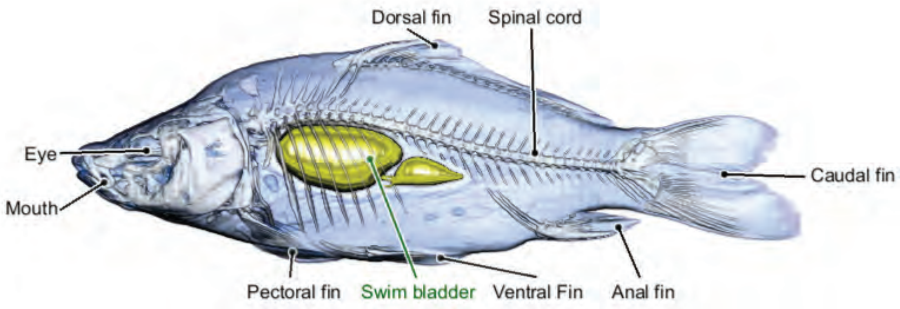
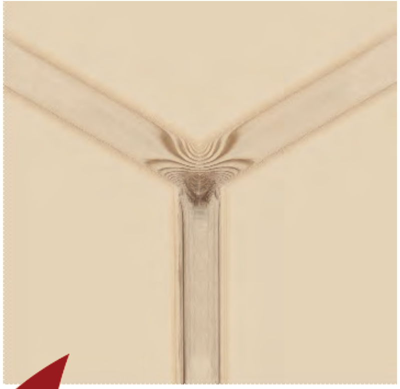
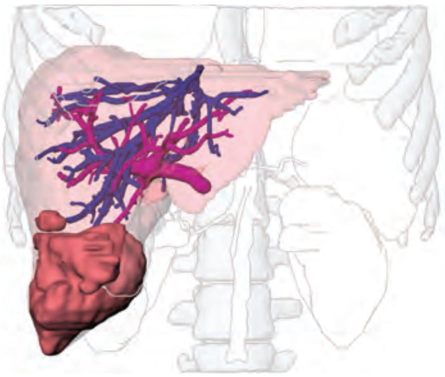
Tutorial 4: Illustrative Visualization

Ivan Viola, Meister Eduard Gröller, Markus Hadwiger, Katja Bühler
Bernhard Preim, Mario Costa Sousa, David Ebert, and Don Stredney







General Information

Illustrative Visualization

Ivan Viola[†], Meister E. Gröller[†], Markus Hadwiger[‡], Katja Bühler[‡],
Bernhard Preim[§], Mario Costa Sousa[¶], David Ebert[||] and Don Stredney[**]

[†]Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria
[‡]VRVis Research Center, Vienna, Austria
[§]Department of Simulation and Graphics, University of Magdeburg, Germany
[¶]Department of Computer Science, University of Calgary, Canada
[||]School of Electrical and Computer Engineering, Purdue University, USA
[**]Ohio Supercomputer Center, USA

[†]{viola | meister}@cg.tuwien.ac.at, [‡]{hadwiger | buehler}@vrvis.at,
[§]preim@isg.cs.uni-magdeburg.de, [¶]mario@cpsc.ucalgary.ca, [||]ebertd@purdue.edu, [**]don@osc.edu

Abstract

The tutorial presents state-of-the-art visualization techniques inspired by traditional technical and medical illustrations. Such techniques exploit the perception of the human visual system and provide effective visual abstractions to make the visualization clearly understandable. Visual emphasis and abstraction has been used for expressive presentation from prehistoric paintings to nowadays scientific and medical illustrations. Many of the expressive techniques used in art are adopted in computer graphics, and are denoted as illustrative or non-photorealistic rendering. Different stroke techniques, or brush properties express a particular level of abstraction. Feature emphasis or feature suppression is achieved by combining different abstraction levels in illustrative rendering.

Challenges in visualization research are very large data visualization as well as multi-dimensional data visualization. To effectively convey the most important visual information there is a significant need for visual abstraction. For less relevant information the dedicated image space is reduced to enhance more prominent features. The discussed techniques in the context of scientific visualization are based on iso-surfaces and volume rendering. Apart from visual abstraction, i.e., illustrative representation, the visibility of prominent features can be achieved by illustrative visualization techniques such as cut-away views or ghosted views. The structures that occlude the most prominent information are suppressed in order to clearly see more interesting parts. A different smart way to provide information on the data is using exploded views or other types of deformation. Furthermore intuitive feature classification via 3D painting and manipulation with the classified data including label placement is presented.

Discussed non-photorealistic and illustrative techniques from visualization and graphics are shown from the perspective as tools for illustrators from medicine, botany, archeology, and zoology. The limitations of existing NPR systems for science illustration are highlighted, and proposals for possible new directions are made. Illustrative visualization is demonstrated via application-specific tasks in medical visualization. An important aspect as compared to traditional medical illustrations is the interactivity and real-time manipulation of the acquired patient data. This can be very useful in anatomy education. Another application area is surgical planning which is demonstrated with two case studies: neck dissection and liver surgery planning.

Prerequisites

The tutorial assumes basic knowledge in scientific visualization algorithms and non-photorealistic rendering techniques. Any knowledge of illustration techniques for science and medicine may be helpful but is not required. In general the level of the tutorial can be considered as beginning.

Intended Audience

Intended audience consists of domain experts like medical doctors and biologists, visualization researchers, programmers, illustrators, and others interested in techniques for meaningful depictions of the data and its applicability to current visualization challenges.

Schedule

The tutorial is planned as a full day tutorial. The talks are grouped into three main parts: Introduction, Illustrative Techniques in Visualization, and Applications of Illustrative Techniques in Science and Medicine. A more detailed schedule including speaker's name and talk length is given in the table in Figure 1. For further details about the tutorial see the associated webpage http://www.cg.tuwien.ac.at/groups/vis/vis_tutorial/.

Outline

The tutorial is divided into the following talks:

K. Bühler: Human Visual Perception and Illustrative Aspects of Art employs a survey on the history of technical, scientific and medical illustrations as motivation to demonstrate how artists and graphic designers developed the ability to encode complex information within a single graphic representation. We start with an overview on physiological and psychological aspects of human perception, and their manifestation in common illustration techniques and design principles. This will include an introduction to commonly used materials, and basic artistic elements like points, lines, continuous tone and colour. A discussion on the use of perspective, focus, selective enhancement, transparency and abstraction will lead us to advanced design principles that aim at representing multi layered information using e.g. focus and context, cut-away views, exploded views, and the combination of realism and abstraction. Weighing up advantages and limitations of "hand made" scientific illustrations will link up with the following talks that introduce and discuss the art of illustrative rendering.

Introduction		
M. E. Gröller	Introduction of Speakers and Topics	10 min
M. E. Gröller and K. Bühler	Human Visual Perception and Illustrative Aspects of Art	25 min
D. Ebert	Illustrative and Non-Photorealistig Rendering in Computer Graphics	25 min
Illustrative Techniques in Visualization		
M. Hadwiger	Illustrative Visualization for Isosurfaces and Volumes	60 min
I. Viola	Smart Visibility in Visualization	60 min
Applications of Illustrative Techniques in Science and Medicine		
M. C. Sousa	Visualization Tools for the Science Illustrators: Evaluations and Requirements	40 min
D. Ebert	Illustration Inspired Flow Visualization	20 min
D. Ebert	Interactive Medical Illustration System for Surgical Simulation and Education	20 min
D. Stredney	Visualization: From My Perspective	40 min
B. Preim	Case Studies for Surgical Planning using Illustrative Visualization	60 min
Closing Remarks and Discussion		
All	Discussion	10 min

Figure 1: Schedule of the Tutorial on Illustrative Visualization

D. Ebert: Illustrative and Non-Photorealistig Rendering in Computer Graphics introduces a category of rendering techniques that simulate a style of a particular artistic painting or illustration technique. In contrast to traditional photorealistic rendering, the category of illustrative or non-photorealistic rendering (NPR) exploits artistic abstraction to express the prominence of rendered objects. We describe general NPR principles and discuss several NPR categories defined by material basis (ink, charcoal, paint) or stroke simulation (brushes, hatching, stippling). Furthermore we show how to use illustrative rendering techniques as visual abstraction levels for form and shape emphasis. Finally we describe how to focus the viewer's attention by varying detail of painterly rendering according to the distance from the focus (see Figure 2).

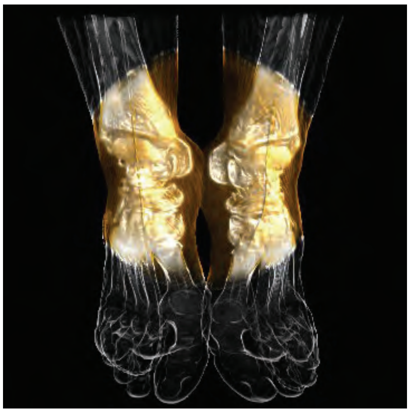


Figure 2: Distance-from-focus rendering combining contour rendering with direct volume rendering.

M. Hadwiger: Illustrative Visualization of Isosurfaces and Volumes describes visualization techniques for rendering isosurfaces with a variety of different shape cues and illustrative techniques such as pen-and-ink style rendering, focusing on styles that use or depict surface curvature information, such as rendering ridge and valley lines, and hatching. In addition to techniques operating on meshes, we illustrate how non-polygonal isosurfaces that are extracted on-the-fly can be annotated with shape cues based on implicit surface curvature. We illustrate a GPU-based rendering pipeline for high-quality rendering of isosurfaces with real-time curvature computation and shading.

After decribing surface-based illustration styles we continue with full volume rendering. We show that segmentation information is an especially powerful tool for depicting the objects contained in medical data sets in varying styles. A combination of non-photorealistic styles with standard direct volume rendering is a very effective means for separating focus from context objects or regions. We describe the concept of two-level volume rendering that integrates different rendering modes and compositing types by using segmented data and per-object attributes (see Figure 3).

I. Viola: Smart Visibility in Visualization first discusses techniques that modify the visual representation of the data by incorporating viewpoint information to provide maximal visual information. In illustration such techniques are called cut-away views or ghosted views. We discuss basic principles and techniques for automatic generation of cut-away and ghosted visualizations. One approach is importance-driven feature enhancement, where the visibility of a particular feature is determined according to assigned importance information (Figure 4). The most appropriate level of abstraction is specified automatically to unveil the most important information. We show the applicability of smart visibility techniques for the visualization of complex dynamical systems, visualization of peripheral arteries, and visualization of the human abdomen. Another approach is context-preserving illustrative volume rendering (Figure 5), which 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

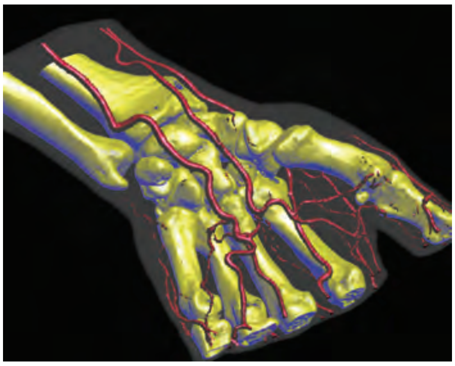


Figure 3: Interactive two level volume rendering where the skin is rendered with MIP, bones with tone shading, and vessels with shaded iso-surfacing.

shape of partially transparent parts and therefore additional information can be shown there.

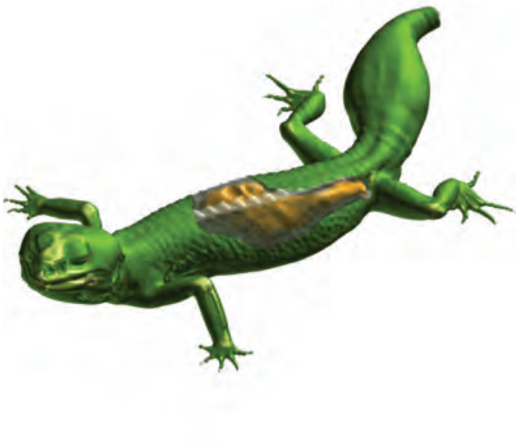


Figure 4: Importance-driven volume rendering of the Leopard gecko dataset. The internal structure is automatically emphasized by suppressing the occluding body parts.

The talk continues with a description of a system for direct volume manipulation (such as 3D painting) in combination with cutaway views. Here manipulation metaphores inspired by traditional illustration are discussed. An important aspect for readily understandable visualization is labeling the data with annotations (see Figure 6). The combination of automatic label placement with visualized data is presented and new labeling metaphors from the field of information visualization are discussed.

The second category of smart visibility techniques are based on object deformation and object splitting. These techniques are closely related to exploded views, often used for assembly instructions. We discuss visualization techniques that separate context information to unveil the inner focus information by splitting the context into parts and moving them apart. Another visualization technique enables browsing within the data by applying deformations like leafing, peeling, or spreading. In the case of time-varying data we present another visualization technique which is related to exploded views and is denoted as fanning in time.

M. C. Sousa: Visualization Tools for the Science Illustrators: Evaluations and Requirements introduces the field of Non-Photorealistic Rendering (NPR) from the point of view of the traditional science illustrator. Topics include the interplay between the NPR pipeline and the communication/production processes of traditional illustration, components of the NPR pipeline, such as the type of input data (images, 3D models, laser scans, MRI), capabilities of existing NPR systems and subject areas such as medicine, botany, archaeology, zoology, among others (Figure 7). This presentation will then focus on discussing the limitations of existing NPR systems for science illustration, followed with proposals for possible extensions and new directions. Evaluations from trained illustrators of the use and quality of the existing techniques and tools will be discussed. We will also present and discuss a number of important requirements provided by science illustrators for devising novel computer graphics/NPR tools within three main categories of systems: (1) fully interactive, expecting the user to produce traditional images from scratch (drawing/painting systems), (2) fully automatic, producing images using automatic techniques (renderers, image processing), and (3) hybrid NPR solutions, known as ”NPR Interactive Rendering”, where traditional renderings are produced partly by the system and partly by the user.

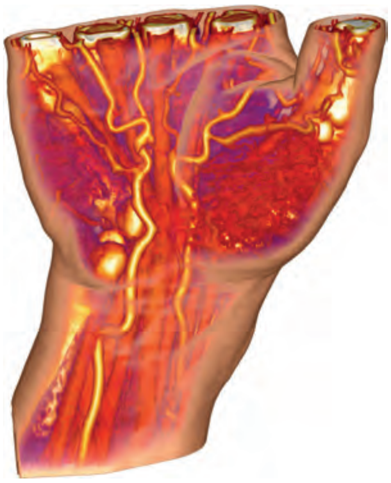


Figure 5: Visualization of a human hand using a dynamic opacity approach as a function of the specular highlight level.

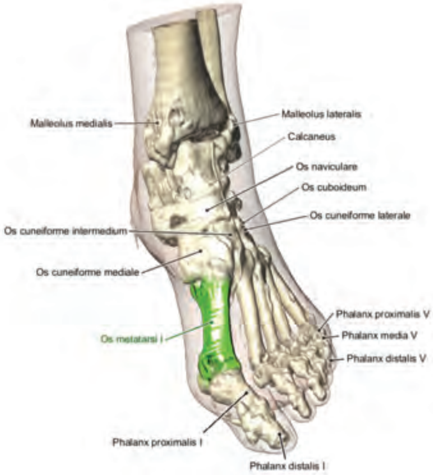


Figure 6: Volume manipulation and classification and automatic label placement. All bonal structures have been classified using direct 3D painting.

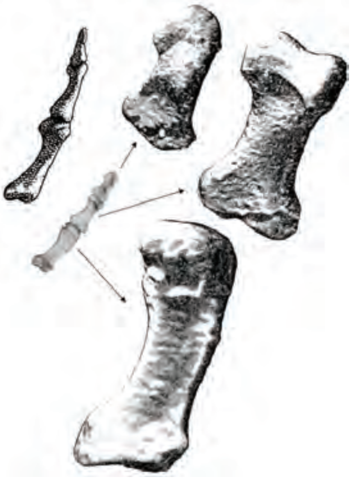


Figure 7: Rendering of three thumb bones (from top to bottom): distal phalange, proximal phalange and metacarpal 1.

D. Ebert: Illustration Inspired Flow Visualization goes through the history of flow illustration over the past centuries, and provides analysis of existing effective styles and visualization techniques. Then a new interactive flow illustration system is introduced. A more detailed overview of the system functionality and implemented interaction techniques is given. The applicability in flow visualization is demonstrated using new visualization techniques applied on several time-varying and unstructured flow datasets (see Figure 8).

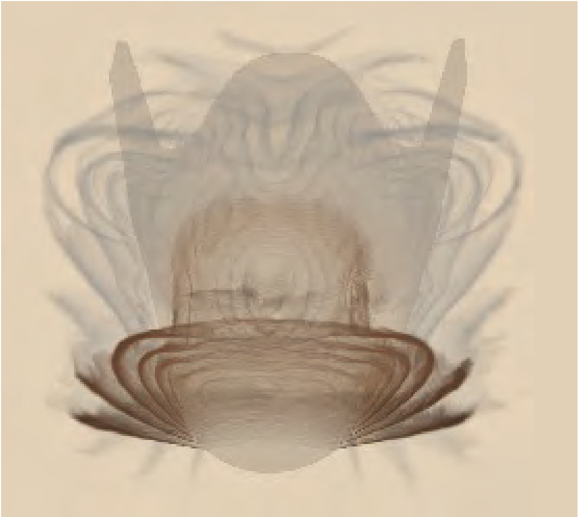


Figure 8: Stylistic illustrative visualization of flow over the X38 spacecraft during re-entry, highlighting the bow shock at the nose of the spacecraft.

D. Ebert: Interactive Medical Illustration System for Surgical Simulation and Education shows the applicability of illustrative visualization in medical visualization. A system for surgical simulation and anatomy education is presented. We point out that the design of an effective illustrative presentation style is application-specific, i.e., there are different criteria for training and for education purposes. The presentation of information is highly dependent on the level of user expertise. We treat interactive illustrative visualization for anatomical education and temporal bone surgical planning.

D. Stredney: Visualization: From My Perspective will present his perspective on visualization and emerging developments in NPR techniques and their use. After a brief introduction of his background, Don will present the key issues of sensemaking and their use in clinical research and training that use visualization. Don will present an overview of representation from a physiological view, and draw parallels between human visual processing, learning, and aesthetics. Current work from funded research projects that integrate aspects of NPR for surgical training will be presented. Finally, suggested guidelines for promoting adoption and creating diverse teams for development and adaptation will be presented.

B. Preim: Case Studies for Surgical Planning using Illustrative Visualization explains how illustrative visualization can significantly improve the spatial perception of feature arrangement for surgical planning and education training. Both discussed applications, i.e., the liver surgical training system and the neck dissection planning (Figure 9), are based on a database of clinical data. In these specific visualization tasks there are many overlapping interesting features. We present how a suitable selection of visual abstractions, such as a combination of silhouette, surface, and volume rendering or cut-away illustrative techniques, can make the visualization clearly understandable.

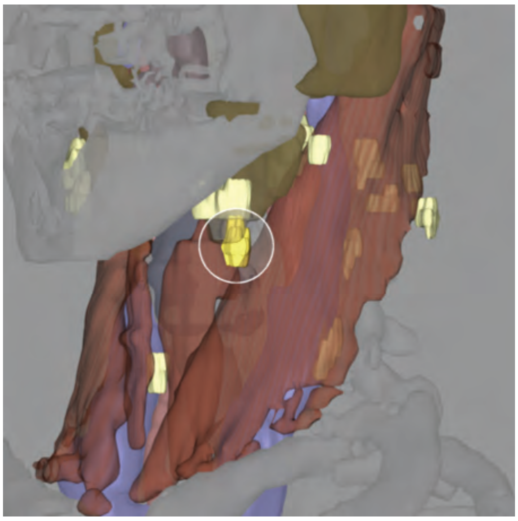


Figure 9: Neck dissection planning with emphasis on the lymph nodes inspired by cut-away views.

Apart from educational aspects, both applications use visualization and interaction techniques to support surgical decisions. The liver surgery planning system is designed for interactive resection planning. The neck dissection planning system is designed for interactive path-planning for minimal invasive interventions.

Presenter's Background

Ivan Viola graduated in 2002 from the Vienna University of Technology, Austria, as a Dipl.-Ing. (MSc) in the field of computer graphics and visualization. He received his PhD in 2005 for his thesis "Importance-Driven Expressive Visualization". Currently he is managing the ex**vis**ation research project (www.cg.tuwien.ac.at/research/vis/exvisation) focusing on development of novel methods for automatically generating expressive visualizations of complex data. Viola has co-authored several scientific works published on international conferences such as IEEE Visualization, EuroVis, and Vision Modeling and Visualization and acted as a reviewer for conferences in the field of computer graphics and visualization.

Meister E. Gröller is associate professor at the Institute of Computer Graphics and Algorithms (ICGA), Vienna University of Technology. In 1993 he received his PhD from the same university. His research interests include computer graphics, flow visualization, volume visualization, and medical visualization. He is heading the visualization group at ICGA. The group performs basic and applied research projects in the area of scientific visualization. Dr. Gröller has given lecture series on scientific visualization at various other universities (Tübingen, Graz, Praha, Bahia Blanca, Magdeburg). He is a scientific proponent and member of the Scientific Advisory Committee of the VRVis Kplus center of excellence. The center performs applied research in virtual reality and visualization. Dr. Gröller co-authored more than 100 scientific publications and acted as a reviewer for numerous conferences and journals in the field. He also serves on various program and paper committees. Examples include Computers&Graphics, IEEE Transactions on Visualization and Graphics, EuroVis, IEEE Visualization conference, Eurographics conference. He is head of the working group on computer graphics of the Austrian Computer Society and member of IEEE Computer Society, ACM (Association of Computing Machinery), GI (Gesellschaft für Informatik), OCG (Austrian Computer Society).

