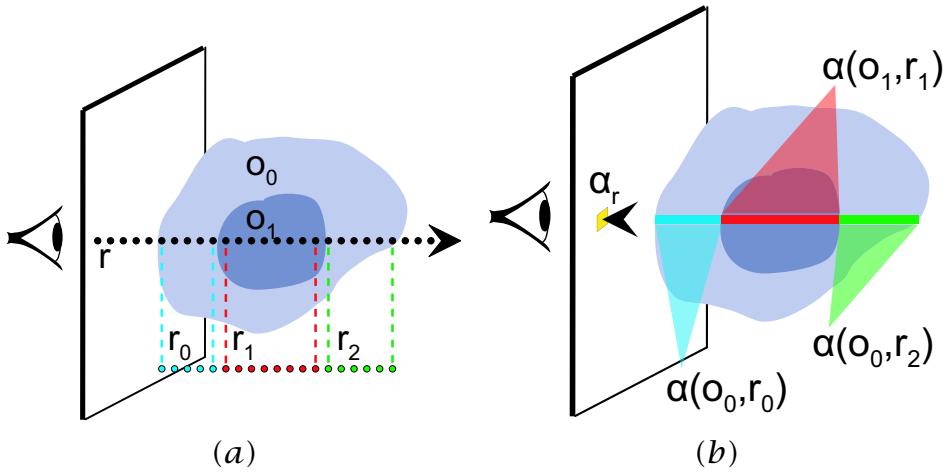


Figure 5.1: *Two-level volume rendering of two objects o_0 and o_1. Ray r is decomposed into three sub-rays (r_0, r_1, and r_2). Sub-rays are composited individually by a specified* local compositing method *(a). The final result is composited in the* global compositing step *(b), where the colors and opacities which result from the local compositing are combined together to produce the overall color and opacity α_r.*

visible. When during the ray traversal the opacity accumulation achieves the maximal value of 1.0, the ray traversal is stopped and the per-pixel visibility value of 1.0 is assigned. This simple visibility evaluation model is used to explain of the concept of visibility-preserving importance compositing. The visibility computation model could be replaced by other more elaborate models that incorporate for example perceptual metrics. A more elaborate visibility evaluation could take into account the following aspects:

» Number of contributing samples: In the visibility of a particular volumetric object all samples have to be considered. For those that are invisible the contribution should be equal to zero or even negative (as a penalty function). The number of visible samples should be positively reflected in the visibility estimation. If all samples are visible, it is possible to make an estimation about the object *thickness*.

» Shape and color cues: The visibility of a particular object should be weighted by the contained level of shape or color cues. For example, in the case of an iso-surface, the information about all samples *behind* the iso-surface is not present, but the samples on the surface

are very well presented and we have a clear information about the shape. Color information can be better conveyed by a lower number of contributing samples. For example along a particular ray there are samples of red, green, and blue color. The accumulated color is a gray value which does not represent the data content very effectively. None of the sample colors along the ray is represented in the resulting pixel color.

» Visibility fraction: In case of occluding features the visibility of an object shall be calculated with and without occluders. The visibility without occluders is the maximal possible visibility according to the current visual settings. The visibility with occlusions is a fraction thereof and can be set into perspective to this maximal visibility.

» Application dependency: The intended application area greatly influences the visibility metrics one should aim for. One application might benefit from the visualization of object thickness, where the shape cue is rather irrelevant. Another application might profit from visualizations where shape and color are emphasized. For different applications different visibility criteria apply, which will make a comparison between two visibility evaluations more difficult. Another possibility can be to consider visibility as a multi-dimensional property, instead of a scalar value. Each visibility *cue* will be represented as a separate dimension.

Visibility-preserving importance compositing uses the result of the visibility evaluation for further feature suppression or emphasis. If a high-importance feature would be represented in the final image only *sparsely*, i.e., its visibility value is low, a more *dense* representation is assigned to the feature. The goal is that the visibility of a feature is proportional to its importance. In the following section this concept is explained in more detail.

5.2 Visibility Preserving Visualization

Visibility preserving importance compositing (VPImC) guarantees constant visibility of the focus object. Independent from the thickness of the context in front of the focus, a constant fraction of visibility is reserved for the focus object. For example under some viewing angle the context object in front of the focus may be *thin* and the context can be represented more densely. Under a different viewing angle the context object may be *thicker* in front of the focus object. Therefore the con-

text should be more transparent. Thus the suppression of the context varies according to its *thickness* in regions that are occluding the focus object. The visibility of the focus remains constant. This is illustrated in Figure 5.2 on an example of cut-away views. Such a type of VPImC uses only a local suppression of the context object in front of the prominent feature. In the following two different visibility preserving visualizations are presented. First local modifications of visual representations for cut-away views are discussed. Second an iterative approach for global modifications of visual settings is described.

Local Visibility Modifications

Using local VPImC the initial visual abstractions (levels of sparseness from Chapter 4) are estimated in the same way as with AImC by using Equation 4.1. In AImC the levels of sparseness are selected for each sample during ray-traversal. With local VPImC an appropriate level of sparseness can only be selected *after* the ray-traversal stage using the results from the visibility evaluation. The level of sparseness is determined as follows: The level of sparseness of the context object in front of the focus object shall be S_o. For local VPImC the goal is to adjust the *average* of the accumulated opacity in the occluding region to be equal to the value S_o. Therefore the context part in front of the focus is rendered separately. All ray opacities of the occluding part are summed together to compute the average per-ray opacity α_{avg}. To preserve a constant visibility of the occluding region and thus also of the focus object the average per-ray opacity has to be equal to S_o. Therefore for each ray within the occluding region, the accumulated opacity value is corrected. This is expressed by Equation 5.1:

$$\alpha'_{acc}(x) = \alpha_{acc}(x)\frac{S_o}{\alpha_{avg}} \tag{5.1}$$

where α'_{acc} is the modified accumulated opacity of the suppressed context part, α_{acc} is the original accumulated opacity value, S_o is the level of sparseness value of the context object and α_{avg} is the average accumulated opacity value of the entire suppressed context part.

The separate rendering of the occluding part is done using the 2lVR approach. Every object or a part thereof (as in the case of the context object) is rendered separately in a *local compositing step*. Then the *visibility correction* is done for the occluding part of the context. Finally a combination the occluding context part, the focus object, and the non-occluding object part takes place in the *global compositing step*.

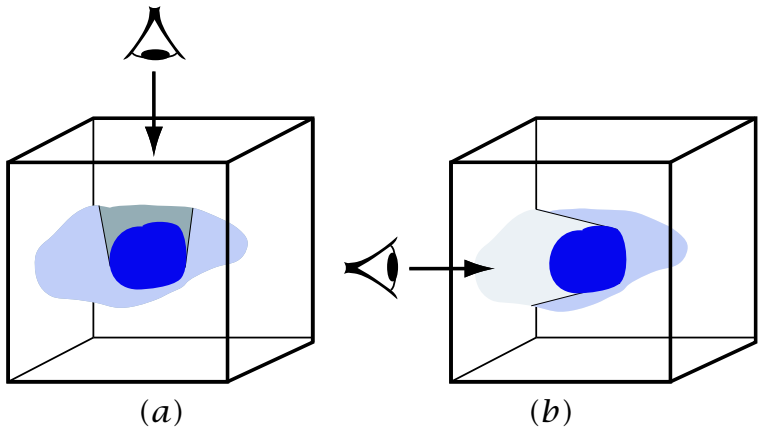


Figure 5.2: *Principle of local visibility preserving importance compositing. Constant visibility allows a denser thin context (a) and requires a sparser thick context (b).*

Figure 5.3 shows the results of VPImC compositing under different viewpoint settings. It offers a comparison to AImC compositing as shown in Figures 4.6 and 4.7 described in Section 4.2. Especially in Figures 4.6 (b) and 4.7 (b) a large occluding context region considerably reduces the visibility of the focus object. This is not the case in Figure 5.3 (b).