Markus Hadwiger is a senior researcher in the Medical Visualization department at the VRVis Research Center in Vienna, Austria. He received a PhD degree in computer science from the Vienna University of Technology in 2004, concentrating on high-quality real-time volume rendering and texture filtering with graphics hardware. Results on rendering segmented volumes and non-photorealistic volume rendering have been presented at IEEE Visualization 2003. He is regularly teaching courses and seminars on computer graphics, visualization, and game programming, including two courses at the annual SIGGRAPH conference, and two tutorials at IEEE Visualization. Before concentrating on scientific visualization, he was working in the area of computer games and interactive entertainment.

Katja Bühler is head of the Medical Visualization department at VRVis Research Center for Virtual Reality and Visualization and external lecturer for medical visualization at the Vienna University of Technology in Vienna, Austria. Her current research topics are motivated by real world applications in the medical field and focus mainly on techniques for computer aided diagnosis and surgery simulation, including specialized solution for segmentation and visualization. She studied Mathematics with focus on Geometry, Numerics and Computer Science at the University of Karlsruhe, Germany and received her diploma in pure Mathematics in 1996. In 2001 she received a PhD in computer science from the Institute of Computer Graphics and Algorithms, Vienna University of Technology for her work on reliable geometry processing. Katja Bühler has worked as researcher at the Institute for Applied Mathematics, University of Karlsruhe, Germany and the Center of Computer Graphics and Applied Geometry, Universidad Central de Venezuela, Caracas, Venezuela. She became assistant professor at the Institute of Computer Graphics and Algorithms, Vienna University of Technology in 1998 and was teaching courses in computer graphics, algorithms and data structures, and programming. In 2002 she joined the medical visualization group at VRVis as senior researcher and became key researcher in 2003.

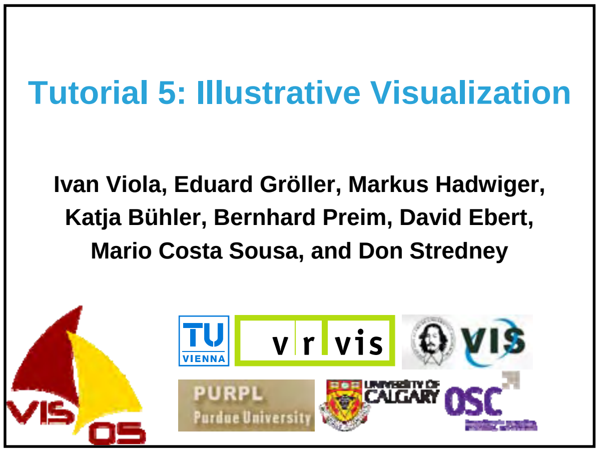
Bernhard Preim worked for four years as project leader Surgery planning at the Center for Medical Visualization and Diagnostic Systems (MeVis Bremen, Germany) before he was appointed as full professor for visualization at the computer science department at the Otto-von-Guericke-University of Magdeburg, Germany. His research group focusses on medical visualization and specific applications in surgical education and surgery planning. He is speaker of the working group Medical Visualization in the German Society for Computer Science. He is member of the scientific advisary boards of ICCAS (International Competence Center on Computer-Assisted Surgery Leipzig, since 2003) and CURAC (German Society for Computer- and Roboter-assisted Surgery, since 2004) and Visiting Professor at the University of Bremen. He is author and co-author of more than 80 publications, most of them dealing with interactive visualizations in medical applications. His research interests include 3D interaction techniques, visualization techniques for medical volume data (visualization of vasculature, transfer function design, illustrative medical visualization) and computer support for medical diagnosis and treatment planning, in particular neck dissection planning and liver surgery planning.

Mario Costa Sousa is an Assistant Professor in the Department of Computer Science at the University of Calgary. He holds a M.Sc. (PUC-Rio, Brazil) and a Ph.D. (University of Alberta) both in Computer Science. He performs research in non-photorealistic rendering (NPR), illustrative visualization, 3D modeling and volumetric display software. His current focus is on research and development of NPR methods for 3D model construction/analysis, natural media simulation, rendering techniques and systems for computer-generated illustrative visualization and composition in two main contexts: (1) traditional illustration, by providing tools to help scientific and medical illustrators; (2) scientific analysis and visualization, by mainly providing novel ways on visualizing scientific data, physical phenomena, simulations, etc., and by presenting abstractions to users in ways that reconcile expressiveness and ease-of-use. Dr. Sousa also coordinates the Render Group, the NPR research wing at the Computer Graphics Lab at the University of Calgary.

David Ebert is an Associate Professor in the School of Electrical and Computer Engineering at Purdue University. His research interests are scientific, medical, and information visualization, computer graphics, animation, and procedural techniques. Dr. Ebert performs research in volume rendering, illustrative visualization, realistic rendering, procedural texturing, modeling, and animation, and modeling natural phenomena. Ebert has been very active in the graphics community, teaching courses, presenting papers, serving on and co-chairing many conference program committees, serving on the ACM SIGGRAPH Executive Committee and serving as Editor in Chief for IEEE Transactions on Visualization and Computer Graphics. Ebert is also editor and co-author of the seminal text on procedural techniques in computer graphics, Texturing and Modeling: A Procedural Approach, whose third edition was published in December 2003.

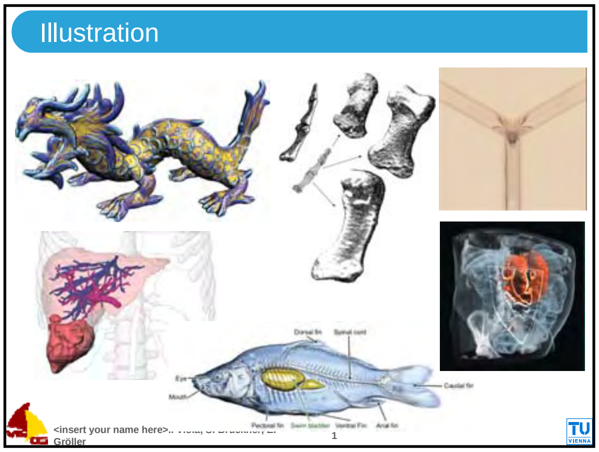
Don Stredney is research scientist for Biomedical Applications and Director of the Interface Lab at OSC (Ohio Supercomputer Center). In addition, Don is a member of the Experimental Therapeutics Program at the Comprehensive Cancer Center, and an Associate Member of the Head and Neck Oncology Program at the Arthur G. James Cancer Hospital and Solove Research Institute in Columbus, Ohio. Dons research involves the exploration of high performance computing and the application of advanced interface technology for the development of more intuitive methods for interaction with large and complex multimodal data sets. His research interests lie in theories of representation, specifically the representation and interaction with synthesized biomedical phenomena for clinical and biomedical research and education. Don is co-recipient of the Smithsonian Institute/Computerworld 1996 Information Technology Leadership Award sponsored by Cray Research Inc. for the design and implementation of a computer simulation environment for training residents in the delivery of regional anesthesia techniques. Don currently has funded projects through NIDCD, NIOSH, NSF and DOE/ASCI. In addition, Don has been an investigator on projects from the National Institutes of Health/National Library of Medicine, the National Institute for Drug Addiction, Department of Defense, Medical Army Material

Command, Department of Energy, Lockheed Martin, the National Institute for Disability and Rehabilitation Research, Harvard Medical School, Ameritech, the Committee on Institutional Cooperation of the Big Ten and University of Chicago, and Cray Research Inc.

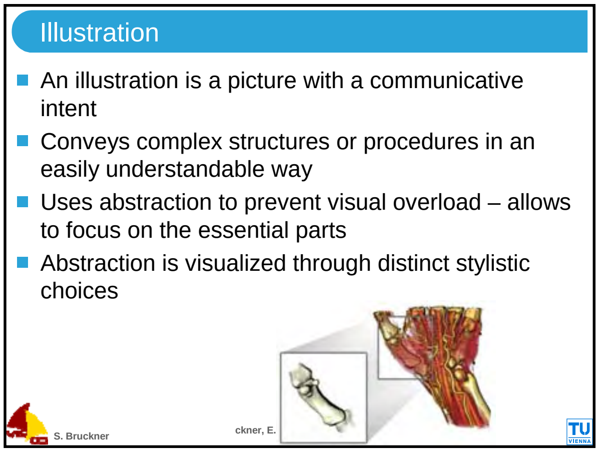


Tutorial 5: Illustrative Visualization

Ivan Viola, Eduard Gröller, Markus Hadwiger,
Katja Bühler, Bernhard Preim, David Ebert,
Mario Costa Sousa, and Don Stredney

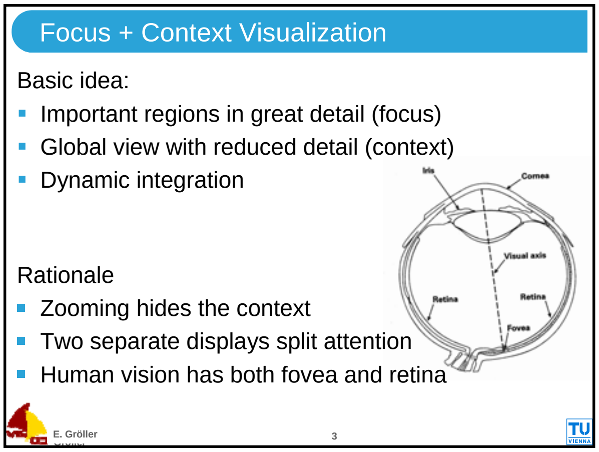


Illustration



Illustration

- An illustration is a picture with a communicative intent
- Conveys complex structures or procedures in an easily understandable way
- Uses abstraction to prevent visual overload – allows to focus on the essential parts
- Abstraction is visualized through distinct stylistic choices



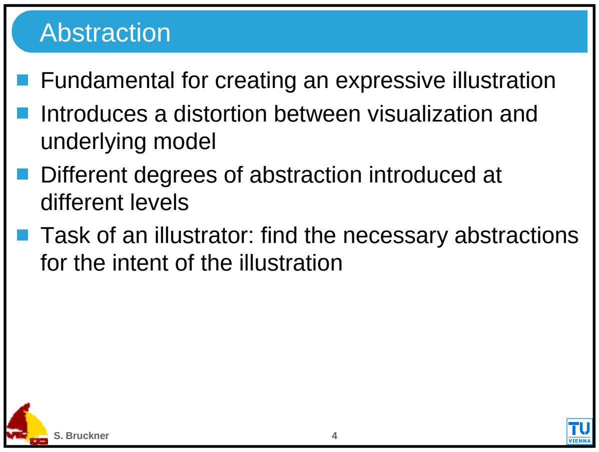
Focus + Context Visualization

Basic idea:

- Important regions in great detail (focus)
- Global view with reduced detail (context)
- Dynamic integration

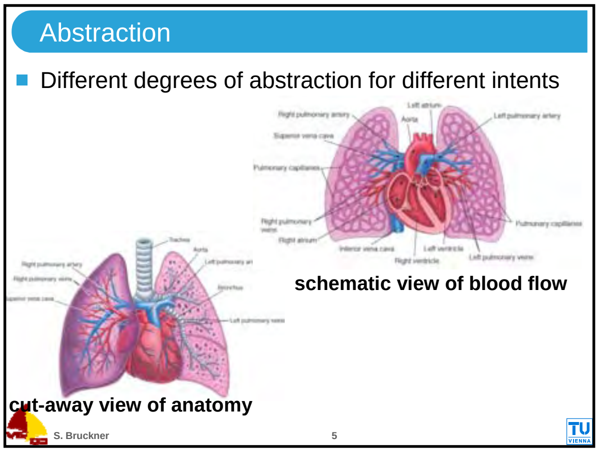
Rationale

- Zooming hides the context
- Two separate displays split attention
- Human vision has both fovea and retina



Abstraction

- Fundamental for creating an expressive illustration
- Introduces a distortion between visualization and underlying model
- Different degrees of abstraction introduced at different levels
- Task of an illustrator: find the necessary abstractions for the intent of the illustration



Abstraction

- Different degrees of abstraction for different intents

schematic view of blood flow

cut-away view of anatomy

Abstraction

- Goals of abstraction techniques
  - Communicate shape and structure
  - Emphasize or de-emphasize
  - Prevent visual overload
  - Suggest artificiality
  - Ensure visibility of important structures
  - Provide spatial context
- **„As detailed as necessary – as simple as possible“**

Low-Level Abstraction Techniques

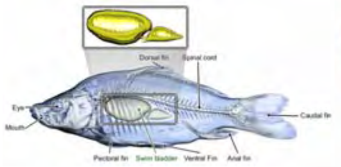
- Concerned with **how** different objects are presented
- Stylized depiction
  - Silhouettes and contours, pen and ink, stippling, hatching, ...

High-Level Abstraction Techniques

- Deal with **what** should be visible and recognizeable
- Smart visibility
  - Cutaways, breakaways, ghosting, exploded views, ...

Illustrative Visualization

- Illustrative Visualization: computer supported interactive and expressive visualizations through abstractions as in traditional illustrations



[Bruckner 2005]

Schedule

Introduction

- **8:30 Eduard Gröller, Katja Bühler**:
  Introduction of Speakers and Topics
  Human Visual Perception and Illustrative Aspects of Art
- **9:05 D. Ebert**:
  Illustrative and Non-Photorealistic Rendering

Illustrative Techniques in Visualization

- **9:30 Markus Hadwiger**:
  Illustrative Visualization for Isosurfaces and Volumes
- **10:00-10:30** *Coffee break*
- **11:00-12:00 Ivan Viola**:
  Smart Visibility in Visualization

Schedule

Applications of Illustrative Techniques in Science and Medicine

- **12:00 Mario Costa Sousa**:
  Visualization Tools for the Science Illustrators: Evaluations and Requirements
- **12:30-13:45** *Lunch*
- **14:15 David Ebert**:
  Illustration Inspired Flow Visualization
  Interactive Medical Illustration System for Surgical Simulation and Education
- **15:05 Don Stredney**:
  Visualization: From Illustrator's Perspective
- **15:45-16:15** *Coffee Break*
- **16:15 Bernhard Preim**:
  Case Studies for Surgical Planning using Illustrative Visualization

Discussion and Closing Remarks

- **17:15 All**

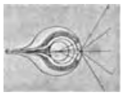
Human Visual Perception and Illustrative Aspects of Art

Human Visual Perception and Illustrative Aspects of Art

Eduard Gröller[1] and Katja Bühler[2]

[1]Institute of Computer Graphics and Algorithms,
Vienna University of Technology

[2] VRVis Research Center, Vienna



Overview

- Part 1: Drawings
  - Media
  - Elements
  - Perceptual Aspects
- Part 2: Scientific Illustrations
  - Development of Scientific Illustrations
  - Towards interactive 3D illustrations….

Part 1 - Drawings

Media, Elements and Perceptual Aspects

Eugene Delacroix; *Study for "The Death of Sardanapalus"* 1827-28; Pastel with chalk over wash on paper; Art Institute of Chicago. (WebMuseum)

Media

- Friable media:
  - Pencils, Graphite sticks
  - Charcoal, Chalk
- Pigments
  - Ink
  - Carbon dust
  - Aquarell, Gouache, ….

Peter Paul Rubens 1577-1640 ; *St. George Slaying the Dragon* Pen with brown ink and brown wash; Louvre (WebMuseum)

Eugene Delacroix; *Study for "The Death of Sardanapalus"* 1827-28; Pastel with chalk over wash on paper; Art Institute of Chicago. (WebMuseum)

Media – Transferring Instruments

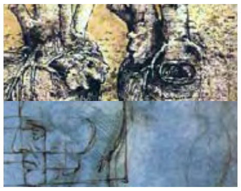
Transferring Instruments

- Pens
  - Reed, Birdfeather, Metal, Technical Pens
- Brushes

Johann Füssli (1741-1825) ; *Perseus Returning the Eye of the Graii*; Pen; City Art Gallery at Birmingham, England (WebMuseum)

Support

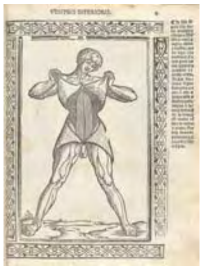
- Stone, Bone, Metal, ….
- Papyrus, Parchment, Wood,…
- Paper, Cardboard



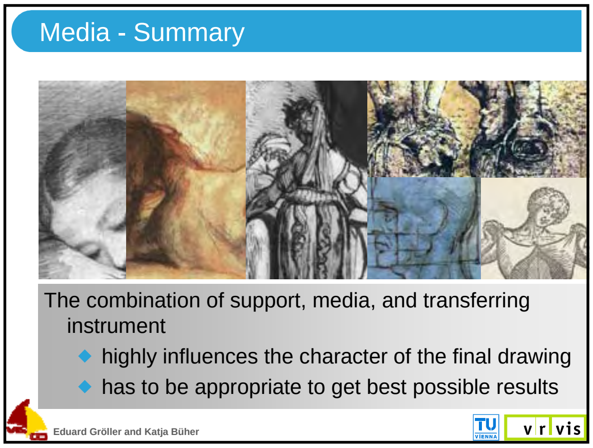
Both images by Leonardo Da Vinci, Downloaded at GFMER

Media - Reproduction Techniques

- **Basic techniques (one color)**
  - Relief printing
  - Gravure / engraving
- **Colored illustrations**
  - Hand coloring, printing multiple layers
- **Modern techniques**
  - Photography
  - Modern digital imaging/printing



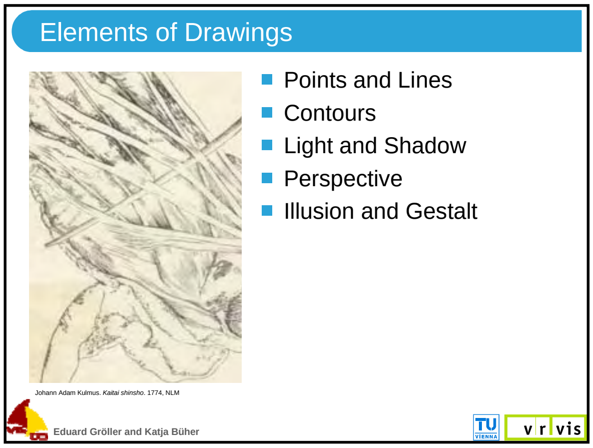
Illustration, Berengario da Carpi, Jacopo. *Isagogae breues, perlucidae ac uberrimae, in anatomiam humani corporis a communi medicorum academia usitatam*. Woodcut, Bolongna 1523. NLM



Media - Summary

The combination of support, media, and transferring instrument
- highly influences the character of the final drawing
- has to be appropriate to get best possible results

Eduard Gröller and Katja Büher

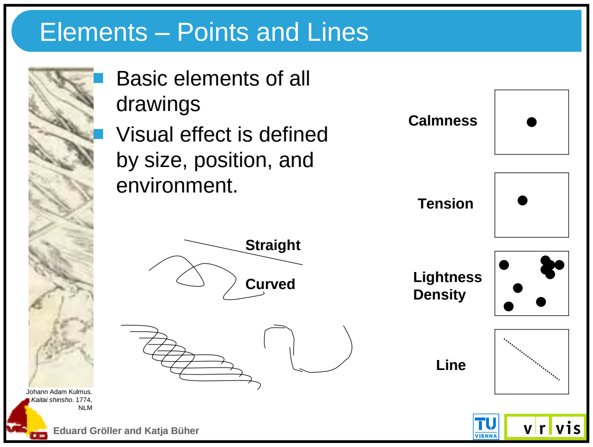


Elements of Drawings

- Points and Lines
- Contours
- Light and Shadow
- Perspective
- Illusion and Gestalt

Johann Adam Kulmus. *Kaitai shinsho*. 1774, NLM

Eduard Gröller and Katja Büher



Elements – Points and Lines

- Basic elements of all drawings
- Visual effect is defined by size, position, and environment.

Straight

Curved

Calmness

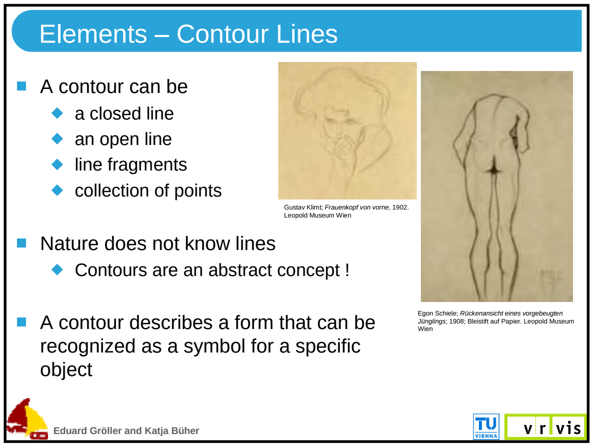
Tension

Lightness Density

Line

Johann Adam Kulmus. *Kaitai shinsho*. 1774, NLM

Eduard Gröller and Katja Büher



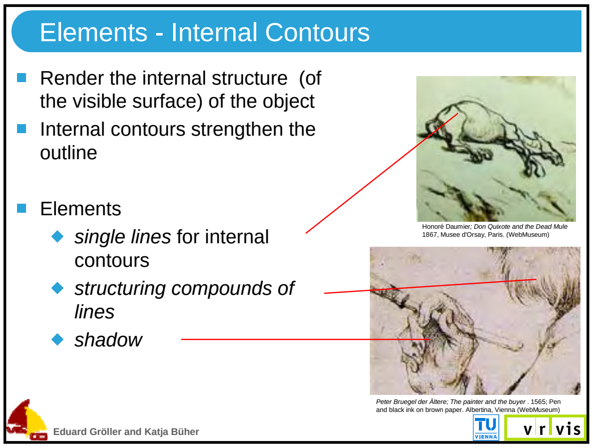
Elements – Contour Lines

- A contour can be
  - a closed line
  - an open line
  - line fragments
  - collection of points
- Nature does not know lines
  - Contours are an abstract concept !
- A contour describes a form that can be recognized as a symbol for a specific object

Gustav Klimt; *Frauenkopf von vorne*, 1902. Leopold Museum Wien

Egon Schiele; *Rückenansicht eines vorgebeugten Jünglings*; 1908; Bleistift auf Papier. Leopold Museum Wien

Eduard Gröller and Katja Büher



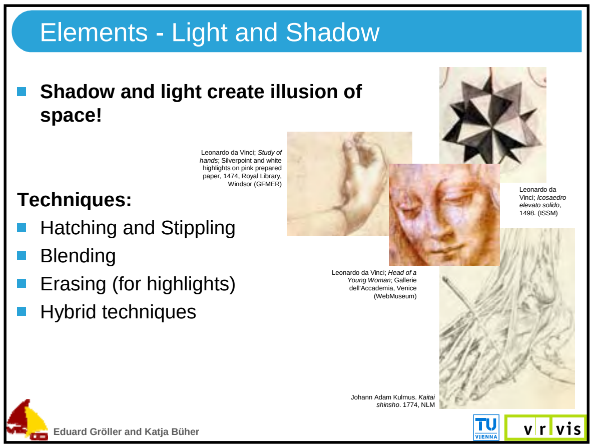
Elements - Internal Contours

- Render the internal structure (of the visible surface) of the object
- Internal contours strengthen the outline
- Elements
  - *single lines* for internal contours
  - *structuring compounds of lines*
  - *shadow*

Honoré Daumier; *Don Quixote and the Dead Mule* 1867, Musee d'Orsay, Paris. (WebMuseum)

Peter Bruegel der Ältere; The painter and the buyer . 1565; Pen and black ink on brown paper. Albertina, Vienna (WebMuseum)

Eduard Gröller and Katja Büher



Elements - Light and Shadow

- **Shadow and light create illusion of space!**

Techniques:
- Hatching and Stippling
- Blending
- Erasing (for highlights)
- Hybrid techniques

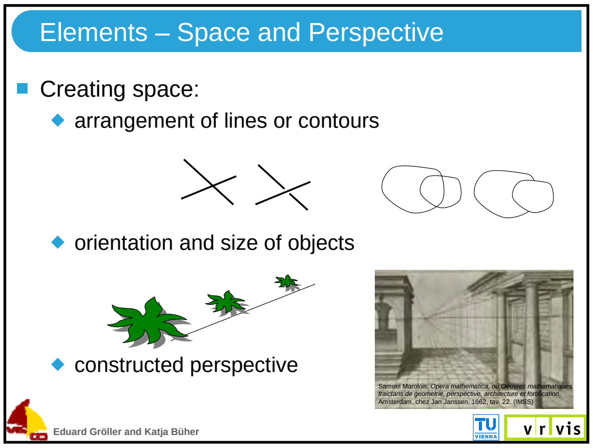
Leonardo da Vinci; *Study of hands*; Silverpoint and white highlights on pink prepared paper, 1474, Royal Library, Windsor (GFMER)

Leonardo da Vinci; *Icosaedro elevato solido*, 1498. (ISSM)

Leonardo da Vinci; *Head of a Young Woman*; Gallerie dell'Accademia, Venice (WebMuseum)

Johann Adam Kulmus. *Kaitai shinsho*. 1774, NLM

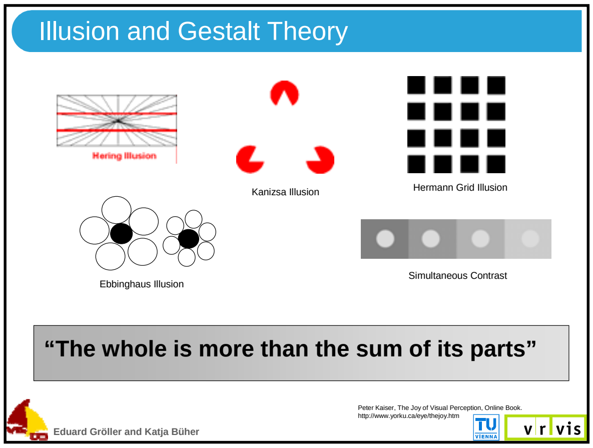
Eduard Gröller and Katja Büher



Elements – Space and Perspective

- Creating space:
  - arrangement of lines or contours
  - orientation and size of objects
  - constructed perspective

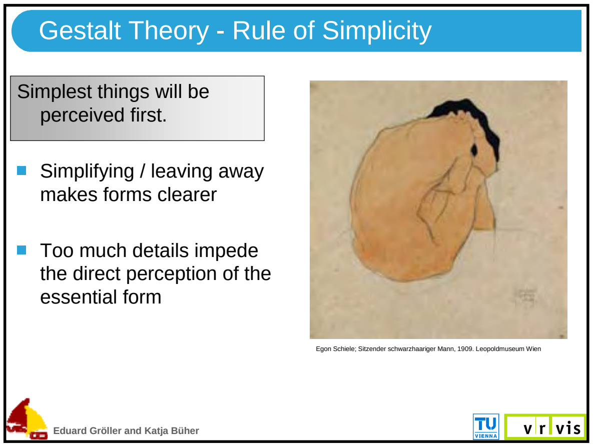
Samuel Marolois, *Opera mathematica, ou Oeuvres mathematiques traictans de geometrie, perspective, architecture et fortification*, Amsterdam, chez Jan Janssen, 1662, tav. 22. (IMSS)



Illusion and Gestalt Theory

“The whole is more than the sum of its parts”

Peter Kaiser, The Joy of Visual Perception, Online Book.
http://www.yorku.ca/eye/thejoy.htm

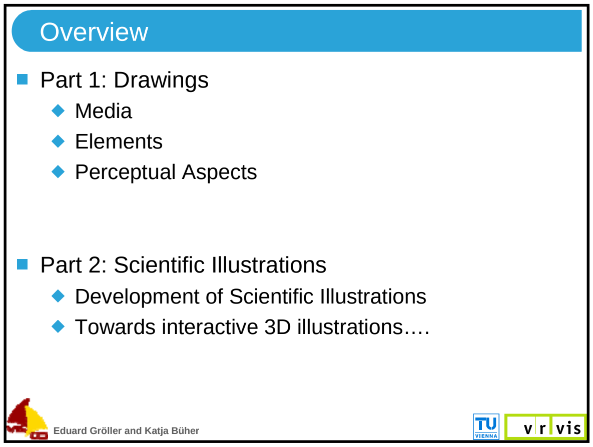


Gestalt Theory - Rule of Simplicity

Simplest things will be perceived first.

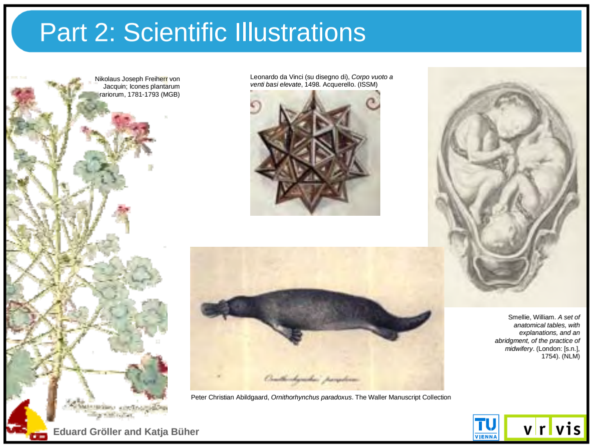
- Simplifying / leaving away makes forms clearer
- Too much details impede the direct perception of the essential form

Egon Schiele; Sitzender schwarzhaariger Mann, 1909. Leopoldmuseum Wien



Overview

- Part 1: Drawings
  - Media
  - Elements
  - Perceptual Aspects
- Part 2: Scientific Illustrations
  - Development of Scientific Illustrations
  - Towards interactive 3D illustrations….



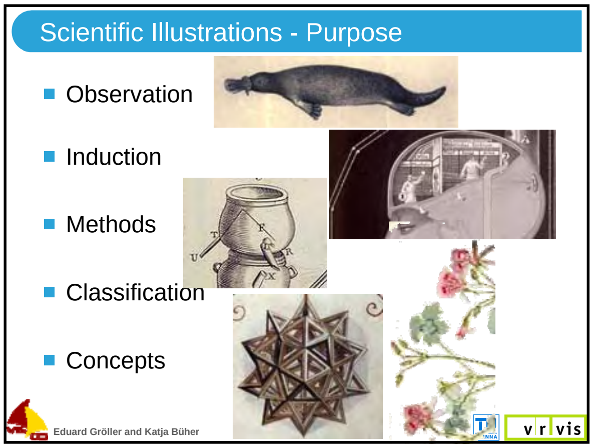
Part 2: Scientific Illustrations

Nikolaus Joseph Freiherr von Jacquin; Icones plantarum rariorum, 1781-1793 (MGB)

Leonardo da Vinci (su disegno di), *Corpo vuoto a venti basi elevate*, 1498. Acquerello. (ISSM)

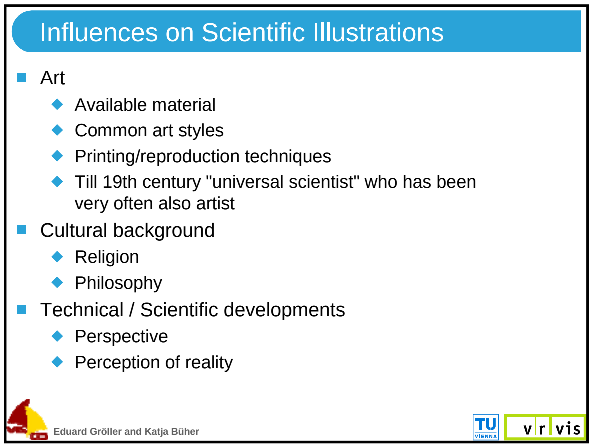
Smellie, William. *A set of anatomical tables, with explanations, and an abridgment, of the practice of midwifery*. (London: [s.n.], 1754). (NLM)

Peter Christian Abildgaard, *Ornithorhynchus paradoxus*. The Waller Manuscript Collection



Scientific Illustrations - Purpose

- Observation
- Induction
- Methods
- Classification
- Concepts

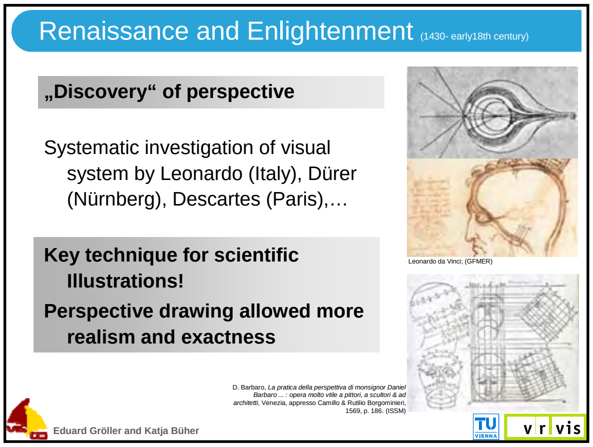


Influences on Scientific Illustrations

- Art
  - Available material
  - Common art styles
  - Printing/reproduction techniques
  - Till 19th century "universal scientist" who has been very often also artist
- Cultural background
  - Religion
  - Philosophy
- Technical / Scientific developments
  - Perspective
  - Perception of reality

Eduard Gröller and Katja Büher

Medical Illustrations - Historical Development



Renaissance and Enlightenment (1430- early18th century)

„Discovery“ of perspective

Systematic investigation of visual system by Leonardo (Italy), Dürer (Nürnberg), Descartes (Paris),…

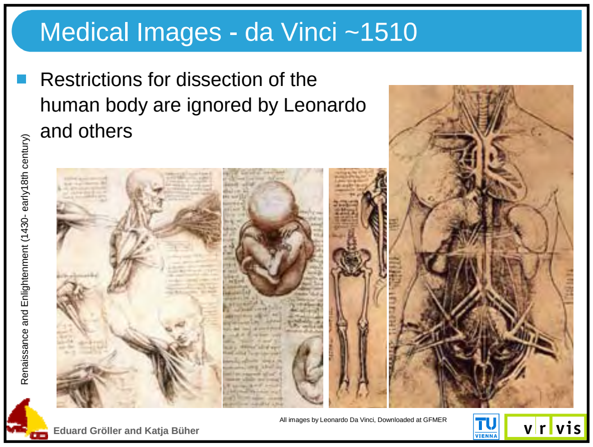
Key technique for scientific Illustrations!

Perspective drawing allowed more realism and exactness

Leonardo da Vinci; (GFMER)

D. Barbaro, *La pratica della perspettiva di monsignor Daniel Barbaro ... : opera molto vtile a pittori, a scultori & ad architetti*, Venezia, appresso Camillo & Rutilio Borgominieri, 1569, p. 186. (ISSM)

Eduard Gröller and Katja Büher



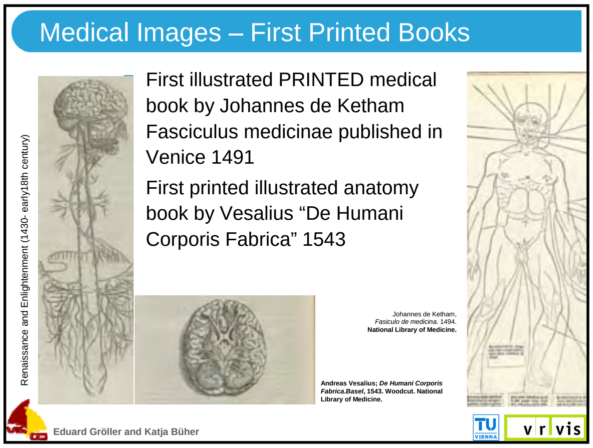
Medical Images - da Vinci ~1510

Renaissance and Enlightenment (1430- early18th century)

- Restrictions for dissection of the human body are ignored by Leonardo and others

All images by Leonardo Da Vinci, Downloaded at GFMER

Eduard Gröller and Katja Büher



Medical Images – First Printed Books

Renaissance and Enlightenment (1430- early18th century)

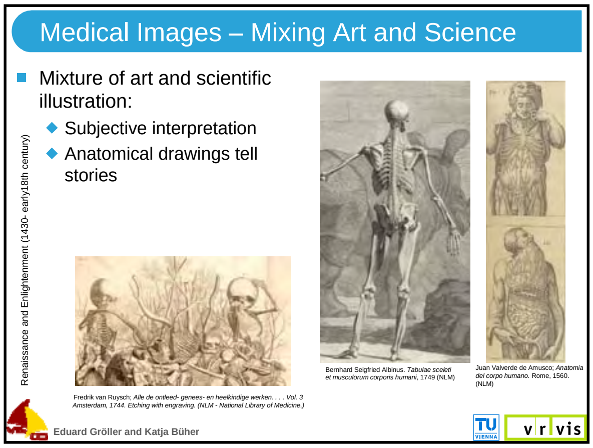
First illustrated PRINTED medical book by Johannes de Ketham Fasciculus medicinae published in Venice 1491

First printed illustrated anatomy book by Vesalius “De Humani Corporis Fabrica” 1543

Johannes de Ketham,
Fasiculo de medicina. 1494.
National Library of Medicine.

Andreas Vesalius; *De Humani Corporis Fabrica.Basel*, 1543. Woodcut. National Library of Medicine.

Eduard Gröller and Katja Büher



Medical Images – Mixing Art and Science

Renaissance and Enlightenment (1430- early18th century)

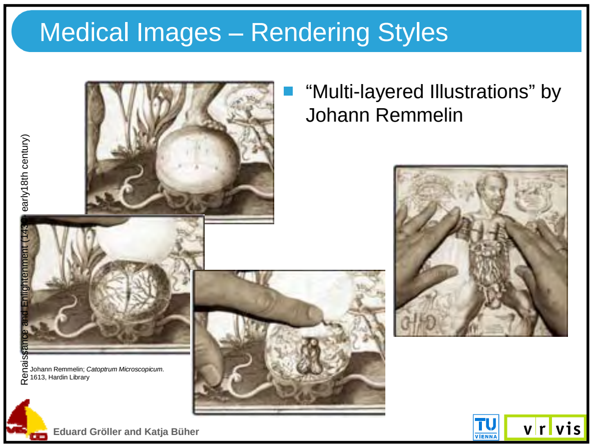
- Mixture of art and scientific illustration:
  - Subjective interpretation
  - Anatomical drawings tell stories

Bernhard Seigfried Albinus. *Tabulae sceleti et musculorum corporis humani*, 1749 (NLM)

Juan Valverde de Amusco; *Anatomia del corpo humano.* Rome, 1560. (NLM)

Fredrik van Ruysch; *Alle de ontleed- genees- en heelkindige werken. . . . Vol. 3 Amsterdam, 1744. Etching with engraving. (NLM - National Library of Medicine.)*

Eduard Gröller and Katja Büher



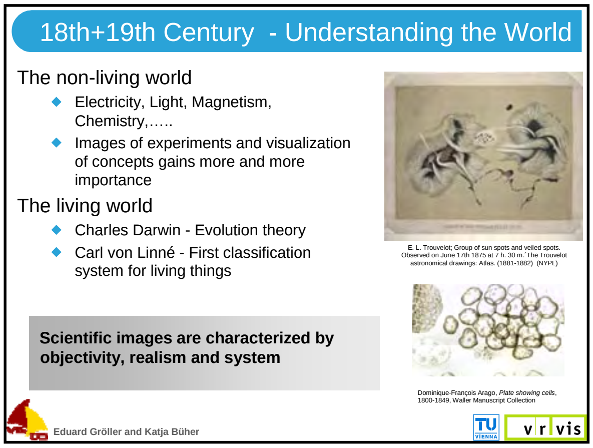
Medical Images – Rendering Styles

- “Multi-layered Illustrations” by Johann Remmelin

Renaissance and Enlightenment (1430- early18th century)

Johann Remmelin; *Catoptrum Microscopicum.* 1613, Hardin Library

Eduard Gröller and Katja Büher



18th+19th Century - Understanding the World

The non-living world

- Electricity, Light, Magnetism, Chemistry,.....
- Images of experiments and visualization of concepts gains more and more importance

The living world

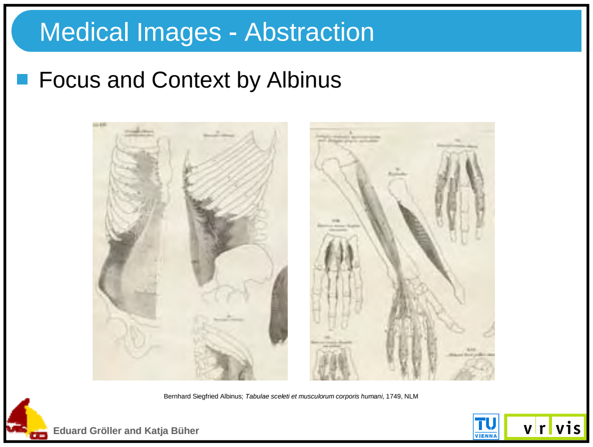
- Charles Darwin - Evolution theory
- Carl von Linné - First classification system for living things

Scientific images are characterized by objectivity, realism and system

E. L. Trouvelot; Group of sun spots and veiled spots. Observed on June 17th 1875 at 7 h. 30 m.´The Trouvelot astronomical drawings: Atlas. (1881-1882) (NYPL)

Dominique-François Arago, *Plate showing cells*, 1800-1849, Waller Manuscript Collection

Eduard Gröller and Katja Büher

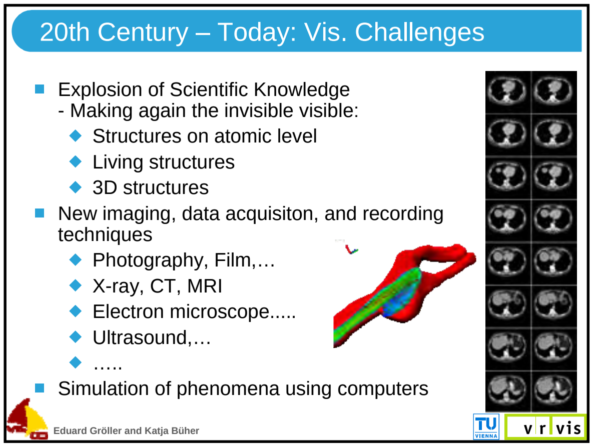


Medical Images - Abstraction

- Focus and Context by Albinus

Bernhard Siegfried Albinus; *Tabulae sceleti et musculorum corporis humani*, 1749, NLM

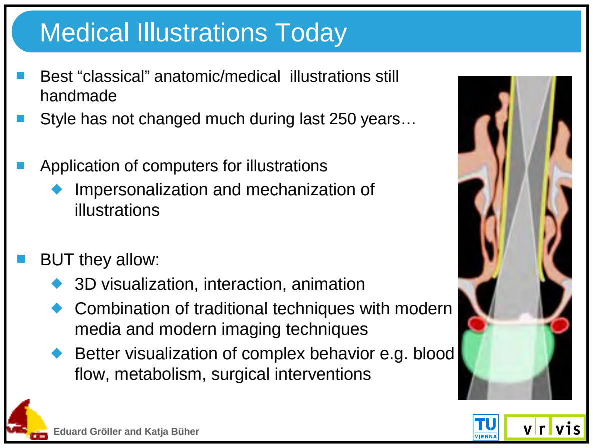
Eduard Gröller and Katja Büher



20th Century – Today: Vis. Challenges

- Explosion of Scientific Knowledge
  - Making again the invisible visible:
  - Structures on atomic level
  - Living structures
  - 3D structures
- New imaging, data acquisiton, and recording techniques
  - Photography, Film,…
  - X-ray, CT, MRI
  - Electron microscope.....
  - Ultrasound,…
  - …..
- Simulation of phenomena using computers

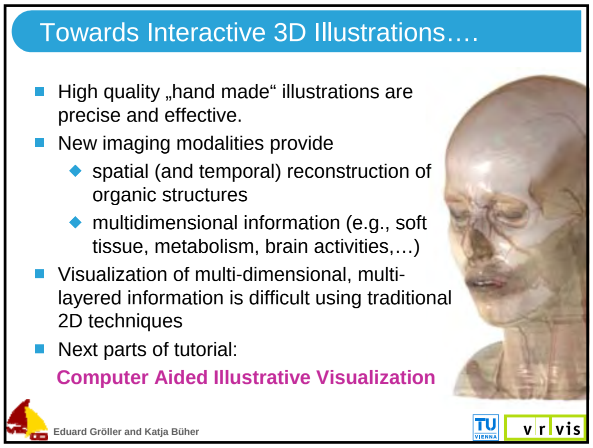
Eduard Gröller and Katja Büher



Medical Illustrations Today

- Best “classical” anatomic/medical illustrations still handmade
- Style has not changed much during last 250 years…
- Application of computers for illustrations
  - Impersonalization and mechanization of illustrations
- BUT they allow:
  - 3D visualization, interaction, animation
  - Combination of traditional techniques with modern media and modern imaging techniques
  - Better visualization of complex behavior e.g. blood flow, metabolism, surgical interventions

Eduard Gröller and Katja Büher



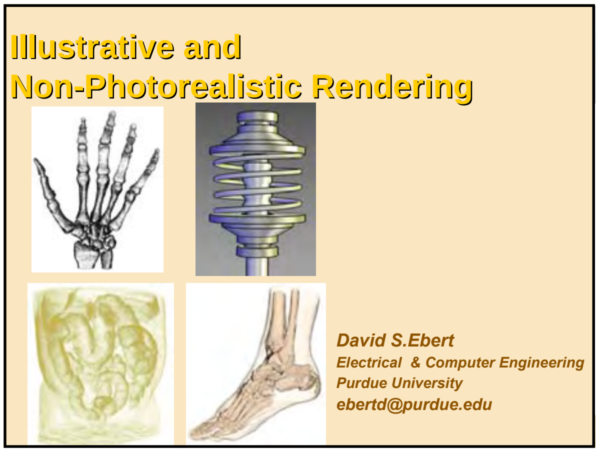
Towards Interactive 3D Illustrations….