Global Visibility Modifications

In the case of importance compositing with global visibility preservation, transfer functions are adjusted so that the overall visibility corresponds to the predefined importance distribution. The framework for a global visibility preserving importance compositing (global VPImC) is given in Figure 5.4. The framework starts out with the importance specification as before. An initial transfer function is specified automatically or user-steered according to the properties of the data. The visibility of the objects is estimated according to the visual representations and viewport settings. If the visibility distribution is not proportional to the importance distribution, the visual settings are adjusted automatically according to the calculated visibility information. This adjustment is repeated iteratively until the desired visibility distribution is achieved.

The feedback from the visibility computation modifies the visual mapping. As the visibility is computed from the opacity values, only the opacity channel of the transfer function is modified and the color chan-

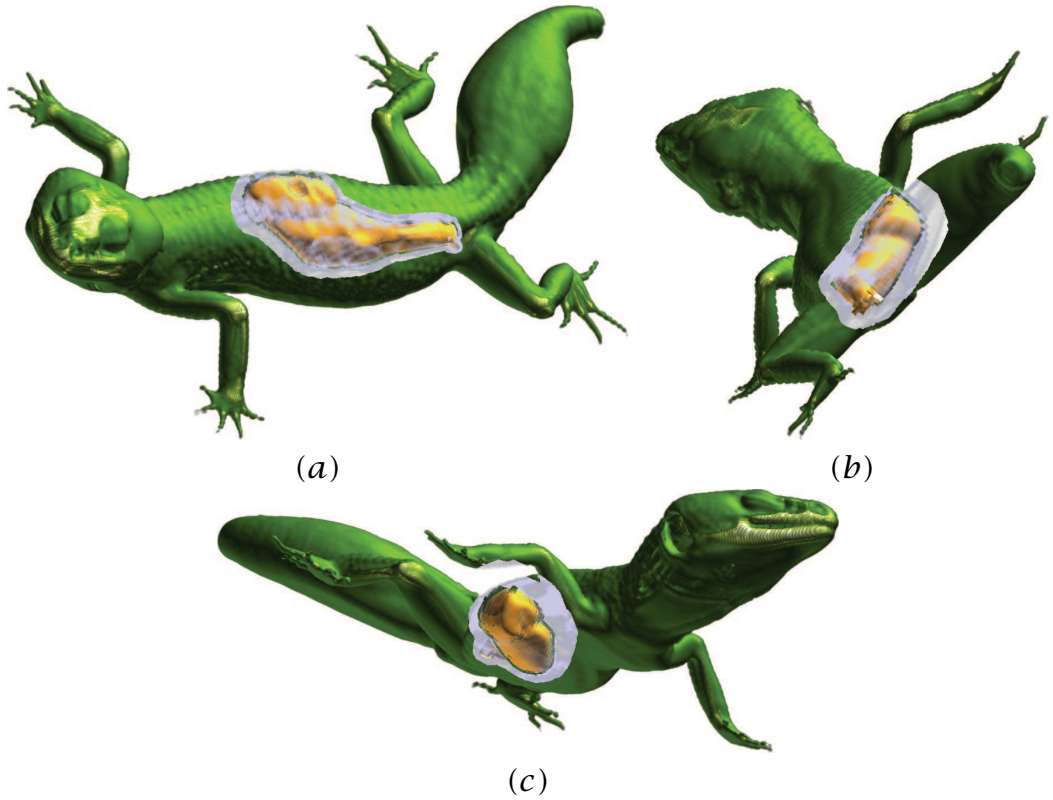


Figure 5.3: *Local visibility preserving importance compositing (VPImC). The dataset is shown under different viewpoint settings (a), (b), and (c) with constant visibility of the focus object. Focus visibility is independent from the thickness of the occluding context part.*

nels remain unchanged. The opacity channel of the transfer function is adjusted until the *desired* visibility is achieved. For example if the volumetric data consists of two objects, where object 1 has importance 1.0 and object 2 has importance of 0.5, the overall visibility distribution has to be split between these objects with a ratio of two thirds to one third.

Figure 5.5 shows three different rendering results of the Leopard Gecko dataset. Figure 5.5 (a) is a rendering with the initial transfer function setting. Figure 5.5 (b) shows the result of a transfer function adjustment where both objects, i.e., the gecko body and the inner object are *equally* visible. This means that for both objects their sums of per-pixel visibilities are equal. This naturally causes a rather strong suppression of objects that take-up much more screen space as compared to smaller

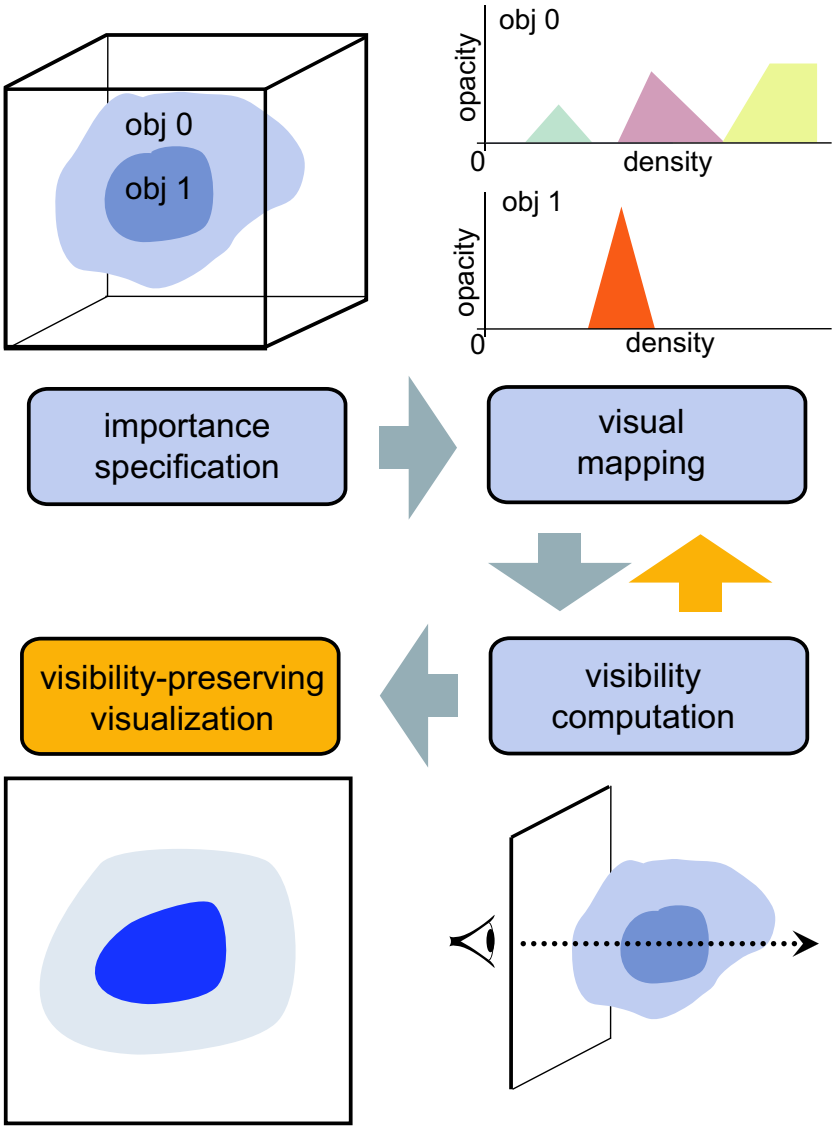


Figure 5.4: *Framework of global visibility-preserving importance compositing (global VPImC).*

structures. Alternatively a normalization of the object visibility can be performed. The projected area, i.e., the number of pixels that contribute to the visibility of the given object, is the normalizing factor. The rendering result of the gecko taking into account this *normalized* visibility is shown in Figure 5.5 (c).