- High quality „hand made“ illustrations are precise and effective.
- New imaging modalities provide
  - spatial (and temporal) reconstruction of organic structures
  - multidimensional information (e.g., soft tissue, metabolism, brain activities,…)
- Visualization of multi-dimensional, multi-layered information is difficult using traditional 2D techniques
- Next parts of tutorial:

Computer Aided Illustrative Visualization

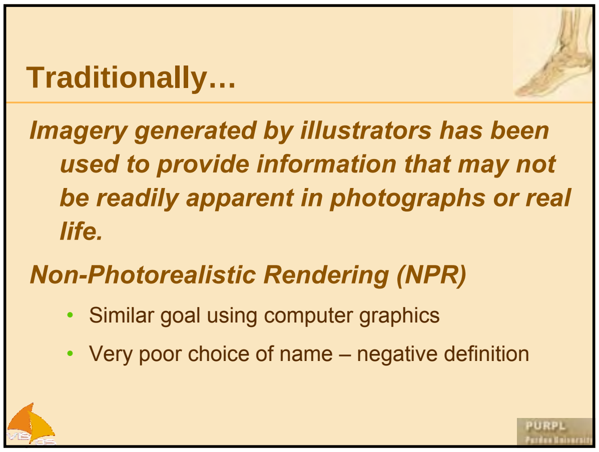
Eduard Gröller and Katja Büher

Illustrative and Non-Photorealistic Rendering



Illustrative and Non-Photorealistic Rendering

David S.Ebert
Electrical & Computer Engineering
Purdue University
ebertd@purdue.edu

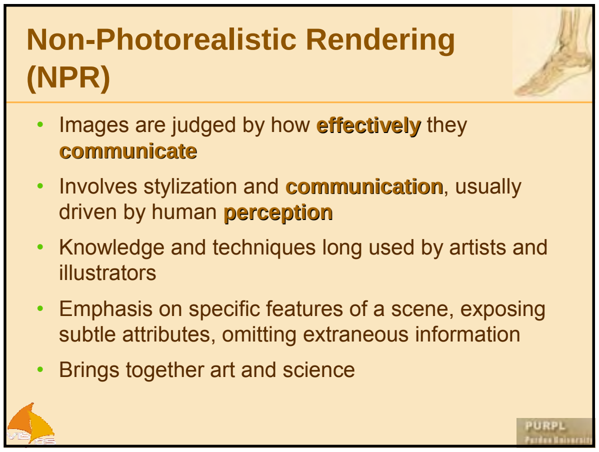


Traditionally…

Imagery generated by illustrators has been used to provide information that may not be readily apparent in photographs or real life.

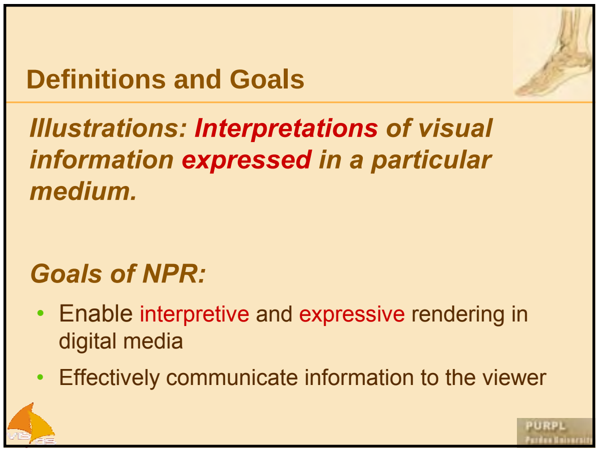
Non-Photorealistic Rendering (NPR)

- Similar goal using computer graphics
- Very poor choice of name – negative definition



Non-Photorealistic Rendering (NPR)

- Images are judged by how **effectively** they **communicate**
- Involves stylization and **communication**, usually driven by human **perception**
- Knowledge and techniques long used by artists and illustrators
- Emphasis on specific features of a scene, exposing subtle attributes, omitting extraneous information
- Brings together art and science

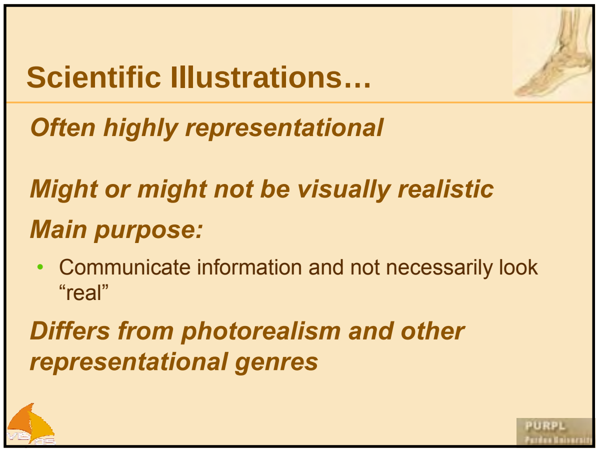


Definitions and Goals

Illustrations: Interpretations of visual information expressed in a particular medium.

Goals of NPR:

- Enable interpretive and expressive rendering in digital media
- Effectively communicate information to the viewer



Scientific Illustrations…

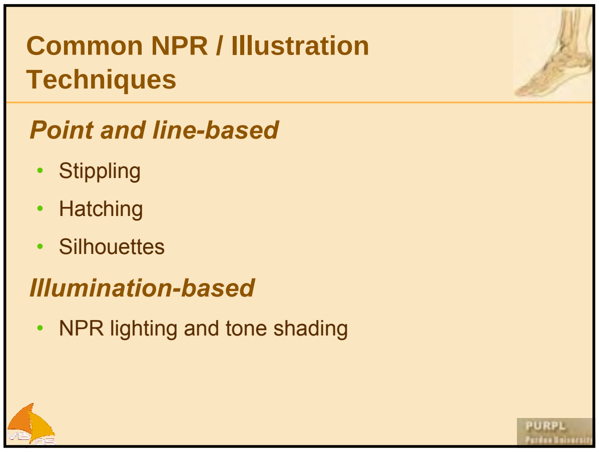
Often highly representational

Might or might not be visually realistic

Main purpose:

- Communicate information and not necessarily look “real”

Differs from photorealism and other representational genres



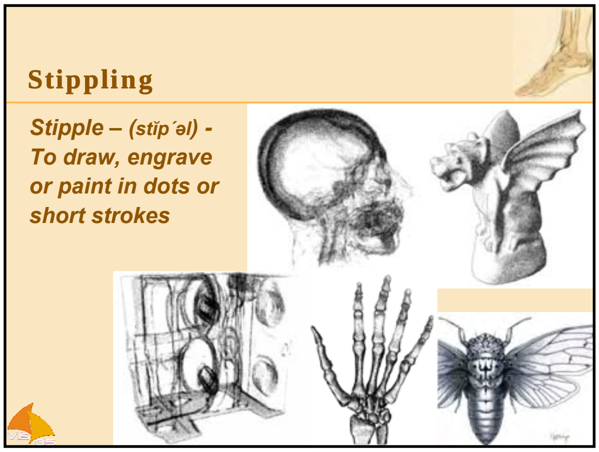
Common NPR / Illustration Techniques

Point and line-based

- Stippling
- Hatching
- Silhouettes

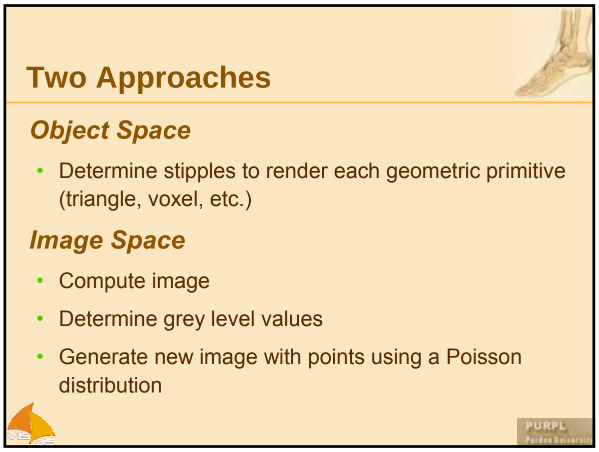
Illumination-based

- NPR lighting and tone shading



Stippling

Stipple – (stĭp´əl) - To draw, engrave or paint in dots or short strokes



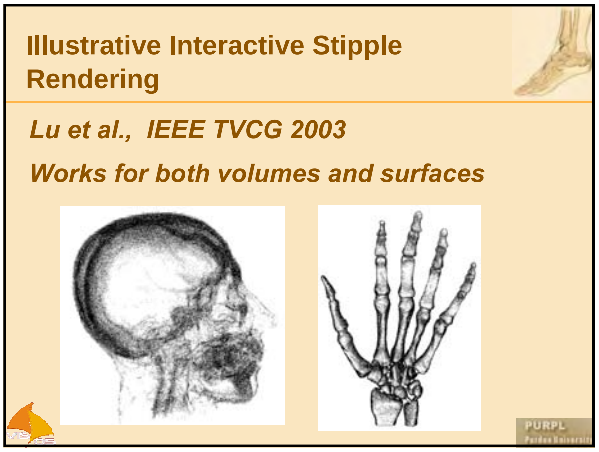
Two Approaches

Object Space

- Determine stipples to render each geometric primitive (triangle, voxel, etc.)

Image Space

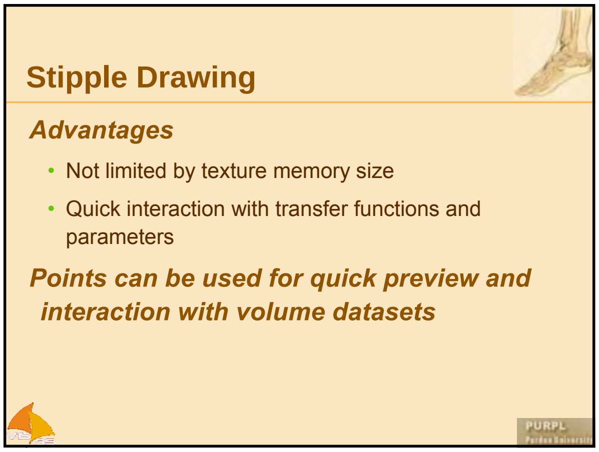
- Compute image
- Determine grey level values
- Generate new image with points using a Poisson distribution



Illustrative Interactive Stipple Rendering

Lu et al., IEEE TVCG 2003

Works for both volumes and surfaces

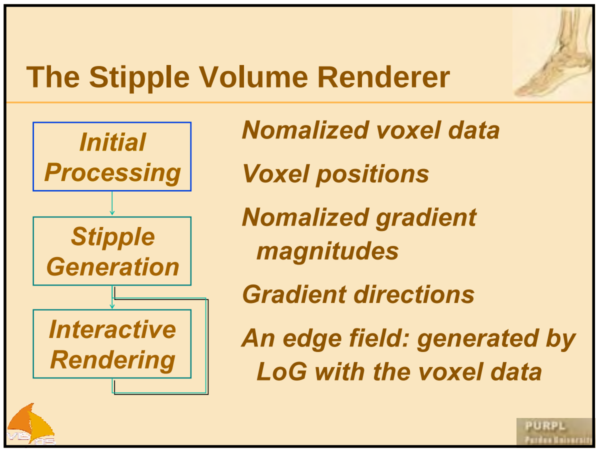


Stipple Drawing

Advantages

- Not limited by texture memory size
- Quick interaction with transfer functions and parameters

Points can be used for quick preview and interaction with volume datasets



The Stipple Volume Renderer

Initial Processing

Stipple Generation

Interactive Rendering

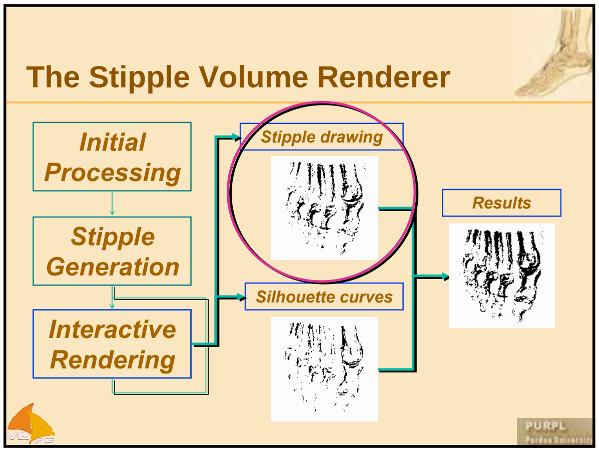
Nomalized voxel data

Voxel positions

Nomalized gradient magnitudes

Gradient directions

An edge field: generated by LoG with the voxel data



The Stipple Volume Renderer

Initial Processing

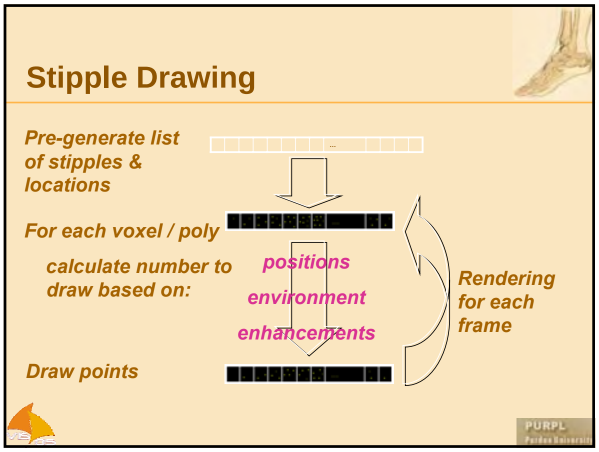
Stipple Generation

Interactive Rendering

Stipple drawing

Silhouette curves

Results



Stipple Drawing

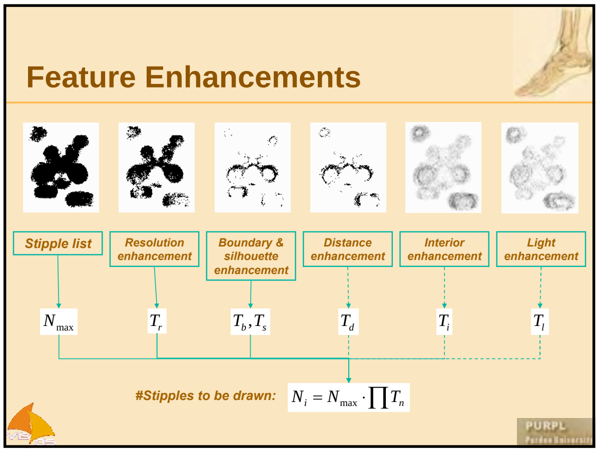
Pre-generate list of stipples & locations

For each voxel / poly

calculate number to draw based on: positions, environment, enhancements

Draw points

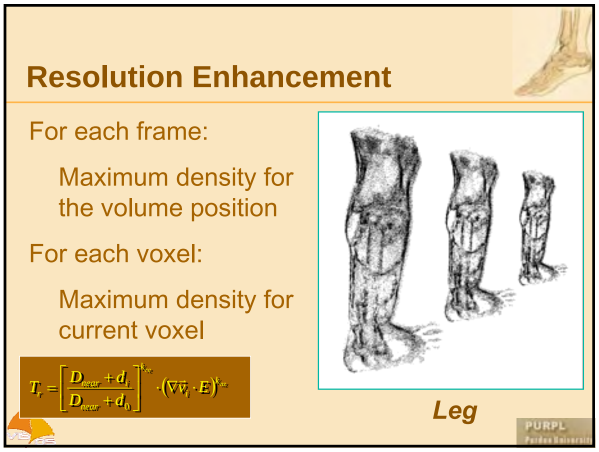
Rendering for each frame



Feature Enhancements

Stipple list	Resolution enhancement	Boundary & silhouette enhancement	Distance enhancement	Interior enhancement	Light enhancement
$N_{\max}$	T_r	T_b, T_s	T_d	T_i	T_l

#Stipples to be drawn: $N_i = N_{\max} \cdot \prod T_n$



Resolution Enhancement

For each frame:

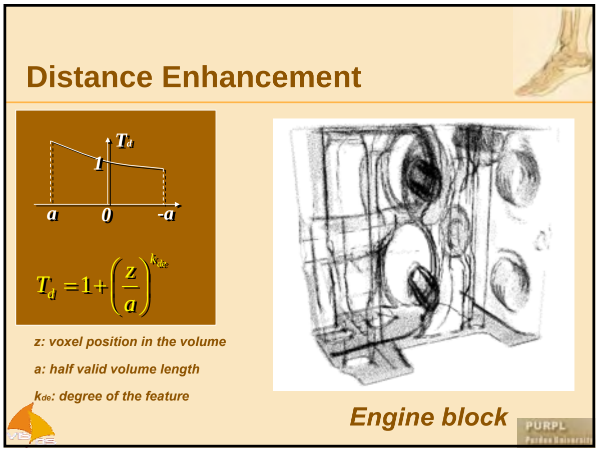
Maximum density for the volume position

For each voxel:

Maximum density for current voxel

$$T_r = \left[\frac{D_{near} + d_i}{D_{near} + d_0}\right]^{k_{re}} \cdot (\nabla \vec{v}_i \cdot E)^{k_{nz}}$$

Leg



Distance Enhancement

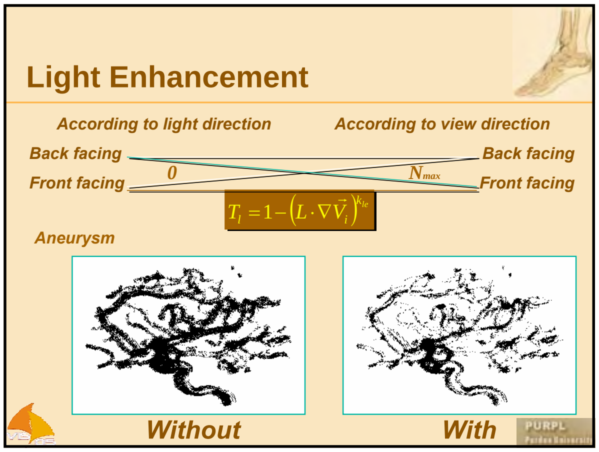
$$T_d = 1 + \left(\frac{z}{a}\right)^{k_{de}}$$

z: voxel position in the volume

a: half valid volume length

k$_{de}$*: degree of the feature*

Engine block



Light Enhancement

According to light direction — *According to view direction*

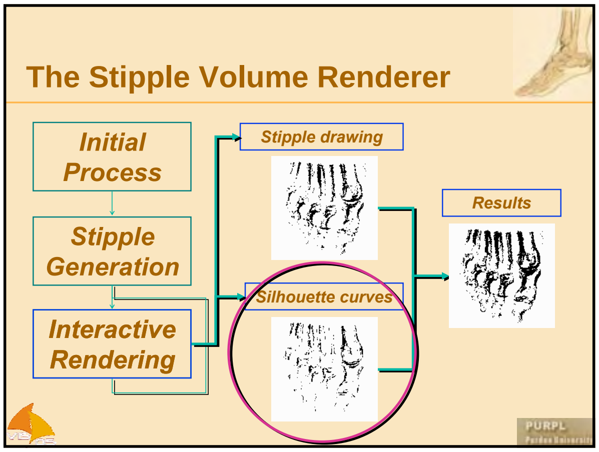
Back facing — *Back facing*

Front facing — *Front facing*

$$T_l = 1 - \left(L \cdot \nabla \vec{V}_i\right)^{k_{le}}$$

Aneurysm

Without *With*



The Stipple Volume Renderer

Initial Process

Stipple Generation

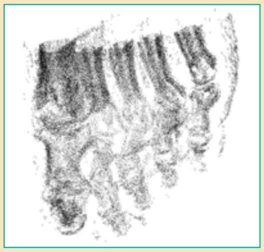
Interactive Rendering

Stipple drawing

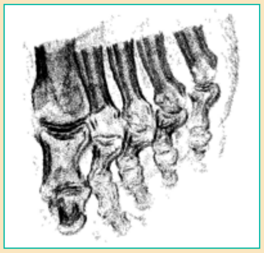
Silhouette curves

Results

Silhouette Curves

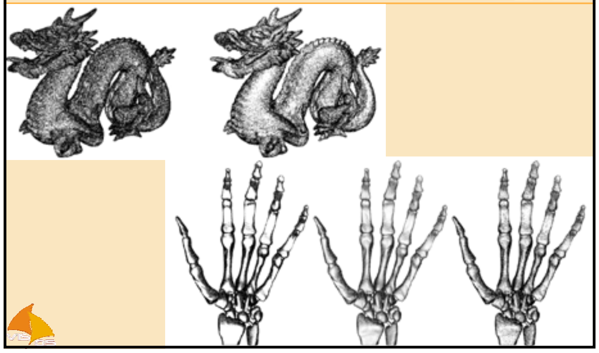


Without



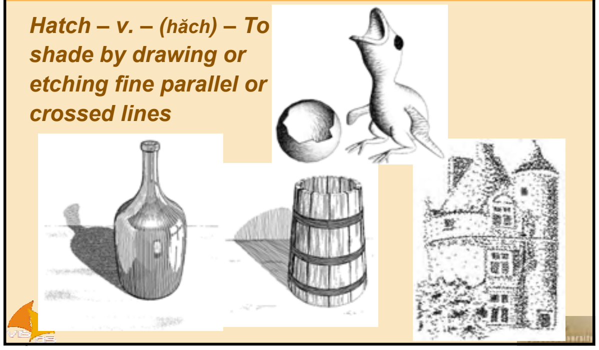
With

Polygonal Results



Hatching

Hatch – v. – (hăch) – To shade by drawing or etching fine parallel or crossed lines



Object Space Hatching

Computer-Generated Pen-and-Ink Illustration (Winkenbach and D. H. Salesin -SIGGRAPH 94)

Apply hatching patterns directly to the 3D geometry

Introduced the concept of stroke textures

- Allow resolution dependent rendering.

Emphasizes tone and texture

- Preserved across resolutions

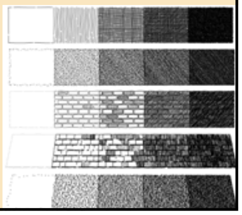
Ensures shadowed areas are shaded consistently with light position, surface orientation, …

Prioritized Stroke Textures

Precompute a texture covered by many strokes

To render

- Use several textures, each with an associated priority
- Render from high to low priority until the appropriate level of grey is achieved



Results

Frank Lloyd Wright's "Robie House"

Roughly consists of ~1000 polygons

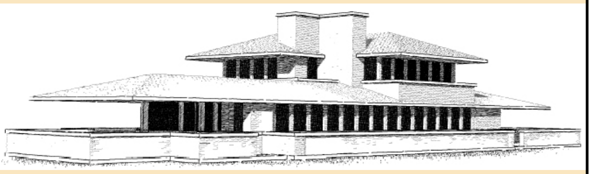


Image-Based Hatching

Salisbury et al. SIGGRAPH '97

Hatching patterns are placed on image using orientable textures

User interactively edits direction field superimposed on a grey-scale image and draws a few sample strokes

Align the direction field with the curvatures and orientations of the object

- Hatching appears to be attached to the object

No geometric information required

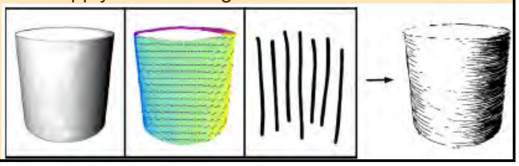
Target Images and Direction Fields

Grey-scale target image

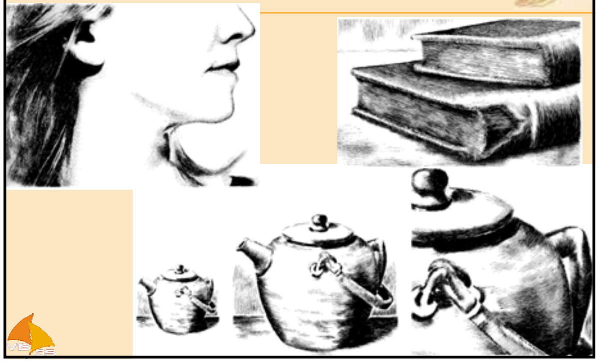
- Allows interactively changing the shading (tone)

Direction field

- Interactively modifiable
- Used to apply the hatching texture



Some Results



Real-Time Hatching

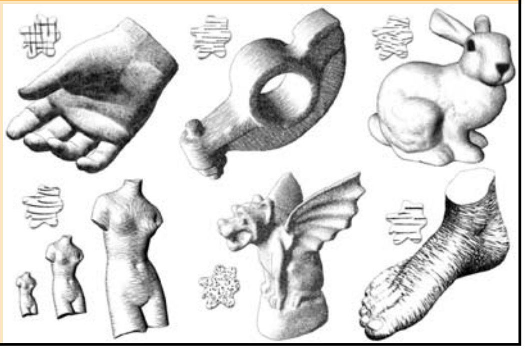
Praun, Hoppe, et al.

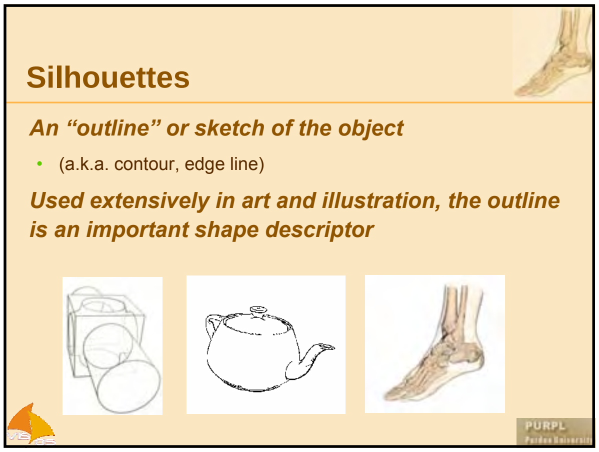
Applies a hatching pattern in object-space using Tonal Art Maps (TAMs) and lapped textures

Uses multi-texturing graphics hardware

- Smoothly blends several hatching image textures with several different stroke densities for shading

Results



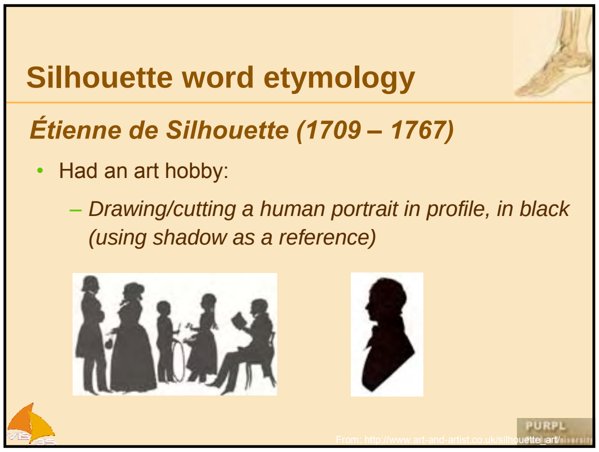


Silhouettes

An "outline" or sketch of the object

- (a.k.a. contour, edge line)

Used extensively in art and illustration, the outline is an important shape descriptor



Silhouette word etymology

Étienne de Silhouette (1709 – 1767)

- Had an art hobby:
  - *Drawing/cutting a human portrait in profile, in black (using shadow as a reference)*

From: http://www.art-and-artist.co.uk/silhouette_art/

Silhouette Approach Classification

Image-space vs. Object-space

Polygonal vs. Smooth

Surfaces vs. Volumes

Software vs. Hardware

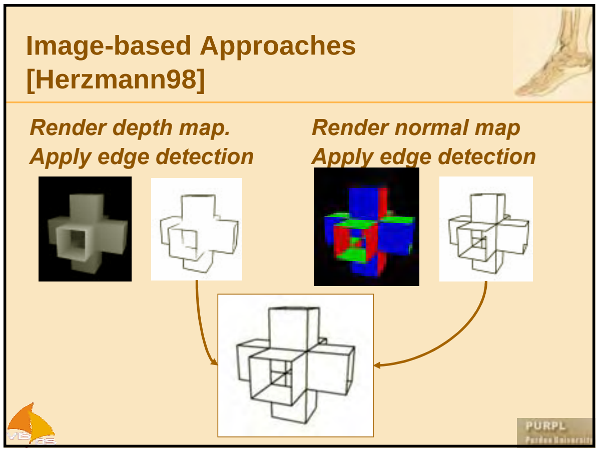
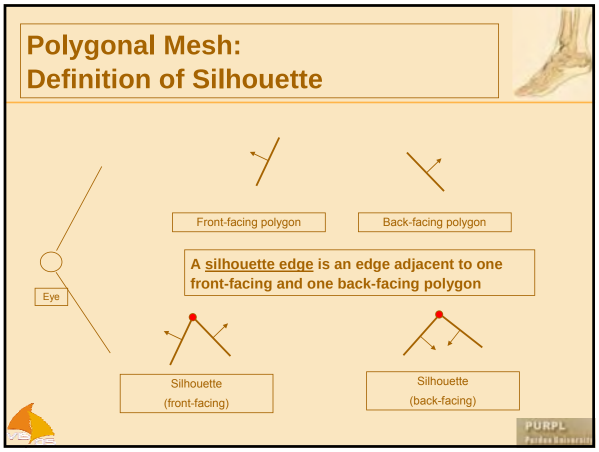


Image-based Approaches [Herzmann98]

Render depth map. Apply edge detection

Render normal map Apply edge detection



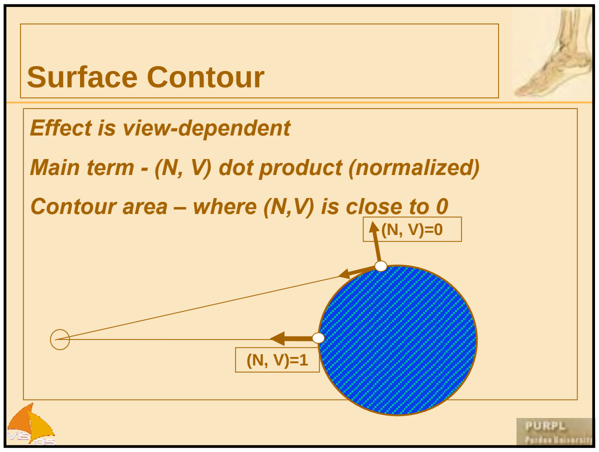
Polygonal Mesh: Definition of Silhouette

A <u>silhouette edge</u> is an edge adjacent to one front-facing and one back-facing polygon

Smooth Surface: Definition of silhouette

Silhouette and contour curves are the 2D projection of points on the 3D surface where the **direction of the surface normal is orthogonal to the line of sight** [Interrante95, Herzmann98]

- *Silhouette* ***curves form a closed outline around the projection***
- *Contour* ***curves may be disjoint and can fall within the projective boundary***



Surface Contour

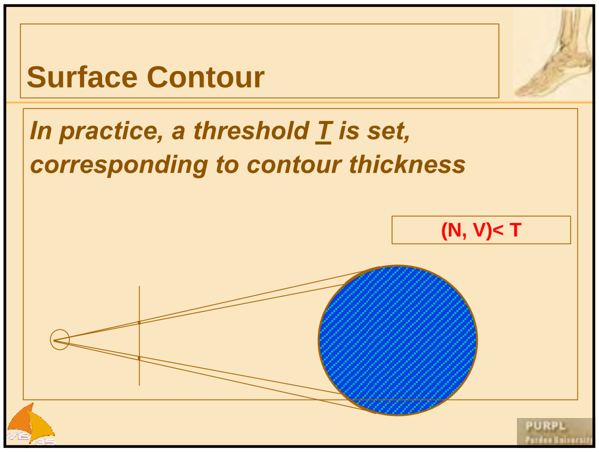
Effect is view-dependent

Main term - (N, V) dot product (normalized)

Contour area – where (N,V) is close to 0

(N, V)=0

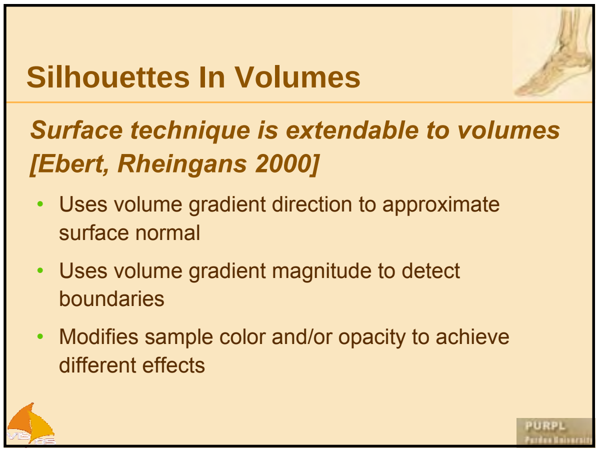
(N, V)=1



Surface Contour

In practice, a threshold T is set, corresponding to contour thickness

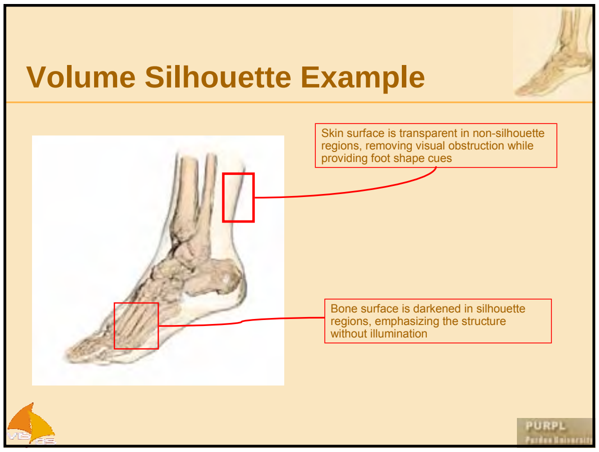
(N, V)< T



Silhouettes In Volumes

Surface technique is extendable to volumes [Ebert, Rheingans 2000]

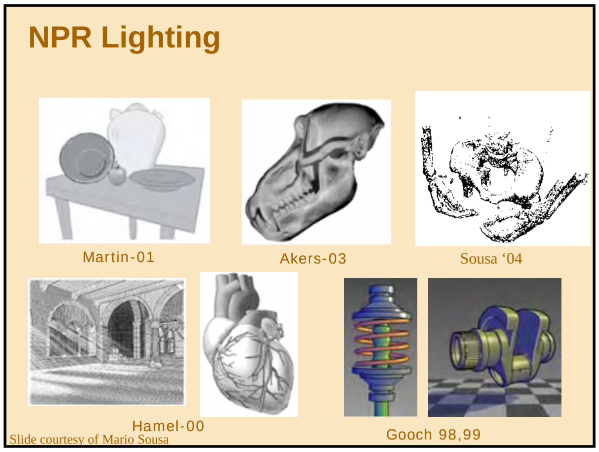
- Uses volume gradient direction to approximate surface normal
- Uses volume gradient magnitude to detect boundaries
- Modifies sample color and/or opacity to achieve different effects



Volume Silhouette Example

Skin surface is transparent in non-silhouette regions, removing visual obstruction while providing foot shape cues

Bone surface is darkened in silhouette regions, emphasizing the structure without illumination



NPR Lighting

Martin-01

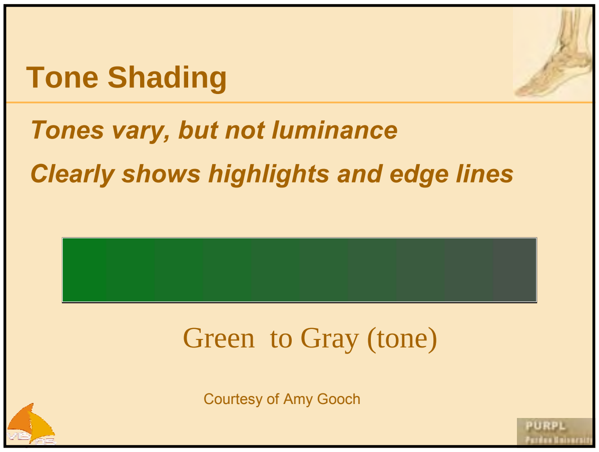
Akers-03

Sousa '04

Hamel-00

Gooch 98,99

Slide courtesy of Mario Sousa



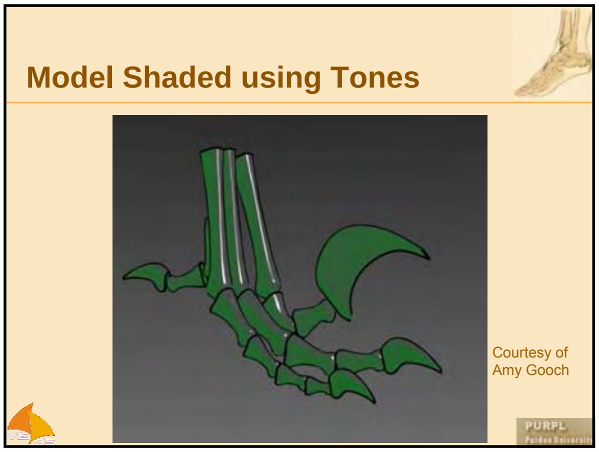
Tone Shading

Tones vary, but not luminance

Clearly shows highlights and edge lines

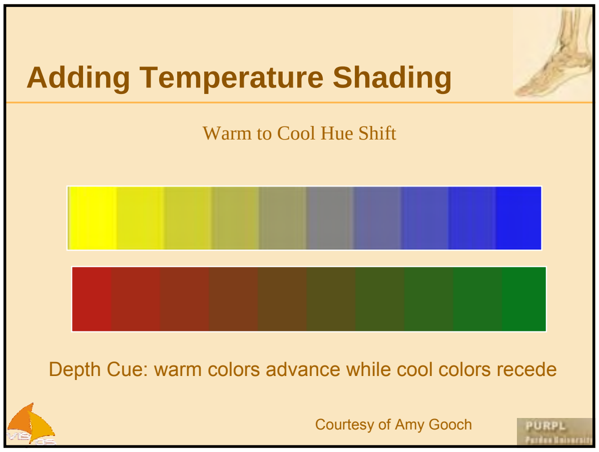
Green to Gray (tone)

Courtesy of Amy Gooch



Model Shaded using Tones

Courtesy of Amy Gooch

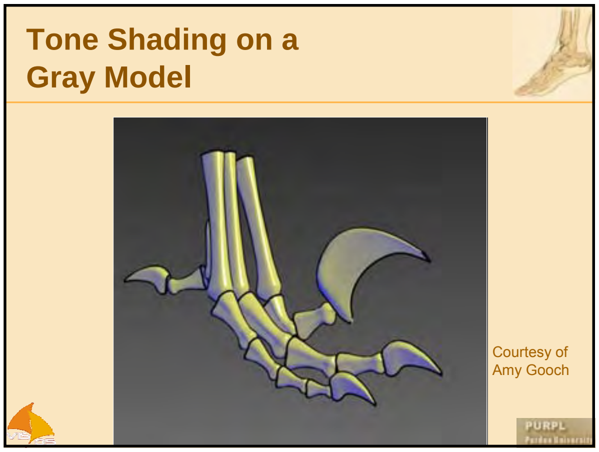


Adding Temperature Shading

Warm to Cool Hue Shift

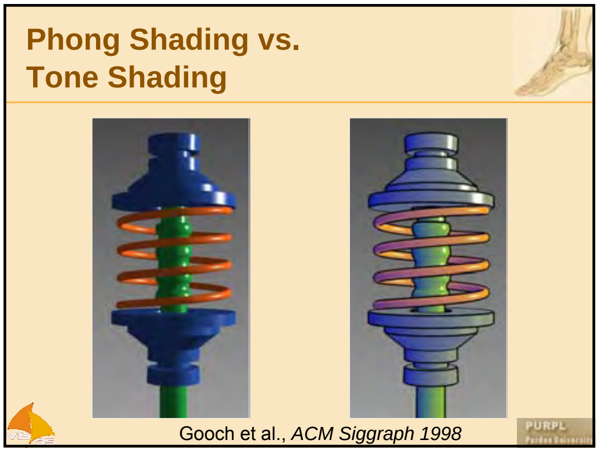
Depth Cue: warm colors advance while cool colors recede