The global VPImC approach as discussed above tries to match the relative visibilities to the importance distribution (Equation 5.2). There are many transfer function settings produce the desired relative visibili-

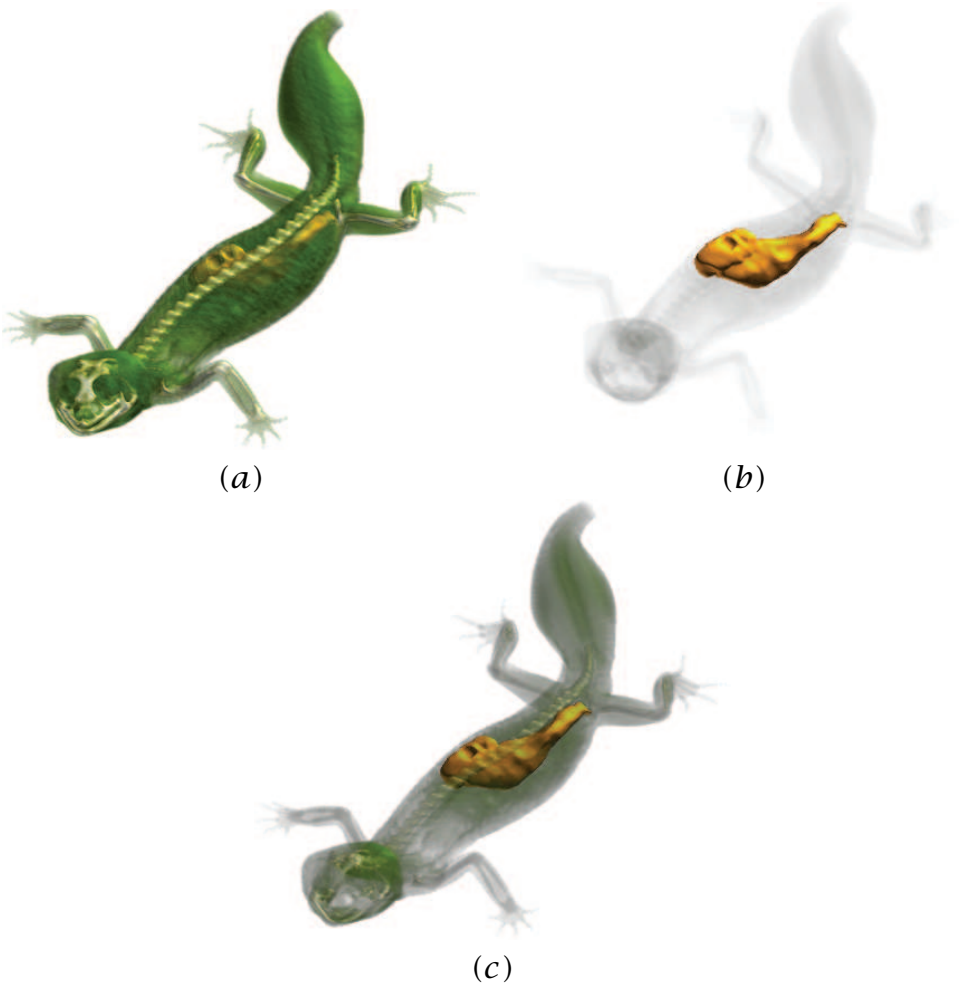


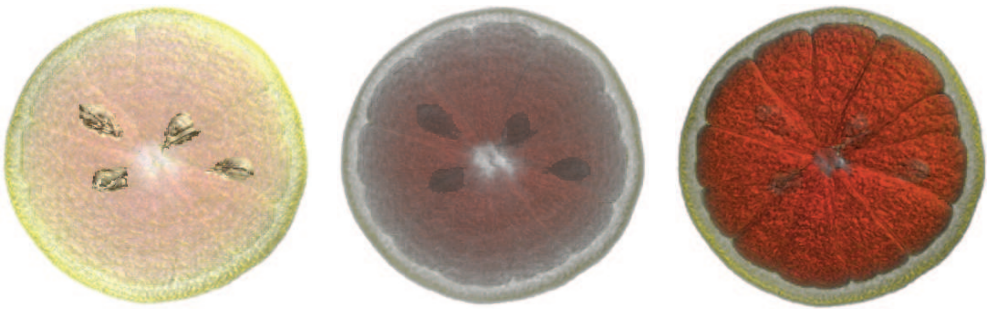
Figure 5.5: *Global visibility preserving importance compositing. The inner organ and the body are equally important. Images are showing results with (a) the initial transfer function, (b) balanced and unnormalized visibility, and (c) balanced visibility normalized to the projection area of respective objects.*

ties. Some settings for example produce a sparse display of the objects although the desired relative visibilities are achieved. For an expressive visualization among those settings that have the correct relative visibilities, the configurations with a higher sum of absolute visibilities are preferable. Figure 5.6 shows the dataset of an orange with three different objects: rind, pulp, and seeds where corresponding importance values are 1.0, 2.0, and 0.5. First the seeds are very prominent, while the visi-

bility of the most important object, i.e., the pulp is rather *sparse*. After several iterative visibility adjustments, the desired relative object visibility for each object is achieved. In every iteration the transfer functions of the individual objects are either scaled-up or scaled-down depending on their relative visibility to be either too low or too high. The result of the iterative search is a transfer function where the visibility distribution corresponds to the importance distribution. The ratio between importance values and object visibilities can be expressed by the following Equation 5.2:

$$\frac{V_o}{\sum_{x=0}^{n} V_x} = \frac{I_o}{\sum_{x=0}^{n} I_x} \pm \epsilon \tag{5.2}$$

Here function V denotes visibility, I denotes importance and ϵ is a given permissible deviation between the distributions. In order to find a transfer function setting with a higher sum of absolute visibilities, the opacity transfer functions are uniformly scaled-up for each object. Figure 5.6 (a) shows an image with the initial visual settings. Figure 5.6 (b) shows an image where the desired visibility distribution has been reached. Figure 5.6 (c) shows another image with the desired relative visibilities but with a bigger sum of absolute visibilities as compared to Figure 5.6 (b).



(*a*) (*b*) (*c*)

Figure 5.6: *Global VPImC: The orange dataset has three pre-segmented objects, i.e., rind, pulp, and seeds. The importance of the objects is* 1.0, 2.0, *and* 0.5. *(a) image with the initial visual settings, (b) image with desired relative visibilities, and (c) image with desired relative visibilities where the sum of absolute visibilities is much higher than in (b).*

Chapter 6

Applications

This chapter describes some approaches where the importance-driven concept can be effectively applied to increase the visualization quality. First, two-potential areas for medical diagnosis are presented. Both applications are targeting to improve the early stage diagnosis of cancer. The first application deals with visualization of small nodules within the lungs which indicate the early stage of the lung cancer. The second application is the visualization of breast cancer detected via dynamic contrast-enhanced magnetic resonance imaging. The last application is VolumeShop, an interactive tool for creating illustrations directly from the volumetric data. It integrates various different features for interactive illustrative visualization including importance-driven visualization. To fully appreciate the strengths of importance-driven visualization viewpoint changes or dynamic scenes are essential. Further examples of potential applications are shown in video sequences available at http://www.cg.tuwien.ac.at/research/vis/adapt/2004_idvr/.

6.1 Lung Nodules Visualization

This section sketches an application for automatic cut-away visualization of lung tumors given as nodules. The goal is to improve the diagnosis of early stage lung cancer and to increase the probability of patient survival. The automatic diagnostic process is divided into two parts. The first part is the detection stage where the data is classified according to the probability of a tumor presence. The second part is the visualization of these suspicious regions. The medical expert is guided by the application to the regions where with high probability lung nodules are present. Thus the time-consuming process of slice-by-slice inspection is replaced by an automatic process. The medical expert has to finally validate which regions are considered as nodules and which regions are false positives.