Courtesy of Amy Gooch



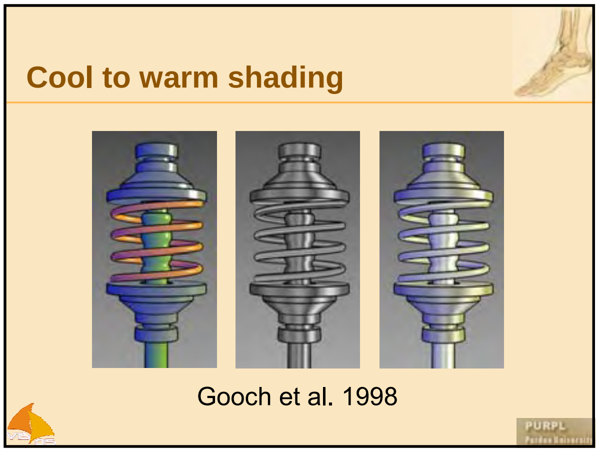
Tone Shading on a Gray Model

Courtesy of Amy Gooch



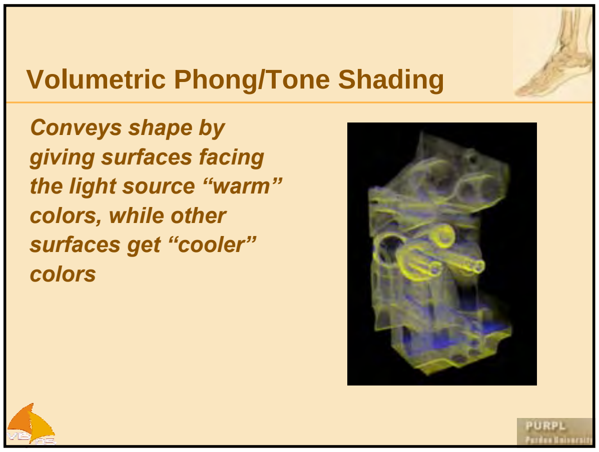
Phong Shading vs. Tone Shading

Gooch et al., *ACM Siggraph 1998*



Cool to warm shading

Gooch et al. 1998



Volumetric Phong/Tone Shading

Conveys shape by giving surfaces facing the light source "warm" colors, while other surfaces get "cooler" colors

Illustrative Visualization of Isosurfaces and Volumes

Tutorial Notes: Illustrative Visualization of Isosurfaces and Volumes

Markus Hadwiger*
VRVis Research Center
Vienna, Austria

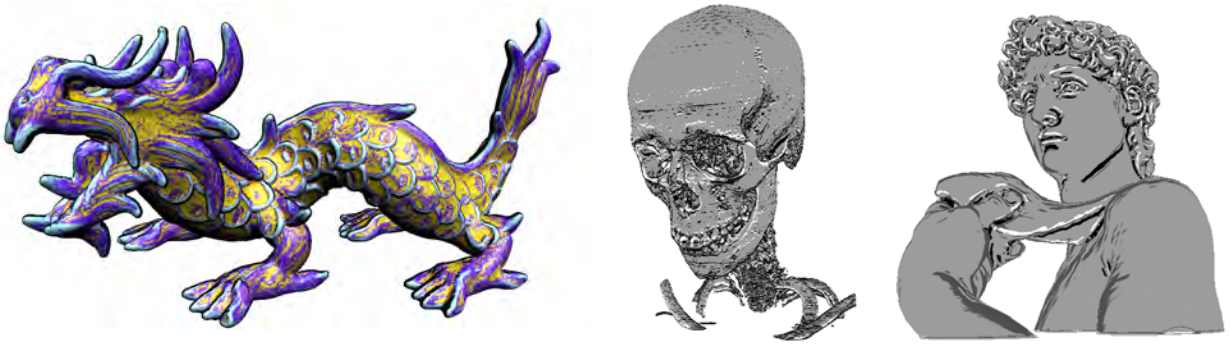


Figure 1: Illustrating the shape of an isosurface of a three-dimensional distance field with curvature color coding (left), and drawing shape cues such as ridge and valley lines and contours on an isosurface of a CT scan (center) and another distance field (right). There is no explicit geometry here. All isosurfaces are rendered directly from the underlying volumetric representation. Here, GPU ray-casting [8] has been used.

Abstract

This part of the tutorial on *Illustrative Visualization* describes visualization techniques for depicting isosurfaces from volumes without extracting explicit geometry, and full volume rendering with non-photorealistic styles for different embedded objects. We start by describing how isosurfaces can be shaded based on differential surface properties that are computed on a per-pixel basis, employing a deferred shading approach. This information can then be used for depicting a variety of different shape cues such as color-coding implicit surface curvature and rendering ridge and valley lines. We illustrate a GPU-based rendering pipeline for high-quality rendering of isosurfaces with real-time curvature computation and shading. After describing surface-based illustration styles we continue with full volume rendering. We show that segmentation information is an especially powerful tool for depicting the objects contained in medical data sets in varying styles. A combination of non-photorealistic styles with standard direct volume rendering is a very effective means for separating focus from context objects or regions. We describe the concept of two-level volume rendering that integrates different rendering modes and compositing types by using segmented data and per-object attributes.

1 Isosurface Illustration With Deferred Shading

Many non-photorealistic volume rendering techniques operate on isosurfaces of volumetric data. Although direct volume rendering as well as other techniques aiming to depict an entire volume in a single image are very important and popular, rendering isosurfaces corresponding to particular structures of interest, or more precisely, their boundaries, play a very important role in the field of volume rendering. Isosurfaces naturally allow depicting their structure with surface-based shape cues such as ridge and valley lines and contours, such as the distance field isosurface shown in Figure 1.

There are two major approaches for rendering isosurfaces of volume data. First, an explicit triangle mesh corresponding to a given iso-value can be extracted prior to rendering, e.g., using marching cubes [23] or one of its variants [17]. Second, ray-isosurface intersections can be determined via ray casting [1, 22]. Naturally, general NPR techniques for rendering surfaces can easily be applied to rendering isosurfaces of volume data.

In hardware-accelerated volume rendering, isosurfaces have traditionally been rendered by slicing the volume in back-to-front order and exploiting the hardware alpha test in order to reject fragments not corresponding to the isosurface [34]. The concept of pre-integration can also be applied to isosurface rendering, which yields results of high quality even with low sampling rates [5]. Recently, GPU-based ray casting approaches have been developed [18, 28], which can also be used to determine ray-isosurface intersections.

The following sections illustrate a high-quality rendering pipeline for direct rendering of isosurfaces by determining ray-isosurface intersections and subsequent deferred shading of the corresponding pixels. The input to the deferred shading stages is a floating point image of ray-isosurface intersection positions, which is obtained from either slicing the volume [7], illustrated in figure 4, or first hit ray casting that stores hit positions into the target buffer using a GPU ray casting method [8].

1.1 Deferred Shading

In standard rendering pipelines, shading equations are often evaluated for pixels that are entirely invisible or whose contribution to the final image is negligible. With the shading equations used in real-time rendering becoming more and more complex, avoiding these computations for invisible pixels becomes an important goal.

A very powerful concept that allows to compute shading only for actually visible pixels is the notion of *deferred shading* [3, 21]. Deferred shading computations are usually driven by one or more input images that contain all the information that is necessary for performing the final shading of the corresponding pixels. Especially in the context of non-photorealistic rendering, these input images are often also called *G-buffers* [29]. The major advantage of the concept of deferred computations is that it reduces their complexity from being proportional to object space, e.g., the number of vox-

*msh@vrvis.at

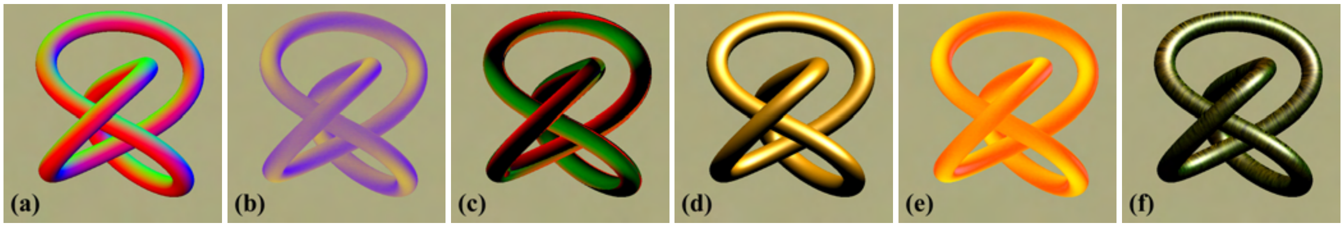


Figure 2: Example image space rendering passes of deferred isosurface shading. Surface properties such as (a) the gradient (here color-coded in RGB), (b) principal curvature magnitudes (here: κ_1), and (c) principal curvature directions can be reconstructed. These properties can be used in shading passes, e.g., (d) Blinn-Phong shading, (e) color coding of curvature measures (here: $\sqrt{\kappa_1^2+\kappa_2^2}$ [15]), and (f) advection of flow along principal curvature directions.

els in a volume, to the complexity of image space, i.e., the number of pixels in the final output image. Naturally, these computations are not limited to shading equations per se, but can also include the derivation of additional information that is only needed for visible pixels and may be required as input for shading, such as differential surface properties.

In this section, we describe deferred shading computations for rendering isosurfaces of volumetric data. The computations that are deferred to image space are not limited to actual shading, but also include the derivation of differential implicit surface properties such as the gradient (first order partial derivatives), the Hessian matrix (second order partial derivatives), and principal curvature information.

Figure 5 illustrates a pipeline for deferred shading of isosurfaces of volume data, and figure 2 shows example images corresponding to the output of specific image space rendering passes. The input to the pipeline is a single floating point image storing ray-surface intersection positions of the viewing rays and the isosurface. This image is obtained via either slicing the volume, or first hit ray casting, as outlined above and illustrated in figure 4.

From this intersection position image, differential isosurface properties such as the gradient and additional partial derivatives such as the Hessian matrix can be computed first. This allows shading with high-quality gradients, as well as computation of high-quality principal curvature magnitude and direction information. Sections 1.2 and 1.3 describe high-quality reconstruction of differential isosurface properties.

In the final actual shading pass, differential surface properties can be used for shading computations such as Blinn-Phong shading, color mapping of curvature magnitudes, and flow advection along curvature directions, as well as applying a solid texture onto the isosurface.

Shading from gradient image

The simplest shading equations depend on the normal vector of the isosurface, i.e., its normalized gradient. The normal vector can for example be used to compute Blinn-Phong shading, and reflection and refraction mapping that index an environment map with vectors computed from the view vector and the normal vector.

Solid texturing

The initial position image that contains ray-isosurface intersection positions can be used for straight-forward application of a solid texture onto an isosurface. Parameterization is simply done by specifying the transformation of object space to texture space coordinates, e.g., via an affine transformation. For solid texturing, real-time tricubic filtering can be used instead of tri-linear interpolation in order to achieve high-quality results.

1.2 Deferred Gradient Reconstruction

The most important differential property of the isosurface that needs to be reconstructed is the gradient of the underlying scalar field f:

$$\mathbf{g} = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right)^{\mathbf{T}} \tag{1}$$

The gradient can then be used as implicit surface normal for shading and curvature computations.

The surface normal is the normalized gradient of the volume, or its negative, depending on the notion of being inside/outside of the object that is bounded by the isosurface: $\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$. The calculated gradient can be stored in a single RGB floating point image, see figure 2(a).

Hardware-accelerated high-quality filtering can be used for reconstruction of high-quality gradients by convolving the original scalar volume three times with the first derivative of a reconstruction kernel, e.g., the derived cubic B-spline kernel that is shown in figure 3(a).

The quality difference between cubic filtering and linear interpolation is even more apparent in gradient reconstruction than it is in value reconstruction. Figure 6 shows a comparison of different combinations of filters for value and gradient reconstruction, i.e., linear interpolation and cubic reconstruction with a cubic B-spline kernel. Figure 7 compares linear and cubic (B-spline) reconstruction using reflection mapping and a line pattern environment map. Reconstruction with the cubic B-spline achieves results with C^2 continuity.

1.3 Other Differential Properties

In addition to the gradient of the scalar volume, i.e., its first partial derivatives, further differential properties can be reconstructed in additional deferred shading passes.

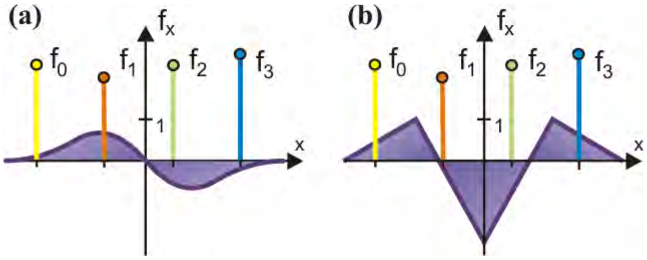


Figure 3: The first order (a) and second order (b) derivatives of the cubic B-spline filter for direct high-quality reconstruction of derivatives via convolution.

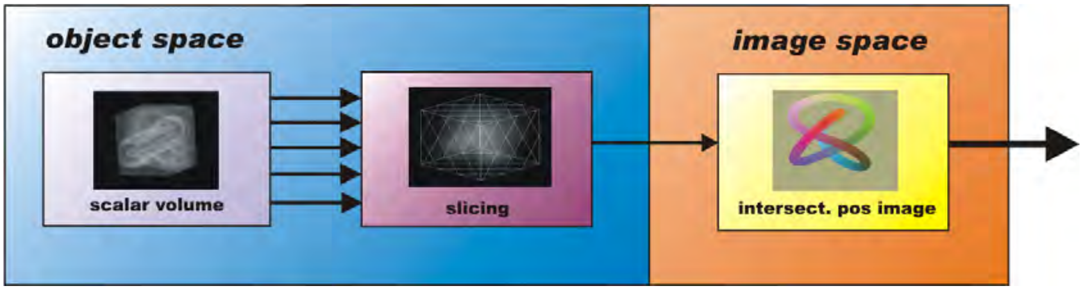


Figure 4: Slicing a volume in order to determine an intersection image of ray-isosurface intersections for deferred shading of an isosurface in subsequent image space rendering passes.

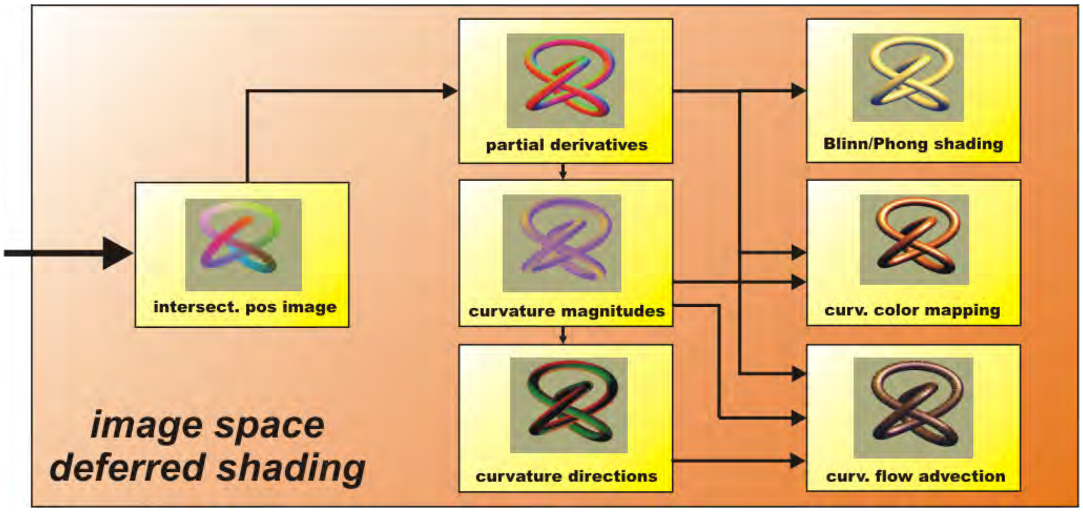


Figure 5: Deferred shading computations for an isosurface given as a floating point image of ray-surface intersection positions. First, differential properties such as the gradient and additional partial derivatives can be computed. These derivatives also allow to compute principal curvature information on-the-fly. In the final shading pass, the obtained properties can be used for high-quality shading computations. All of these computations and shading operations have image space instead of object space complexity and are only performed for visible pixels.

For example, implicit principal curvature information can be computed from the second order partial derivatives of the volume. Curvature has many applications in surface investigation and rendering, e.g., non-photorealistic rendering equations incorporating curvature magnitudes in order to detect surface structures such as ridge and valley lines, or rendering silhouettes of constant screen space thickness.

Second order partial derivatives: the Hessian

The Hessian matrix **H** is comprised of all second order partial derivatives of the scalar volume f:

$$\mathbf{H} = \nabla \mathbf{g} = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{pmatrix} \quad (2)$$

Due to symmetry, only six unique components need to be calculated, which can be stored in two RGB floating point images.

High-quality second order partial derivatives can be computed by convolving the scalar volume with a combination of first and second order derivatives of the cubic B-spline kernel, for example, which is illustrated in figure 3.

Principal curvature magnitudes

The first and second principal curvature magnitudes (κ_1, κ_2) of the isosurface can be estimated directly from the gradient **g** and the Hessian **H** [15], whereby tri-cubic filtering in general yields high-quality results. This can be done in a single rendering pass, which uses the three partial derivative RGB floating point images generated by previous pipeline stages as input textures. It amounts to a moderate number of vector and matrix multiplications and solving a quadratic polynomial.

The result is a floating point image storing (κ_1, κ_2), which can then be used in the following passes for shading and optionally calculating curvature directions. See figure 2(b).

Principal curvature directions

The principal curvature magnitudes are the eigenvalues of a 2x2 eigensystem in the tangent plane specified by the normal vector, which can be solved in the next rendering pass for the corresponding eigenvectors, i.e., the 1D subspaces of principal curvature directions. Representative vectors for either the first or second principal directions can be computed in a single rendering pass.

The result is a floating point image storing principal curvature direction vectors. See Figure 2(c).

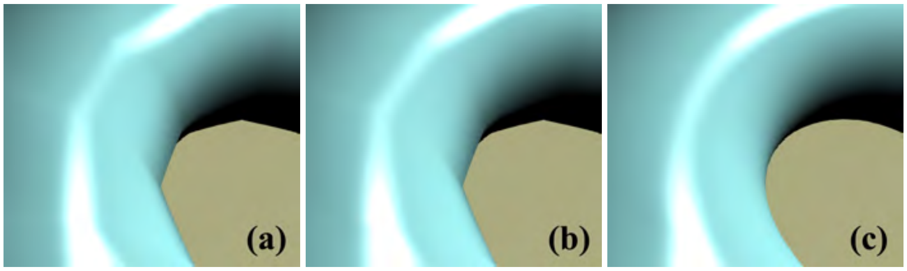


Figure 6: Linear and cubic filtering for value and gradient reconstruction on a torus: (a) both value and gradient are linear; (b) value is linear and gradient cubic; (c) both value and gradient are cubic. For cubic filtering, a cubic B-spline kernel has been used.

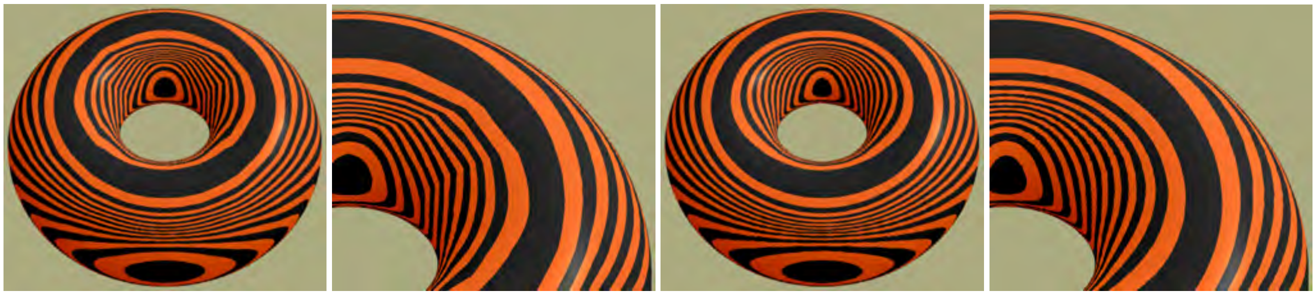


Figure 7: Comparing linear and cubic gradient reconstruction with a cubic B-spline filter using reflection lines. The left two images are with linear, the right two with cubic filtering.

Filter kernel considerations

All curvature reconstructions in this chapter employ a cubic B-spline filter kernel and its derivatives. It has been shown that cubic filters are the lowest order reconstruction kernels for obtaining high-quality curvature estimates. They also perform very well when compared with filters of even higher order [15].

The B-Spline filter is a good choice for curvature reconstruction because it is the only fourth order BC-spline filter which is both accurate and continuous for first and second derivatives [26, 15]. Hence it is the only filter of this class which reconstructs continuous curvature estimates. Cubic B-spline filters can be implemented very efficiently on current GPUs [30].

However, although B-spline filters produce smooth and visually pleasing results, they might be inappropriate in some applications where data interpolation is required [25]. Using a combination of the first and second derivatives of the cubic B-spline for derivative reconstruction, and a Catmull-Rom spline for value reconstruction is a viable alternative that avoids smoothing the original data [15].