A slice-by-slide inspection also has the drawback that spherical nodules as well as tubular vessels show up as circles in the slices.

The initial implementation of the detection part is based on local shape properties of a sample. The detection algorithm is in spirit of the method proposed by Sato [68] (see Section 3.1). Local shape properties are derived from the Hessian matrix which consists of the second-order partial derivatives [27]. The eigenvalues of the Hessian matrix are used to determine the probability of a nodule-like shape. Results show that this method is capable of detecting nodules of a size as low as 3 mm in diameter from a high-resolution thorax CT. The tests also show a high probability in detecting real nodules with a reasonable number of false positives.

The visualization is based on importance-driven feature enhancement where the probability of lung nodule presence is mapped to importance values. Volume parts that occlude suspicious areas will be represented sparsely. The suspicious regions shall be clearly visible. The 3D view is linked to a slice viewer for coronal, saggital, and axial slices. The 3D view quickly allows the medical expert to focus on suspicious regions. He can then validate true positives through slice-by-slice inspection.

The algorithm has been also tested on thorax CT data sets featuring small nodules. The dataset was a high-resolution thorax CT with a resolution of $512 \times 512 \times 198$ and voxel spacing of $0.68 \times 0.68 \times 2.0$ mm. The detection process took approximately 5 minutes. Three suspicious regions have been selected. The visualization of suspicious regions is shown in a series of images, where the suspicious regions are visualized in red color enclosed by a green halo. The series of images from different viewing positions are shown in Figure 6.1. Visualizations show the spatial position of the most interesting regions irrespective from the viewpoint settings. This increases the visualization expressivity as compared to traditional visualization methods (e.g., direct volume rendering or slicing).

6.2 Breast Cancer Visualization

Computer-aided diagnosis of the breast cancer is an intensively researched area in medical imaging. There are several techniques to perform the diagnosis prior to surgical intervention. The state-of-the-art techniques are based on three-dimensional mammograms using the magnetic resonance imaging (MRI) modality. However, also with the high-precision MRI scanners it is difficult to differentiate cancer tissue from

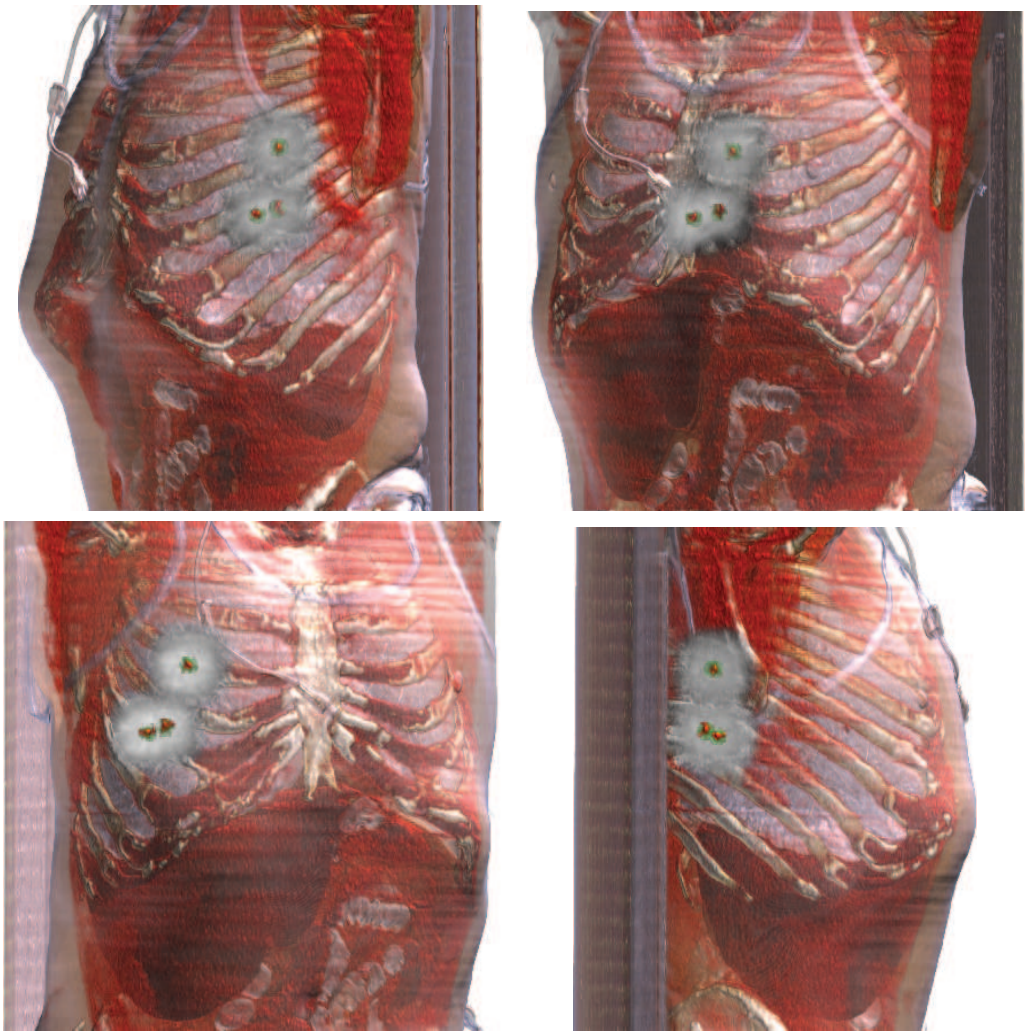


Figure 6.1: *Series of three-dimensional importance-driven visualizations of suspicious regions (possibly early stage lung cancer nodules) within a thorax CT data. The suspicious regions are highlighted by red color enclosed by a green halo.*

healthy parts. One of characteristics of the cancer tissue is that it increases the blood flow as vascular structures in its vicinity are well developed. This can be detected by introducing contrast medium to the patient's body before or during the scanning process. Current scanning technology allows to produce multiple three-dimensional scans in a relatively short time. Such medical data acquisition is called dynamic

contrast-enhanced MRI (DCE-MRI). From a time-series of DCE-MRI scans the medical doctor can inspect the characteristics of the contrast enhanced blood distribution within the breast tissue. This allows to determine the presence of the cancer and it is even possible to estimate with a high probability the type of the lesion, i.e., malignant or benign.

The mammograms are mostly inspected by viewing two-dimensional slices for diagnosis. There are several additional inspection techniques for advanced diagnosis [8], but they are always used in combination with analysis of the 2D slices. The problem with 2D slicing is that they do not effectively convey three-dimensional feature location. The diagnosis thus requires an experienced clinician that is able to build the mental map of spatial characteristics just from the 2D slices. Inspection of 2D slices is not very suitable for the planning of a surgical treatment. It does not provide enough information about the spatial location of the cancer tissue. Applying advanced volume rendering techniques such as direct volume rendering in combination with contours, eases this problem only to a certain degree. Breast tissue, similar to the structures within the lung, is very complex and DVR often results in images where structures are occluding each other. It is difficult to guarantee a clear view on the inspected area.

Importance-driven visualization can be used to enhance the visibility of suspicious regions and their vicinity. This is illustrated in Figure 6.2. Images visualize two different DCE-MRI scans with pre-segmented tumor findings. The spatial location of the tumor is shown from different viewpoints to better convey the information about the exact spatial position of the finding. An additional visual cue to perceive the distance of the tumor from the breast surface is the diameter of the cut-out as the cut-out slope is taken to be constant.

6.3 Interactive Importance-Driven Visualization

The first implementation of importance-driven visualization has been intended as a proof of concept and is not interactive. However, direct interaction with the data is very important and often crucial to achieve a particular task using visualization. Hand crafted visualizations generated by an illustrator are often superior to computer-generated visualizations. But an artwork usually does not provide direct interaction with the visualized data. Thus, interactivity is the big advantage of computer generated illustrative visualization as compared to hand-crafted illustration.

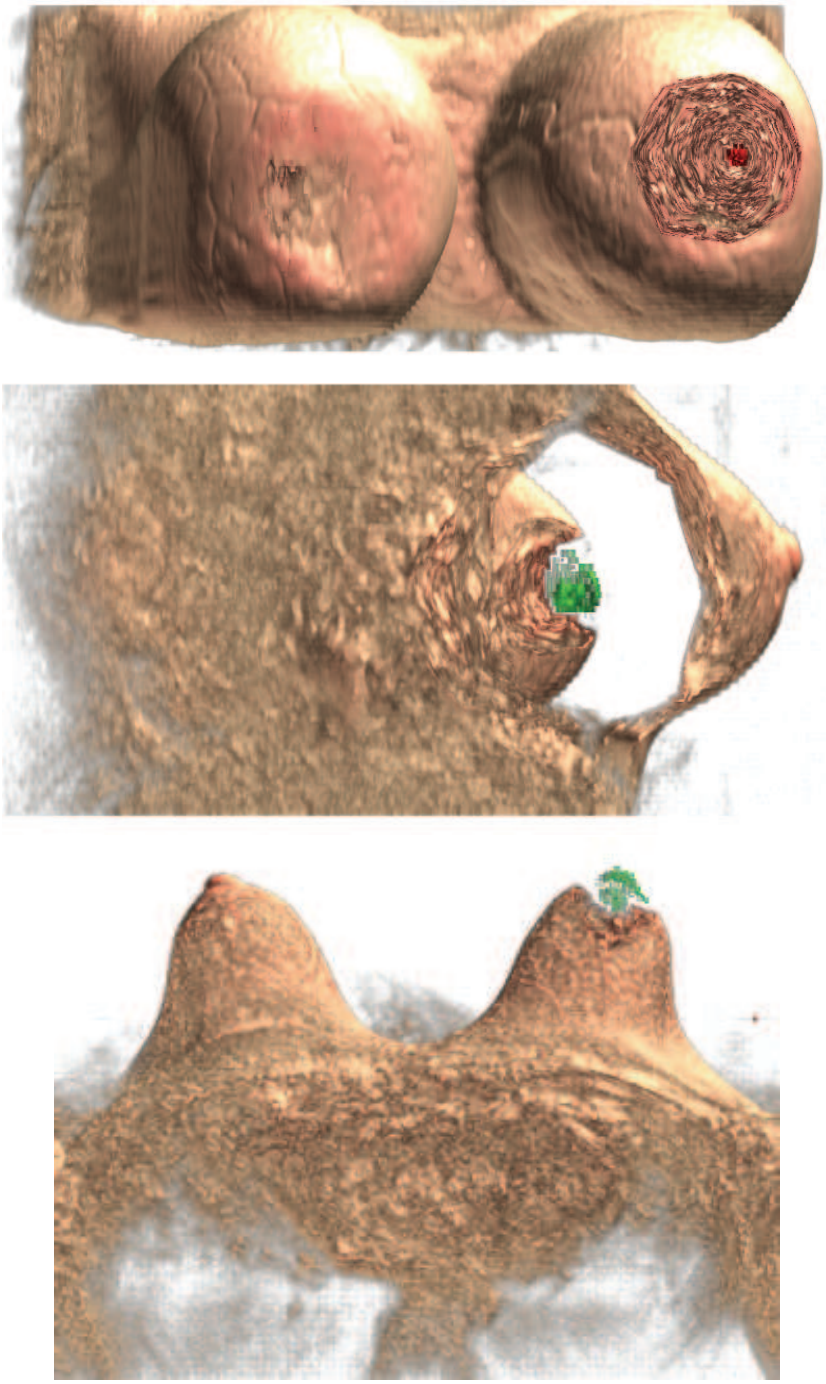


Figure 6.2: *Importance-driven visualization applied to a single time-step of DCE-MRI mammography scans with high probability of breast cancer. The important feature is the suspicious region, the rest is provided as context information.*

Importance-driven visualization has been recently integrated into the VolumeShop [4, 5] tool. VolumeShop is an interactive system which features advanced manipulation techniques and illustrative rendering techniques to generate interactive illustrations directly from the volumetric data. The system is using latest-generation texture-mapping hardware to perform interactive rendering using various kinds of rendering styles. It implements a multi-volume concept to enable individual manipulations of each volume part. The segmentation of the volumetric objects can be done directly via 3D painting, as discussed in Section 3.1. Importance-driven cut-away visualization turned-out to be very useful for fine-tuning the segmentation specification. Features that occlude the brushed feature are cut away and the brush stroke does not interact with the occluding structures as they are *invisible*. From a particular view it is then possible to brush or erase specific regions that are not easy to access otherwise.

Apart from importance-driven visualization resulting into cut-away and ghosted views, VolumeShop features a label management to introduce basic descriptions for the visualized data. To focus at a particular feature, this feature can be moved from its original spatial position. To indicate its original spatial position it is possible to display a *ghost* there, or add additional markers such as fanning or arrows.

Some VolumeShop features are indicated in Figures 6.3 and 6.4. The image series of the engine block dataset shown in Figure 6.3 are illustrating different levels of sparseness for the ghost region in front of the selected feature. The feature is additionally shown dislocated from its original position to enhance the focus. Figure 6.4 shows visualizations of other volumetric datasets featuring illustrative techniques to increase expressivity.