1.4 Rendering from Implicit Curvature

Computing implicit surface curvature is a powerful tool for isosurface investigation and non-photorealistic rendering of isosurfaces.

When differential isosurface properties have been computed in preceding deferred shading passes (see section 1.3), this information can be used for performing a variety of mappings to shaded images in a final shading pass.

Curvature-based transfer functions

Principal curvature magnitudes can be visualized on an isosurface by mapping them to colors via one-dimensional or two-dimensional transfer function lookup textures.

One-dimensional curvature transfer functions. Simple color mappings of first or second principal curvature magnitude via 1D transfer function lookup tables can easily be computed during shading. The same approach can be used to depict additional curvature measures directly derived from the principal magnitudes, such as mean curvature $(\kappa_1 + \kappa_2)/2$ or Gaussian curvature $\kappa_1 \kappa_2$. See figures 10(left), 13(top, left), and 15(top, left) for examples.

Two-dimensional curvature transfer functions. Transfer functions in the 2D domain of both principal curvature magnitudes (κ_1, κ_2) are especially powerful, since color specification in this domain allows to highlight different structures on the surface [12], including ridge and valley lines [13, 15]. Curvature magnitude information can also be used to implement silhouette outlining with constant screen space thickness [15]. See figures 10, 11, 13, 14, and 15 for examples. Figure 8 illustrates 2D transfer functions in the domain of curvature measures for ridge and valley lines (left), and constant silhouette thickness (right).

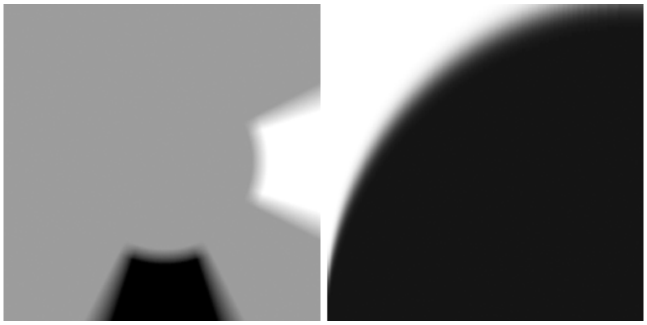


Figure 8: Two-dimensional transfer functions in curvature space. (left) Ridge and valley lines in (κ_1, κ_2) domain [15]; (right) contour thickness control [15].

Curvature-aligned flow advection

Direct mappings of principle curvature directions to RGB colors are hard to interpret, see figure 2(c), for example.

However, principal curvature directions on an isosurface can be visualized using image-based flow visualization [33]. In particular, flow can be advected on the surface entirely in image space [19, 20]. These methods can easily be used in real-time, complementing the capability to generate high-quality curvature information on-the-fly, which also yields the underlying, potentially unsteady, ”flow” field in real-time. See figure 2(f). In this case, it is natural to perform per-pixel advection guided by the floating point image containing principal direction vectors instead of warping mesh vertex or texture coordinates.

A problem with advecting flow along curvature directions is that their orientation is not uniquely defined and thus seams in the flow cannot be entirely avoided [33].

See figures 9 and 11(top, left) for examples.

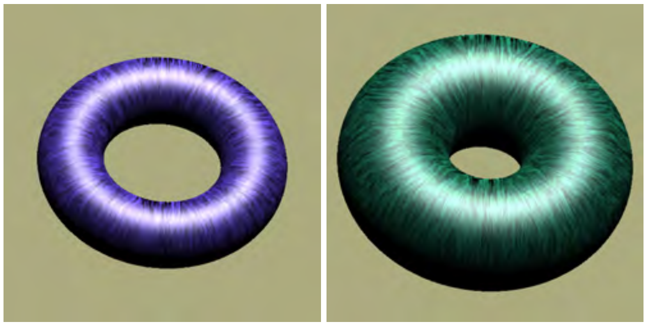


Figure 9: Changing the iso-value of a torus isosurface represented by a signed distance field. Color is derived from maximum principal curvature magnitude, and flow is advected in image space in maximum principal curvature direction. All changes to the iso-value take effect in real-time. Curvature directions constitute an unsteady flow field.

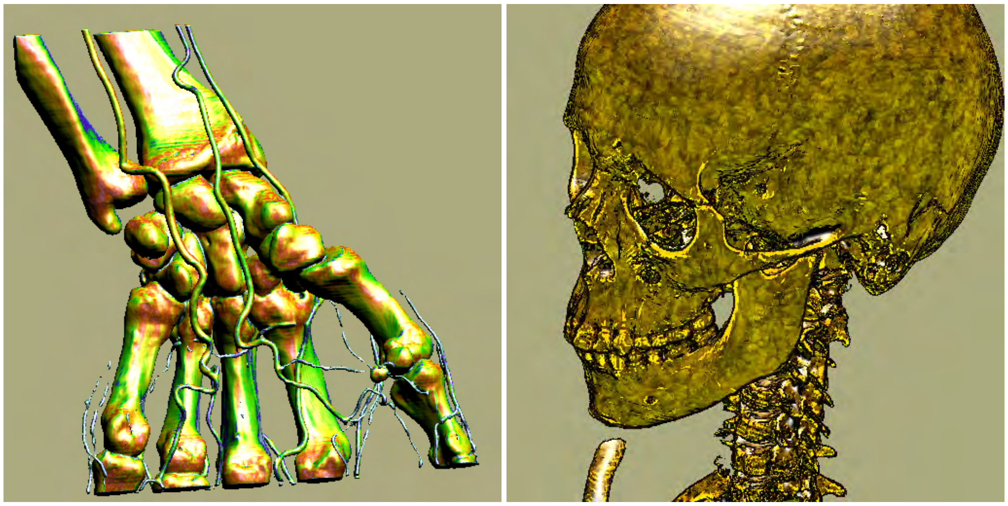


Figure 10: Two examples of implicit curvature-based isosurface rendering. (left) CT scan (256x128x256) with color mapping from a 1D transfer function depicting $\sqrt{\kappa_1^2+\kappa_2^2}$; (right) CT scan (256x256x333) with contours, ridges and valleys, tone shading, and principal curvature-aligned flow advection to generate a noise pattern on the surface.

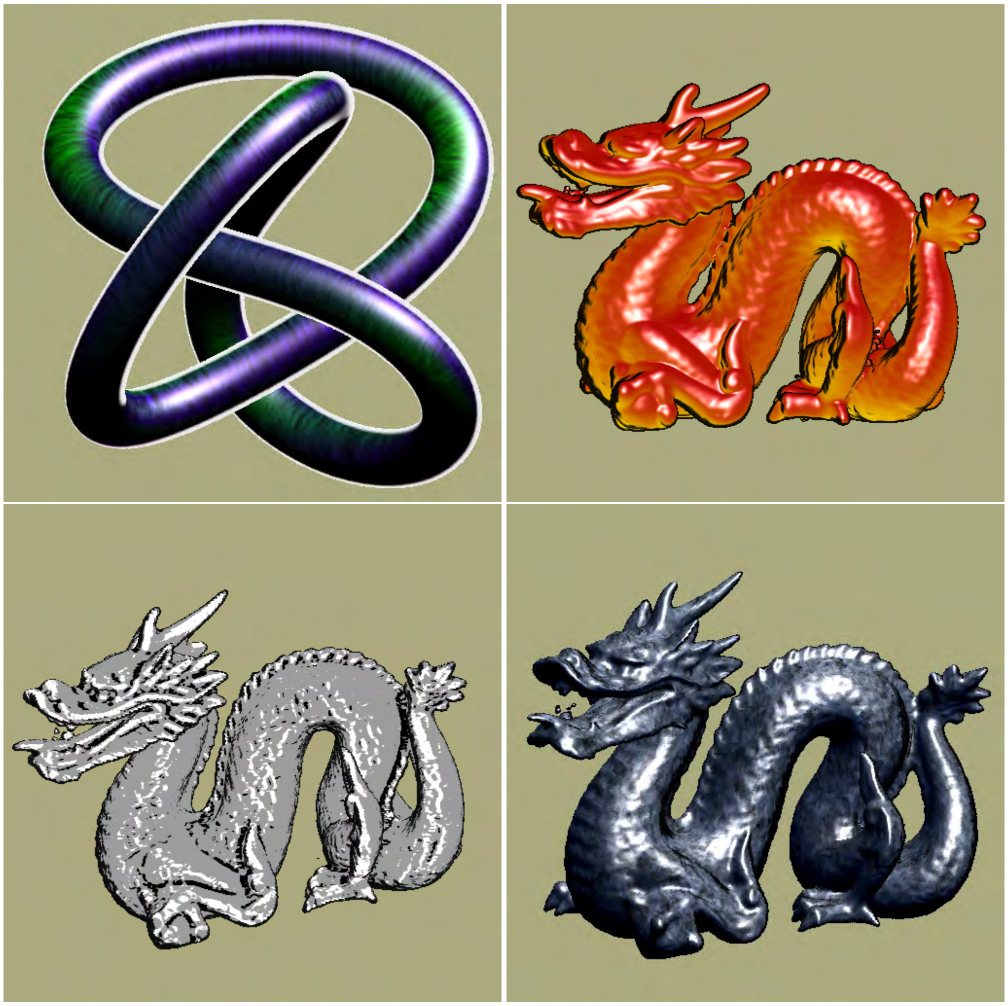


Figure 11: Curvature-based NPR. (top, left) Contours, curvature magnitude colors, and flow in curvature direction; (top, right) tone shading and contours; (bottom, left) contours, ridges, and valleys; (bottom, right) flow in curvature direction.

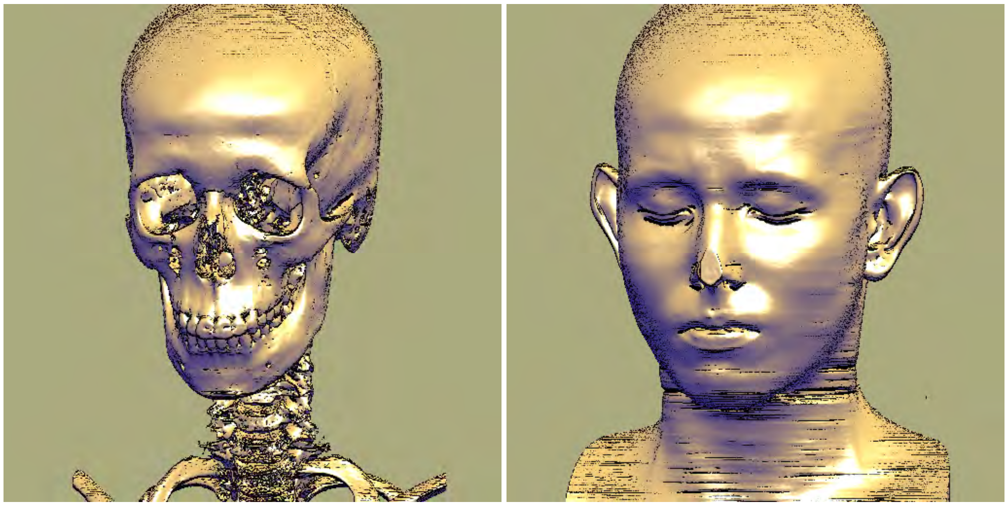


Figure 12: CT scan (512x512x333) with tone shading and curvature-controlled contours with ridge and valley lines specified in the (κ_1, κ_2) domain via a 2D transfer function.

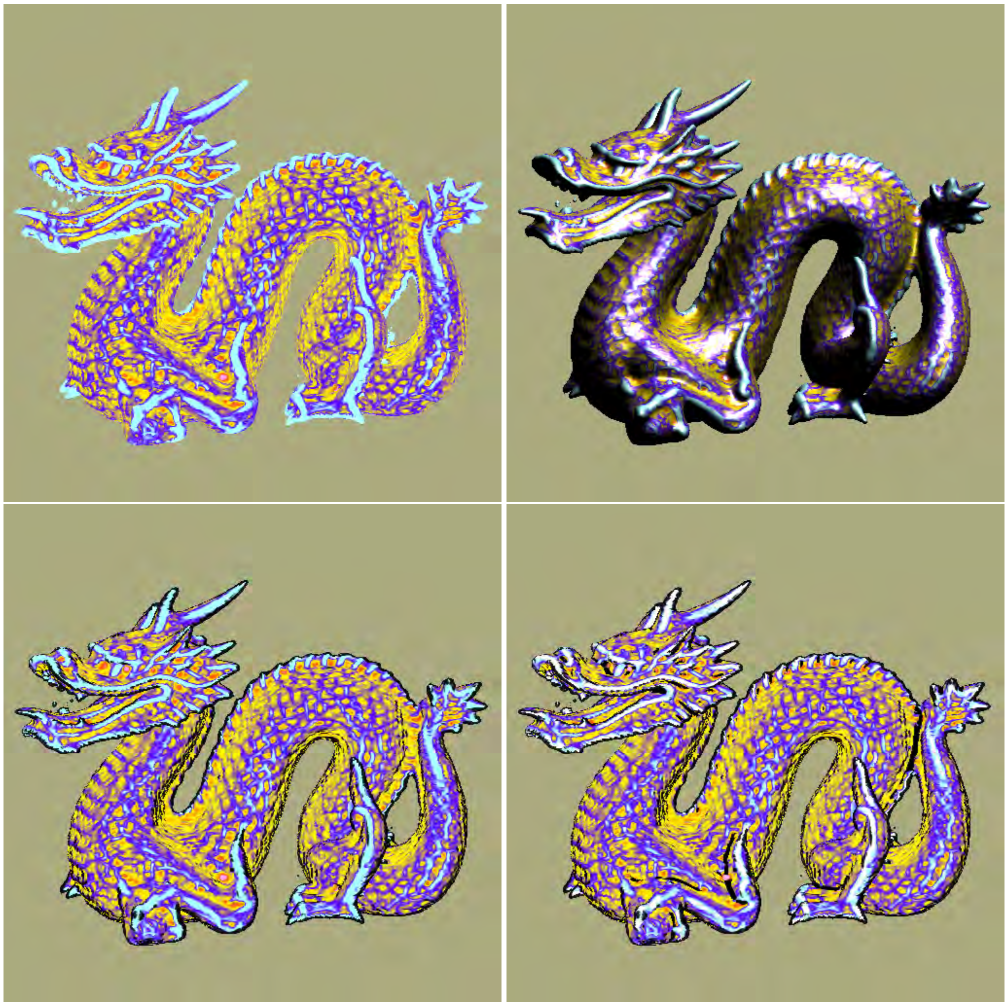


Figure 13: Dragon distance field (128x128x128) with colors from curvature magnitude (top, left); with Phong shading (top, right); with contours (bottom, left); with ridge and valley lines (bottom, right).

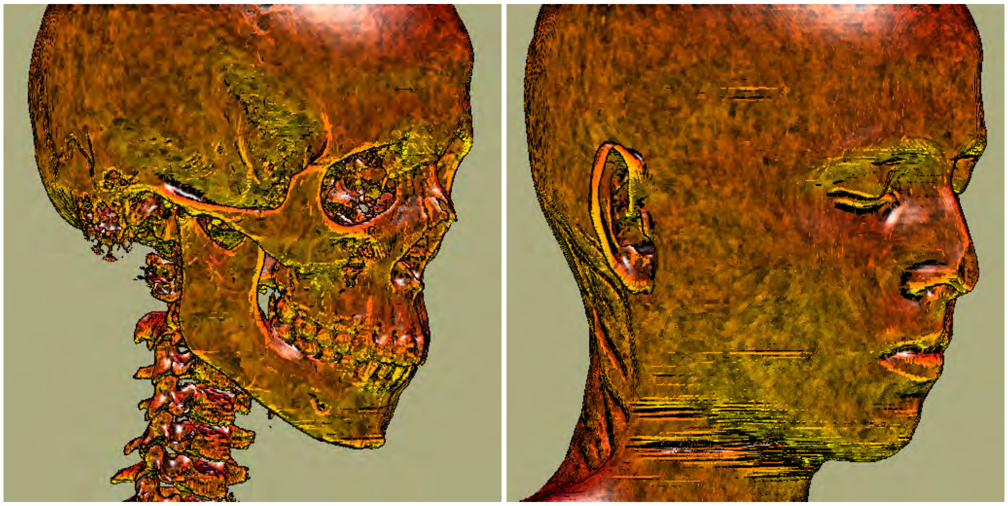


Figure 14: CT scan (256x256x333) with contours, ridges and valleys, tone shading, and image space flow advection to generate a noise pattern on the surface.

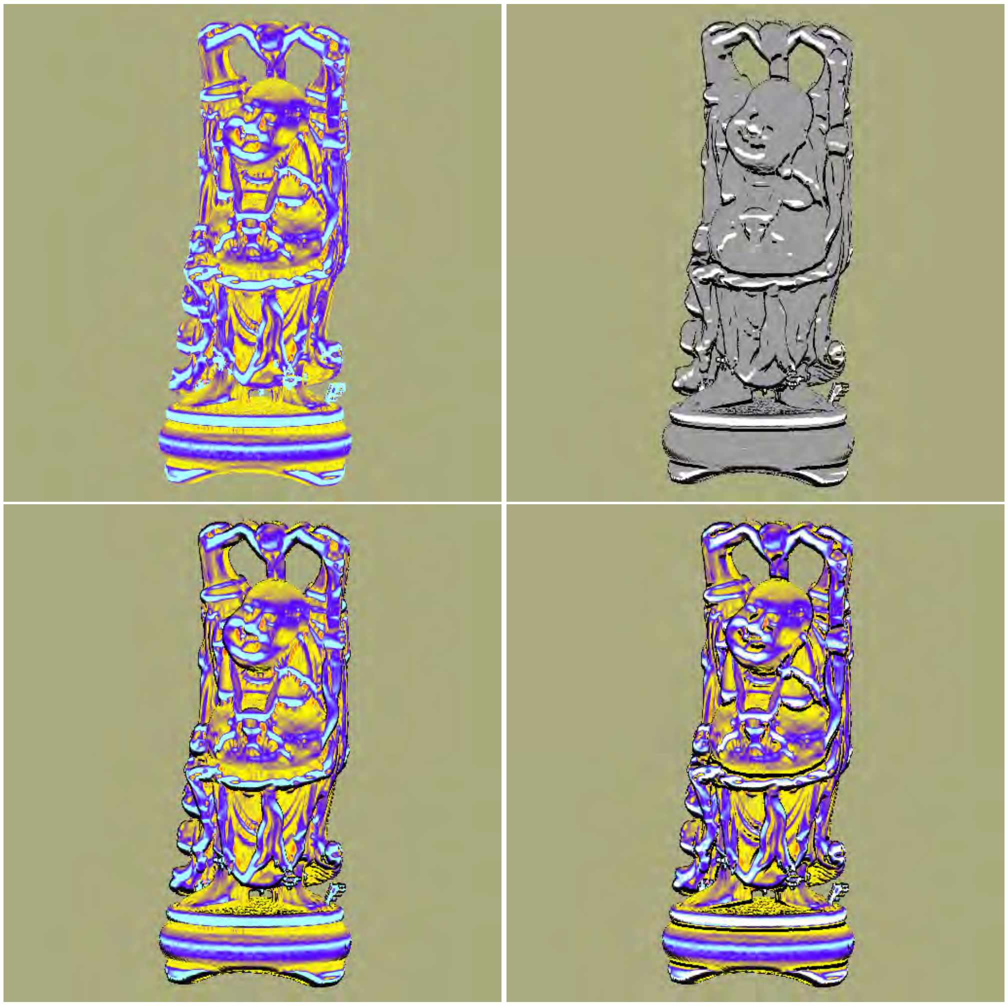


Figure 15: Happy Buddha distance field (128x128x128) with colors from curvature magnitude (top, left); only ridge and valley lines (top, right); with contours (bottom, left); with contours, and ridge and valley lines (bottom, right).

2 Volume Rendering of Segmented Data

One of the most important goals in volume rendering, especially when dealing with medical data, is to be able to visually separate and selectively enable specific objects of interest contained in a single volumetric data set.

A very powerful approach to facilitate the perception of individual objects is to create explicit object membership information via *segmentation* [32]. The process of segmentation determines a set of voxels that belong to a given object of interest, usually in the form of one or several *segmentation masks*. There are two major ways of representing segmentation information in masks. First, each object can be represented by a single binary segmentation mask, which determines for each voxel whether it belongs to the given object or not. Second, an object ID volume can specify segmentation information for all objects in a single volume, where each voxel contains the ID of the object it belongs to. These masks can then be used to selectively render only some of the objects contained in a single data set, or render different objects with different optical properties such as transfer functions, for example. Volumes with object ID tags are often also called *tagged volumes* [9].

Other approaches for achieving visual distinction of objects are for example rendering multiple semi-transparent isosurfaces, or direct volume rendering with an appropriate transfer function. In the latter approach, multi-dimensional transfer functions [14, 16] have proven to be especially powerful in facilitating the perception of different objects. However, it is often the case that a single rendering method or transfer function does not suffice in order to distinguish multiple objects of interest according to a user's specific needs, especially when spatial information needs to be taken into account. Non-photorealistic volume rendering methods [2, 4, 24] have also proven to be promising approaches for achieving better perception of individual objects.

An especially powerful approach is to combine different non-photorealistic and traditional volume rendering methods in a single volume rendering. When segmentation information is available, different objects can be rendered with individual per-object rendering modes, which allows to use specific modes for structures they are well suited for, as well as separating *focus* from *context*. Even further, different objects can be rendered with their own individual compositing mode, combining the contributions of all objects with a single global compositing mode. This two-level approach to object compositing can facilitate object perception very effectively and is known as *two-level volume rendering* [10, 11].