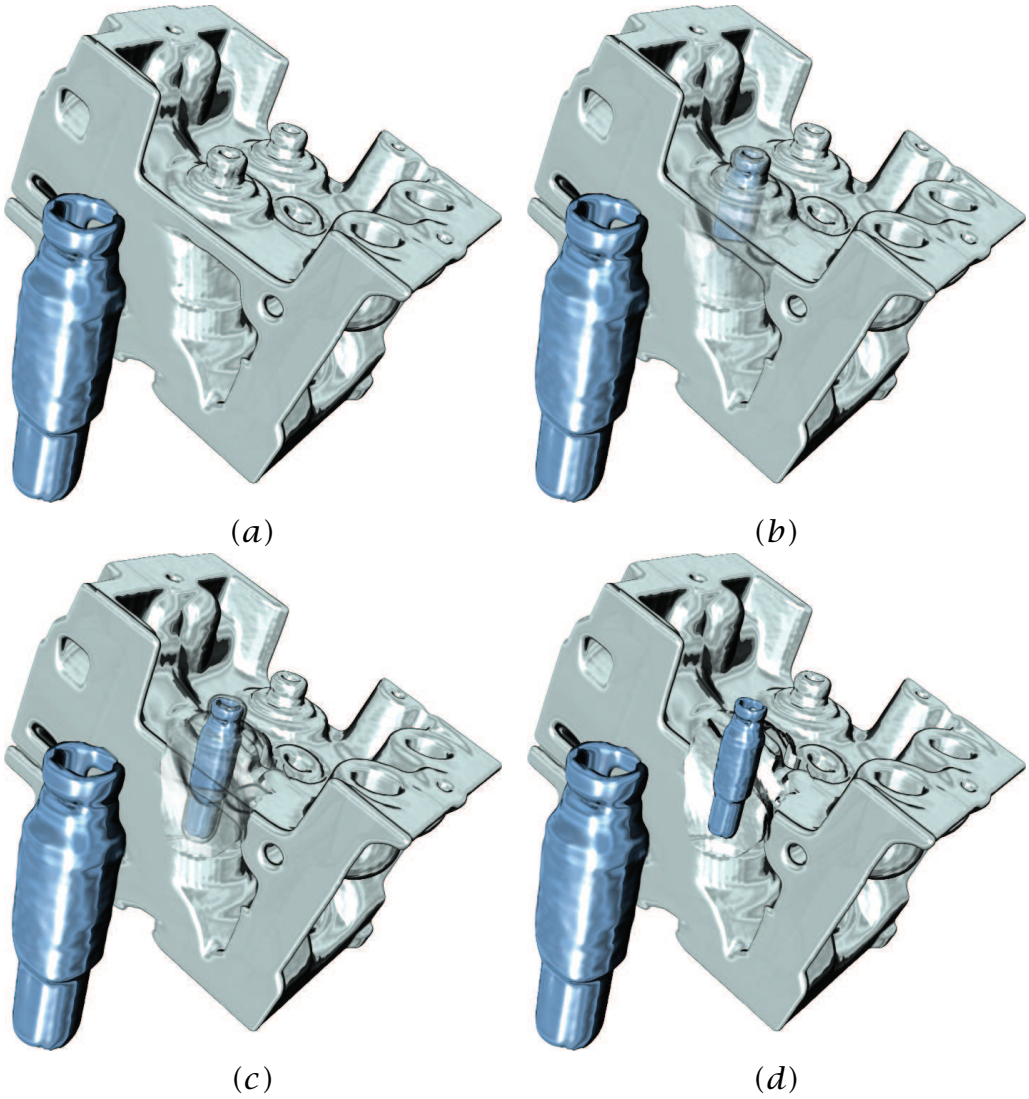


Figure 6.3: *Interactive importance-driven visualization of the engine block dataset. Series of images illustrate different levels of sparseness for the occluding engine region from the (a) most dense to (d) the most sparse one [4, 5].*

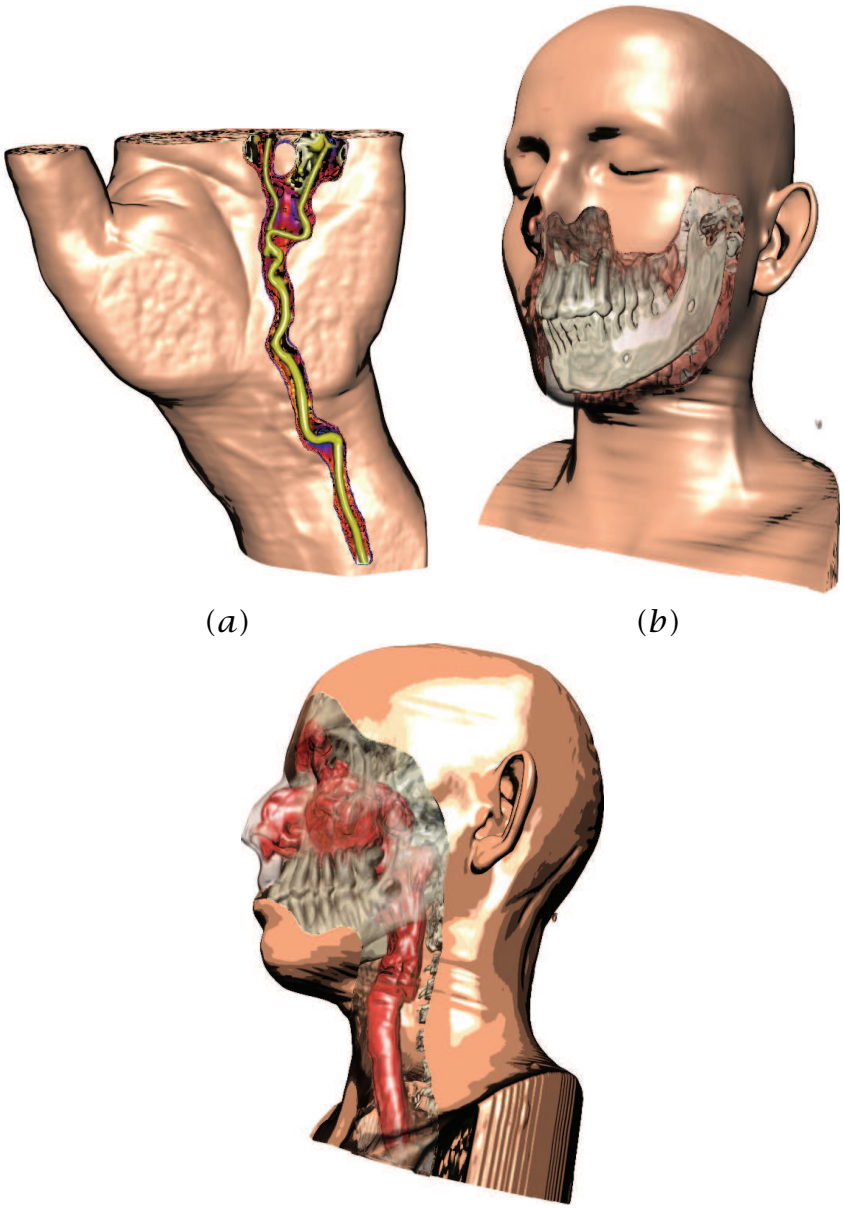


Figure 6.4: *Interactive importance-driven visualization of a CT dataset of the human head and hand. Images are rendered using non-photorealistic illustrative techniques. Cut-outs uncover (a) artery in the hand dataset, and (b) the jaw, and (c) the trachea in the head dataset [4, 5].*

Chapter 7

Summary and Conclusions

In this thesis importance-driven visualization as a viewpoint and feature dependent approach for automatic focus+context volume visualization has been presented. A new concept the importance dimension, is introduced to the traditional volume rendering pipeline. According to the importance and viewpoint settings each object is rendered in order to maximize the visual information in the result image. This method allows to see prominent structures within the volume as densely as possible. A sparse representation is chosen if other, more important structures are occluded. We have presented a model for importance-driven visualization consisting of *importance classification*, *importance compositing*, and *levels of sparseness*.

Importance compositing defines how the occluding context information should be visualized. It selects an appropriate level of sparseness for each feature. This additional stage in the volume rendering pipeline can either completely remove less prominent features if they are occluding more important features or it can suppress their visual representation. This can be done either globally, so the whole feature is removed or suppressed, or locally to create effects like cut-away or ghosted views known from traditional illustrations. Importance compositing can also perform validation of feature visibility. This guarantees a close correspondence between achieved visibility and the predefined importance.

The thesis also discusses several schemes for levels of sparseness. Levels of sparseness control the optical properties of the amount of visible elements in the volume data. Smooth opacity changes work well in combination with desaturation. The visible volume elements can be distributed uniformly over the whole volume, or the first- and second-order derivatives can be used for visibility distribution. Levels of sparseness specify transitions of data representations from the most dense to the most sparse ones. Another approach defines levels of sparseness

through combining different rendering techniques.

The thesis also shows the applicability of importance-driven visualization in medical visualization and diagnosis. Two potential diagnostic areas dealing with early cancer detection have been presented. The importance-driven visualization provides in both cases better visibility of suspicious regions as compared to traditional volume rendering. Additionally importance-driven visualization has been integrated into the VolumeShop, i.e., an interactive illustration tool. Also in this application importance-driven visualization turned-out to provide expressive images.

Bibliography

[1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing effective step-by-step assembly instructions. In *Proceedings of ACM SIGGRAPH'03*, pages 828–837, 2003.

[2] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986.

[3] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving volume rendering. In *Proceedings of EuroVis'05 (to appear)*, 2005.

[4] S. Bruckner and M. E. Gröller. VolumeShop: An interactive system for direct volume illustration. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2005.

[5] S. Bruckner, I. Viola, and M. E. Gröller. VolumeShop: Interactive direct volume illustration. SIGGRAPH 2005 Sketch.

[6] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization : Using Vision to Think*. Academic Press, 1999.

[7] S. Carpendale, D. Cowperthwaite, and F. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings of IEEE Symposium on Information Visualization'96*, pages 46–53, 1996.

[8] E. Coto, S. Grimm, S. Bruckner, M. E. Gröller, A. Kanitsar, and O. Rodriguez. Mammoexplorer: An advanced cad application for breast dce-mri. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2005.

[9] B. Csébfalvi and E. Gröller. Interactive volume rendering based on a "bubble model". In *Proceedings of Graphics interface 2001*, pages 209–216, 2001.

[10] B. Csébfalvi, L. Mroz, H. Hauser, A. König, and M. E. Gröller. Fast visualization of object contours by non-photorealistic volume rendering. In *Procceedings of Eurographics'01*, pages 452–460, 2001.

[11] D. DeCarlo, A. Finkelstein, and S. Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *Symposium on Non-Photorealistic Animation and Rendering'04*, pages 15–24, 2004.

[12] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. In *Proceedings of ACM SIGGRAPH'03*, pages 848–855, 2003.

[13] J. Diepstraten, D. Weiskopf, and T. Ertl. Transparency in interactive technical illustrations. In *Procceedings of Eurographics'02*, pages 317–326, 2002.

[14] J. Diepstraten, D. Weiskopf, and T. Ertl. Interactive cutaway illustrations. In *Procceedings of Eurographics'03*, pages 523–532, 2003.

[15] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation. In *Proceedings of VisSym'03*, pages 239–248, 2003.

[16] S. Feiner and D. Seligmann. Cutaways and ghosting: Satisfying visibility constraints in dynamic 3D illustrations. *Visual Computer: International Journal of Computer Graphics*, 8(5-6):292–302, 1992.

[17] A. Fuhrmann and E. Gröller. Real-time techniques for 3D flow visualization. In *Proceedings of IEEE Visualization'98*, pages 305–312, 1998.

[18] F. Giesecke, A. Mitchell, H. Spencer, I. Hill, J. Dygdon, and J. Novak. *Technical Drawing*. Prentice Hall, 2002.

[19] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of ACM SIGGRAPH'98*, pages 447–452, 1998.

[20] B. Gooch and A. Gooch. *Non-Photorealistic Rendering*. AK Peters, Ltd., 2001.

[21] S. Grimm. *Real-Time Mono- and Multi-Volume Rendering of Large Medical Datasets on Standard PC Hardware*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2005.

[22] S. Grimm, S. Bruckner, A. Kanitsar, and E. Gröller. Flexible direct multi-volume rendering in interactive scenes. In *Proceedings of Vision, Modeling, and Visualization'04*, pages 379–386, 2004.

[23] S. Grimm, S. Bruckner, A. Kanitsar, and M. E. Gröller. Memory efficient acceleration structures and techniques for CPU-based volume raycasting of large data. In *Proceedings of IEEE/SIGGRAPH Symposium on Volume Visualization*, pages 1–8, 2004.

[24] S. Gumhold. Maximum entropy light source placement. In *Proceedings of IEEE Visualization'02*, pages 275–282, 2002.

[25] H. Hauser and M. Mlejnek. Interactive volume visualization of complex flow semantics. In *Proceedings of Vision, Modeling, and Visualization'03*, pages 191–198, 2003.

[26] H. Hauser, L. Mroz, G. I. Bischi, and M. E. Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, 2001.

[27] J. Hladůvka. *Derivatives and Eigensystems for Volume-Data Analysis and Visualization*. PhD thesis, Vienna University of Technology, Austria, 2001.

[28] J. Hladůvka, A. König, and E. Gröller. Curvature-based transfer functions for direct volume rendering. In *Proceedings of SCCG'00*, pages 58–65, 2000.

[29] E. Hodges, editor. *The Guild Handbook of Scientific Illustration*. Wiley, 2003.

[30] Howell MediGraphics, http://www.medigraphics.com/, 2005.

[31] K. Hulsey technical illustration, http://www.khulsey.com/, 2005.

[32] V. Interrante, H. Fuchs, and S. Pizer. Illustrating transparent surfaces with curvature-directed strokes. In *Proceedings of IEEE Visualization'96*, pages 211–218, 1996.

[33] S. Islam, S. Dipankar, D. Silver, and M. Chen. Spatial and temporal splitting of scalar fields in volume graphics. In *Proceedings of IEEE/SIGGRAPH Symposium on Volume Visualization'04*, pages pp. 87–94, 2004.

[34] J-Vision medical workstation, http://www.tiani.com/, 2005.

[35] A. Kanitsar. *Curved Planar Reformation for Vessel Visualization*. PhD thesis, Vienna University of Technology, Austria, 2004.

[36] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–323, 1987.

[37] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Symposium on Volume Visualization'98*, pages 79–86, 1998.

[38] G. Kindlmann, R. Whitaker, T. Tasdizen, and Torsten Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization'03*, pages 513–520, 2003.

[39] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

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

[41] A. König, H. Doleisch, and M. E. Gröller. Multiple views and magic mirrors - fMRI visualization of the human brain. In *Proceedings of SCCG'99*, pages 130–139, 1999.

[42] R. Kosara, S. Miksch, and H. Hauser. Semantic depth of field. In *Proceedings of IEEE Symposium on Information Visualization'01*, pages 97–104, 2001.

[43] A. Krüger, C. Tietjen, J. Hintze, B. Preim, I. Hertel, and G. Strauß. Interactive visualization for neck dissection planning. In *Proceedings of EuroVis'05 (to appear)*, 2005.

[44] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of ACM SIGGRAPH'94*, pages 451–458, 1994.

[45] A. Lake, C. Marshall, M. Harris, and M. Blackstein. Stylized rendering techniques for scalable real-time 3D animation. In *Symposium on Non-Photorealistic Animation and Rendering'00*, pages 13–20, 2000.

[46] Lascaux cave, http://www.culture.gouv.fr/culture/arcnat/lascaux/, 2005.

[47] C. H. Lee, X. Hao, and A. Varshney. Light collages: Lighting design for effective visualization. In *Proceedings of IEEE Visualization'04*, pages 281–288, 2004.

[48] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[49] M. Levoy and R. Whitaker. Gaze-directed volume rendering. In *Proceedings of Symposium on Interactive 3D Graphics'90*, pages 217–223, 1990.

[50] W. Li, M. Agrawala, and D. Salesin. Interactive image-based exploded view diagrams. In *Proceedings of Graphics Interface'04*, pages 203–212, 2004.

[51] H. Löffelmann, L. Mroz, and E. Gröller. Hierarchical streamarrows for the visualization of dynamical systems. In *Eurographics Workshop on Visualization in Scientific Computing'97*, pages 155–164, 1997.

[52] A. Lu, C. Morris, J. Taylor, D. Ebert, C. Hansen, P. Rheingans, and M. Hartner. Illustrative interactive stipple rendering. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):127–138, 2003.

[53] E. Lum and K.-L. Ma. Hardware-accelerated parallel non-photorealistic volume rendering. In *Symposium on Non-Photorealistic Animation and Rendering'02*, pages 67–74, 2002.

[54] E. Lum, A. Stompel, and K.-L. Ma. Using motion to illustrate static 3d shape - kinetic visualization. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):115–126, 2003.

[55] K.-L. Ma and V. Interrante. Extracting feature lines from 3d unstructured grids. In *Proceedings of IEEE Visualization'97*, pages 285–292, 1997.

[56] S. Mann. *Intelligent Image Processing*. Wiley, 2001.

[57] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[58] M. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of IEEE Visualization'03*, pages 401–408, 2003.

[59] L. Mroz, H. Hauser, and M. E. Gröller. Interactive high-quality maximum intensity projection. In *Proceedings of Eurographics'00*, pages 341–350, 2000.

[60] S. Omar. *Ibn al-Haytham's Optics: A Study of the Origins of Experimental Science*. Bibliotheca Islamica, 1977.

[61] S. Osher and J. Sethian. Fronts propogating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1981.

[62] S. Owada, F. Nielsen, M. Okabe, and T. Igarashi. Volumetric illustration: Designing 3D models with internal textures. In *Proceedings of ACM SIGGRAPH'04*, pages 322–328, 2004.

[63] O. Meruvia Pastor, B. Freudenberg, and T. Strothotte. Real-time animated stippling. *IEEE Computer Graphics and Applications*, 23(4):62–68, 2003.

[64] B. Pflesser, A. Petersik, U. Tiede, K. H. Höhne, and R. Leuwer. Volume cutting for virtual petrous bone surgery. *Computer Aided Surgery*, 7(2):74–83, 2002.

[65] B. Phong. Illumination for computer generated pictures. *ACM Communications*, 18(6):311–317, 1975.

[66] P. Rheingans and D. Ebert. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):253–264, 2001.

[67] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. Local maximum intensity projection (LMIP): A new rendering method for vascular visualization. *Journal of Computer Assisted Tomography*, 22(6):912–917, 1998.

[68] Y. Sato, C. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Yoshida, and R. Kikinis. Tissue classification based on 3d local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.

[69] F. Šebej. *Karate (in Slovak language)*. Šport, 1986.

[70] B. Shin and Y. Shin. Fast normal estimation using surface characteristics. In *Proceedings of IEEE Visualization'95*, pages 159–167, 1995.

[71] M. Šonka, V. Hlaváč, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 1999.

[72] M. Straka, M. Červeňanský, A. La Cruz, A. Köchl, M. Šrámek, M. E. Gröller, and D. Fleischmann. The VesselGlyph: Focus & context visualization in CT-angiography. In *Proceedings of IEEE Visualization'04*, pages 385–392, 2004.

[73] J. Suri, S. Setarehdan, and S. Singh, editors. *Advanced Algorithmic Approaches to Medical Image Segmentation*. Springer, 2002.

[74] C. Tietjen, T. Isenberg, and B. Preim. Combining silhouettes, surface, and volume rendering for surgery education and planning. In *Proceedings of EuroVis'05 (to appear)*, 2005.

[75] T. Totsuka and M. Levoy. Frequency domain volume rendering. In *Proceedings of ACM SIGGRAPH'93*, pages 271–278, 1993.

[76] S. Treavett and M. Chen. Pen-and-ink rendering in volume visualisation. In *Proceedings of IEEE Visualization'00*, pages 203–210, 2000.

[77] F.-Y. Tzeng, E. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.

[78] P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Proceedings of Vision, Modeling, and Visualization'01*, pages 273–280, 2001.

[79] P. Vázquez, M. Feixas, M. Sbert, and A. Llobet. Viewpoint entropy: a new tool for obtaining good views of molecules. In *Proceedings of VisSym'02*, pages 183–188, 2002.

[80] A. Vilanova. *Visualization Techniques for Virtual Endoscopy*. PhD thesis, Vienna University of Technology, Austria, 2001.

[81] I. Viola, M. E. Gröller, K. Bühler, M. Hadwiger, B. Preim, and D. Ebert. Illustrative visualization. Eurographics 2005 Tutorial.

[82] I. Viola, A. Kanitsar, and M. E. Gröller. Importance-driven volume rendering. In *Proceedings of IEEE Visualization'04*, pages 139–145, 2004.

[83] I. Viola, A. Kanitsar, and M. E. Gröller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.

[84] Web gallery of art, http://www.wga.hu/, 2005.

[85] D. Weiskopf, K. Engel, and T. Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, 2003.

[86] L. Westover. Footprint evaluation for volume rendering. In *Proceedings of ACM SIGGRAPH'90*, pages 367–376, 1990.

[87] Wikipedia free encyclopedia web site, http://wikipedia.org/, 2005.

[88] G. Winkenbach and D. Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of ACM SIGGRAPH'94*, pages 91—-100, 1994.

[89] J. Woodring, C. Wang, and H.-W. Shen. High dimensional direct rendering of time-varying volumetric data. In *Proceedings of IEEE Visualization'03*, pages 417–424, 2003.

[90] J. Zhou, A. Döring, and K. Tönnies. Distance based enhancement for focal region based volume rendering. In *Proceedings of Bildverarbeitung für die Medizin'04*, pages 199–203, 2004.

[91] K. Zuiderveld. *Visualization of multimodality medical volume data using object-oriented methods*. PhD thesis, University of Utrecht, Netherlands, 1995.

Appendix A

Curriculum Vitae

Personal data

Ivan Viola
born on June 25th, 1977, in Bratislava, Czechoslovakia.

Address:
Dietrichsteingasse 8/14, 1090 Vienna, Austria
Phone: +421 (903) 770625, +43 (650) 2362885
eMail: viola@cg.tuwien.ac.at

Education

05/2002 -	Ph.D. student in Computer Science Vienna University of Technology, Austria
10/1997 - 04/2002	Computer Science Vienna University of Technology, Austria
10/1995–06/1997	Computer Science Slovak Technical University, Bratislava, Slovakia
09/1992–06/1995	High school Gymnasium Vazovova, Bratislava, Slovakia
09/1986–06/1992	Elementary school Drieňová, Bratislava, Czechoslovakia
09/1983–06/1986	Elementary school Odborárska, Bratislava, Czechoslovakia

Jobs

04/2002–	Research Associate Vienna University of Technology, Austria
09/2000–01/2004	Teaching assistant Vienna University of Technology, Austria
06/1999–06/2004	Technician and Web Design, ArtFilm Festival Trenčianske Teplice, Slovakia
07/1998–08/2000	Programming Holiday Jobs Siemens AG, Vienna, Austria.

Publications

I. Viola, A. Kanitsar, and M. E. Gröller
Importance-driven feature enhancement in volume visualization.
IEEE Transactions on Visualization and Computer Graphics, 11(4), 2005

M. Artner, T. Möller, **I. Viola**, and M. E. Gröller
High-Quality Volume Rendering with Resampling in the Frequency Domain
In *Proceedings of EuroVis 2005*

I. Viola, A. Kanitsar, and M. E. Gröller
Importance-Driven Volume Rendering
In *Proceedings of IEEE Visualization 2004*

I. Viola, A. Kanitsar, and M. E. Gröller
GPU-based Frequency Domain Volume Rendering
In *Proceedings of SCCG 2004*

I. Viola, A. Kanitsar, and M. E. Gröller
Hardware-Based Nonlinear Filtering and Segmentation using HLSL
In *Proceedings of IEEE Visualization 2003*

M. Hadwiger, **I. Viola**, T. Theußl, H. Hauser
Fast and Flexible HQ Texture Filtering With Tiled High-Res Filters
In *Proceedings of Vision, Modeling and Visualization 2002*

L. Neumann, B. Csébfalvi, **I. Viola**, M. Mlejnek, and M. E. Gröller
Feature-Preserving Volume Filtering
In *Data Visualization 2002*

I. Viola
Applications of Hardware-Accelerated Filtering in Computer Graphics
M.Sc. Thesis, Vienna University of Technology 2002

Professional Activities

- » Eurographics 2005 Tutorial on *Illustrative Visualization*
- » Organizer of CESCG 2004 international seminar for students
- » Referee of the following conferences:
  - › Eurographics
  - › IEEE Visualization
  - › Eurographics / IEEE VGTC Symposium on Visualization
  - › Vision, Modeling, and Visualization
  - › Symposium on Interactive 3D Graphics and Games
  - › Spring Conference on Computer Graphics
  - › Winter School on Computer Graphics
  - › GRAPHITE
  - › SIBGRAPI
  - › SPIE Conference On Visualization and Data Analysis
  - › Central European Seminar on Computer Graphics

Hobbies

Photography, Film Art, Music, Books, Running, Snowboarding, Ski-Alpinism, Trekking, Traveling

Status Quo May 2005