Overview

Integrating segmentation information and multiple rendering modes with different sets of parameters into a fast high-quality volume renderer is not a trivial problem, especially in the case of consumer hardware volume rendering, which tends to only be fast when all or most voxels can be treated identically. On such hardware, one would also like to use a single segmentation mask volume in order to use a minimal amount of texture memory. Graphics hardware cannot easily interpolate between voxels belonging to different objects, however, and using the segmentation mask without filtering gives rise to artifacts. Thus, one of the major obstacles in such a scenario is filtering object boundaries in order to attain high quality in conjunction with consistent fragment assignment and without introducing non-existent object IDs due to interpolation.

In this chapter, we show how segmented volumetric data sets can be rendered efficiently and with high quality on current consumer graphics hardware. The segmentation information for object distinction can be used at multiple levels of sophistication, and we describe how all of these different possibilities can be integrated into a single coherent hardware volume rendering framework.

First, different objects can be rendered with the same rendering technique (e.g., DVR), but with different transfer functions. Separate per-object transfer functions can be applied in a single rendering pass even when object boundaries are filtered during rendering. On an ATI Radeon 9700, up to eight transfer functions can be folded into a single rendering pass with linear boundary filtering. If boundaries are only point-sampled, e.g., during interaction, an arbitrary number of transfer functions can be used in a single pass. However, the number of transfer functions with boundary filtering in a single pass is no conceptual limitation and increases trivially on architectures that allow more instructions in the fragment shader.

Second, different objects can be rendered using different hardware fragment shaders. This allows easy integration of methods as diverse as non-photorealistic and direct volume rendering, for instance. Although each distinct fragment shader requires a separate rendering pass, multiple objects using the same fragment shader with different rendering parameters can effectively be combined into a single pass. When multiple passes cannot be avoided, the cost of individual passes is reduced drastically by executing expensive fragment shaders only for those fragments active in a given pass. These two properties allow highly interactive rendering of segmented data sets, since even for data sets with many objects usually only a couple of different rendering modes are employed. We have implemented direct volume rendering with post-classification, pre-integrated classification [5], different shading modes, non-polygonal isosurfaces, and maximum intensity pro-

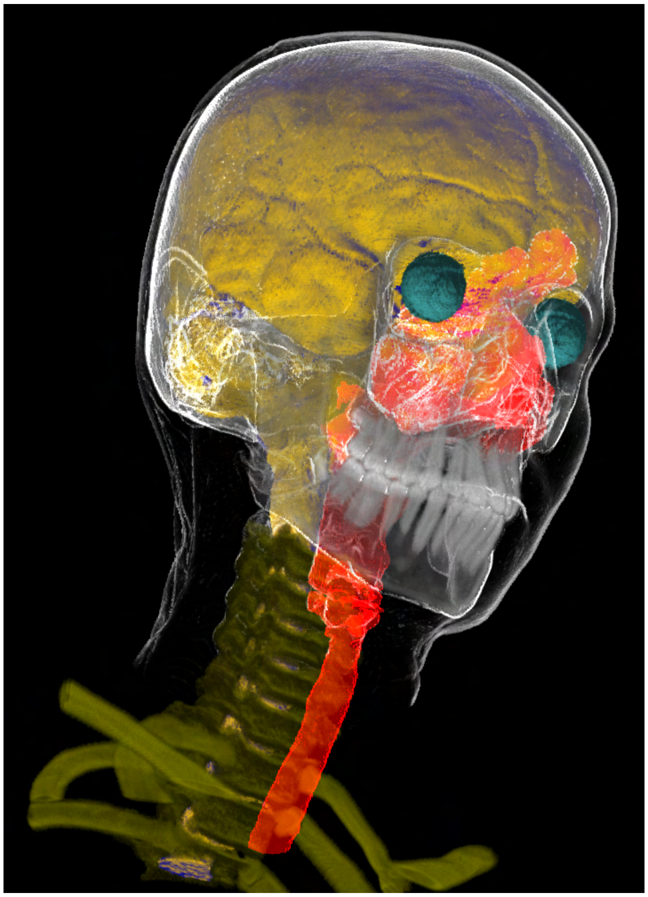


Figure 17: Segmented head and neck data set (256x256x333) with eight different enabled objects – brain: tone shading; skin: contour enhancement with clipping plane; eyes and spine: shaded DVR; skull, teeth, and vertebrae: unshaded DVR; trachea: MIP.

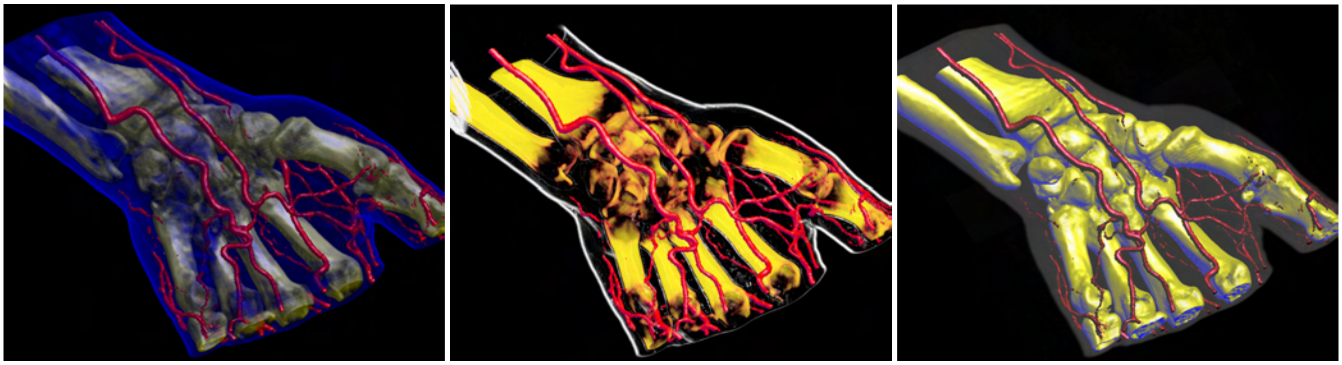


Figure 16: Segmented hand data set (256x128x256) with three objects: skin, blood vessels, and bone. Two-level volume rendering integrates different transfer functions, rendering and compositing modes: (left) all objects rendered with shaded DVR; the skin partially obscures the bone; (center) skin rendered with non-photorealistic contour rendering and MIP compositing, bones rendered with DVR, vessels with tone shading; (right) skin rendered with MIP, bones with tone shading, and vessels with shaded isosurfacing; the skin merely provides context.

jection. See figures 16 and 18 for example images. In addition to non-photorealistic contour enhancement [2] (figure 16, center; figure 18, skull), we have also used a volumetric adaptation of tone shading [6] (figure 16, right), which improves depth perception in contrast to standard shading.

Finally, different objects can also be rendered with different compositing modes, e.g., alpha blending and maximum intensity projection (MIP), for their contribution to a given pixel. These per-object compositing modes are object-local and can be specified independently for each object. The individual contributions of different objects to a single pixel can be combined via a separate global compositing mode. This two-level approach to object compositing [10, 11] has proven to be very useful in order to improve perception of individual objects.

The major points of this chapter are:

- A systematic approach to minimizing both the number of rendering passes and the performance cost of individual passes when rendering segmented volume data with high quality on current GPUs. Both filtering of object boundaries and the use of different rendering parameters such as transfer functions do not prevent using a single rendering pass for multiple objects. Even so, each pass avoids execution of the corresponding potentially expensive fragment shader for irrelevant fragments by exploiting the early z-test. This reduces the performance impact of the number of rendering passes drastically.

- An efficient method for mapping a single object ID volume to and from a domain where filtering produces correct results even when three or more objects are present in the volume. The method is based on simple 1D texture lookups and able to map and filter blocks of four objects simultaneously.

- An efficient object-order algorithm based on simple depth and stencil buffer operations that achieves correct compositing of objects with different per-object compositing modes and an additional global compositing mode. The result is conceptually identical to being able to switch compositing modes for any given group of samples along the ray for any given pixel.

2.1 Segmented Data Representation

For rendering purposes, we simply assume that in addition to the usual data such as a density and an optional gradient volume, a *segmentation mask volume* is also available. If embedded objects are represented as separate masks, we combine all of these masks into a single volume that contains a single object ID for each voxel in a pre-process. Hence we will also be calling this segmentation mask volume the *object ID volume*. IDs are simply enumerated consecutively starting with one, i.e., we do not assign individual bits to specific objects. ID zero is reserved (see later sections).

The object ID volume consumes one byte per voxel and is ei-

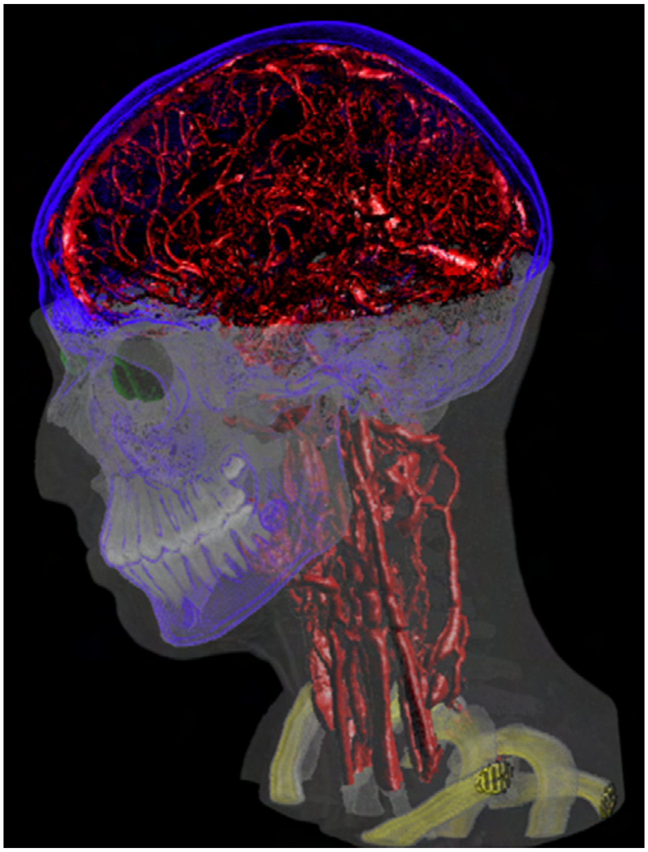


Figure 18: Segmented head and neck data set (256x256x333) with six different enabled objects. The skin and teeth are rendered as MIP with different intensity ramps, the blood vessels and eyes are rendered as shaded DVR, the skull uses contour rendering, and the vertebrae use a gradient magnitude-weighted transfer function with shaded DVR. A clipping plane has been applied to the skin object.

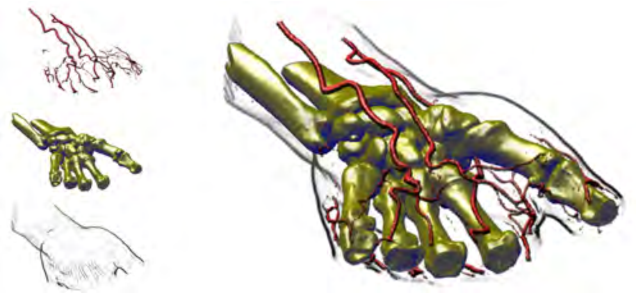


Figure 19: CT scan of a human hand (256x128x256) with three segmented objects (skin, blood vessels, and bone structure). The skin is rendered with contour enhancement, the vessels with shaded DVR, and the bones with tone shading.

ther stored in its own 3D texture in the case of view-aligned slicing, or in additional 2D slice textures for all three slice stacks in the case of object-aligned slicing. With respect to resolution, we have used the same resolution as the original volume data, but all of the approaches we describe could easily be used for volume and segmentation data of different resolutions.

2.2 Rendering Segmented Data

In order to render a segmented data set, we determine object membership of individual fragments by filtering object boundaries in the hardware fragment shader (section 2.3). Object membership determines which transfer function, rendering, and compositing modes should be used for a given fragment. See figure 19 for an example of three segmented objects rendered with per-object rendering modes and transfer functions.

We render the volume in a number of rendering passes that is basically independent of the number of contained objects. It most of all depends on the required number of different hardware configurations that cannot be changed during a single pass, i.e., the fragment shader and compositing mode. Objects that can share a given configuration can be rendered in a single pass. This also extends to the application of multiple per-object transfer functions (section 2.3) and thus the actual number of rendering passes is usually much lower than the number of objects or transfer functions. It depends on several major factors:

Enabled objects. If all the objects rendered in a given pass have been disabled by the user, the entire rendering pass can be skipped. If only some of the objects are disabled, the number of passes stays the same, independent of the order of object IDs. Objects are disabled by changing a single entry of a 1D lookup texture. Additionally, per-object clipping planes can be enabled. In this case, all objects rendered in the same pass are clipped identically, however.

Rendering modes. The rendering mode, implemented as an actual hardware fragment shader, determines what and how volume data is re-sampled and shaded. Since it cannot be changed during a single rendering pass, another pass must be used if a different fragment shader is required. However, many objects often use the same basic rendering mode and thus fragment shader, e.g., DVR and isosurfacing are usually used for a large number of objects.

Transfer functions. Much more often than the basic rendering mode, a change of the transfer function is required. For instance, all objects rendered with DVR usually have their own individual transfer functions. In order to avoid an excessive number of rendering passes due to simple transfer function changes, we apply multiple transfer functions to different objects in a single rendering pass while still retaining adequate filtering quality (section 2.3).

Compositing modes. Although usually considered a part of the rendering mode, compositing is a totally separate operation in graphics hardware. Where the basic rendering mode is determined by the fragment shader, the compositing mode is specified as blend function and equation in OpenGL, for instance. It determines how already shaded fragments are combined with pixels stored in the frame buffer. Changing the compositing mode happens even more infrequently than changing the basic rendering mode, e.g., alpha blending is used in conjunction with both DVR and tone shading.

Different compositing modes per object also imply that the (conceptual) ray corresponding to a single pixel must be able to combine the contribution of these different modes (figure 26). Especially in the context of texture-based hardware volume rendering, where no actual rays exist and we want to obtain the same result with an object-order approach instead, we have to use special care when compositing. The contributions of individual objects to a given pixel should not interfere with each other, and are combined with a single global compositing mode.

In order to ensure correct compositing, we are using two render buffers and track the current compositing mode for each pixel. Whenever the compositing mode changes for a given pixel, the already composited part is transferred from the *local compositing buffer* into the *global compositing buffer*. Section 2.4 shows that this can actually be done very efficiently without explicitly considering individual pixels, while still achieving the same compositing behavior as a ray-oriented image-order approach, which is crucial for achieving high quality. For faster rendering we allow falling back to single-buffer compositing during interaction (figure 27).

The basic rendering loop

We will now outline the basic rendering loop that we are using for each frame. Table 1 gives a high-level overview.

Although the user is dealing with individual objects, we automatically collect all objects that can be processed in the same rendering pass into an *object set* at the beginning of each frame. For each object set, we generate an *object set membership texture*, which is a 1D lookup table that determines the objects belonging to the set. In order to further distinguish different transfer functions in a single object set, we also generate 1D *transfer function assignment textures*. Both of these types of textures are shown in figure 20 and described in sections 2.2 and 2.3.

After this setup, the entire slice stack is rendered. Each slice must be rendered for every object set containing an object that intersects the slice, which is determined in a pre-process. In the case of 3D volume textures, all slices are always assumed to be intersected by all objects, since they are allowed to cut through the volume at arbitrary angles. If there is more than a single object set for the current slice, we optionally render all object set IDs of the slice into

```
DetermineObjectSets();
CreateObjectSetMembershipTextures();
CreateTFAssignmentTextures();
FOR each slice DO
  TransferLocalBufferIntoGlobalBuffer();
  ClearTransferredPixelsInLocalBuffer();
  RenderObjectIdDepthImageForEarlyZTest();
  FOR each object set with an object in slice DO
    SetupObjectSetFragmentRejection();
    SetupObjectSetTFAssignment();
    ActivateObjectSetFragmentShader();
    ActivateObjectSetCompositingMode();
    RenderSliceIntoLocalBuffer();
```

Table 1: The basic rendering loop that we are using. Object set membership can change every time an object's rendering or compositing mode is changed, or an object is enabled or disabled.

the depth buffer before rendering any actual slice data. This enables us to exploit the early z-test during all subsequent passes for each object set, see below.

We proceed by rendering actual slice data. Before a slice can be rendered for an object set, the fragment shader and compositing mode corresponding to this set must be activated. Using the two types of textures mentioned above, the fragment shader filters boundaries, rejects fragments not corresponding to the current pass, and applies the correct transfer function.

In order to attain two compositing levels, slices are rendered into a local buffer, as already outlined above. Before rendering the current slice, those pixels where the local compositing mode differs from the previous slice are transferred from the local into the global buffer using the global compositing mode. After this transfer, the transferred pixels are cleared in the local buffer to ensure correct local compositing for subsequent pixels. In the case when only a single compositing buffer is used for approximate compositing, the local to global buffer transfer and clear are not executed.

Finally, if the global compositing buffer is separate from the viewing window, it has to be transferred once after the entire volume has been rendered.

Early fragment culling via early z-test

On current graphics hardware, it is possible to avoid execution of the fragment shader for fragments where the depth test fails as long as the shader does not modify the depth value of the fragment. This early z-test is crucial to improving performance when multiple rendering passes have to be performed for each slice.

If the current slice's object set IDs have been written into the depth buffer before, see above, we reject fragments not belonging to the current object set even before the corresponding fragment shader is started. In order to do this, we use a depth test of `GL_EQUAL` and configure the vertex shader to generate a constant depth value for each fragment that exactly matches the current object set ID. Figure 21 graphically illustrates the performance difference of using the early z-test as opposed to also shading voxels that will be culled.

Excluding individual fragments from processing by an expensive fragment shader via the early z-test is also crucial in the context of GPU-based ray casting in order to be able to terminate rays individually [18].

Fragment shader operations

Most of the work in volume renderers for consumer graphics hardware is done in the fragment shader, i.e., at the granularity of individual fragments and, ultimately, pixels. In contrast to approaches using lookup tables, i.e., paletted textures, we are performing all shading operations procedurally in the fragment shader. However, we are most of all interested in the operations that are required for rendering segmented data. The two basic operations in the fragment shader with respect to the segmentation mask are fragment rejection and per-fragment application of transfer functions:

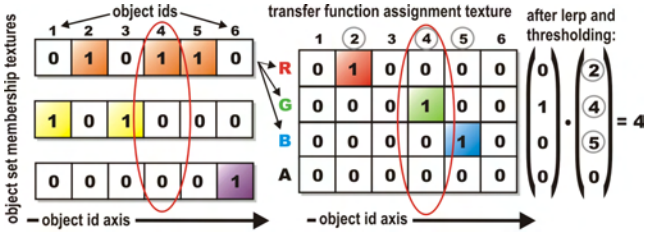


Figure 20: Object set membership textures (left; three 1D intensity textures for three sets containing three, two, and one object, respectively) contain a binary membership status for each object in a set that can be used for filtering object IDs and culling fragments. Transfer function assignment textures (right; one 1D RGBA texture for distinction of four transfer functions) are used to filter four object boundaries simultaneously and determine the corresponding transfer function via a simple dot product.

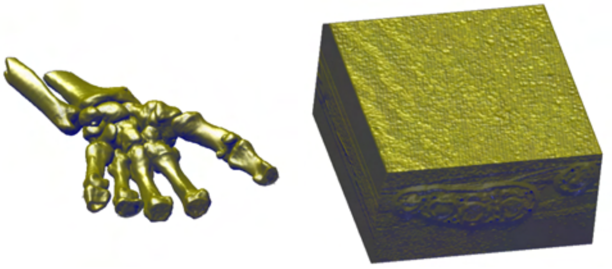


Figure 21: In order to render the bone structure shown on the left, many voxels need to be culled. The early z-test allows to avoid evaluating shading equations for culled voxels. If it is not employed, performance will correspond to shading all voxels as shown on the right.

Fragment rejection. Fragments corresponding to object IDs that cannot be rendered in the current rendering pass, e.g., because they need a different fragment shader or compositing mode, have to be rejected. They, in turn, will be rendered in another pass, which uses an appropriately adjusted rejection comparison.

For fragment rejection, we do not compare object IDs individually, but use 1D lookup textures that contain a binary membership status for each object (figure 20, left). All objects that can be rendered in the same pass belong to the same object set, and the corresponding object set membership texture contains ones at exactly those texture coordinates corresponding to the IDs of these objects, and zeros everywhere else. The re-generation of these textures at the beginning of each frame, which is negligible in terms of performance, also makes turning individual objects on and off trivial. Exactly one object set membership texture is active for a given rendering pass and makes the task of fragment rejection trivial if the object ID volume is point-sampled.

When object IDs are filtered, it is also crucial to map individual IDs to zero or one before actually filtering them. Details are given in section 2.3, but basically we are using object set membership textures to do a binary classification of input IDs to the filter, and interpolate after this mapping. The result can then be mapped back to zero or one for fragment rejection.

Per-fragment transfer function application. Since we apply different transfer functions to multiple objects in a single rendering pass, the transfer function must be applied to individual fragments based on their density value and corresponding object ID. Instead of sampling multiple one-dimensional transfer function textures, we sample a single global two-dimensional transfer function texture (figure 22). This texture is not only shared between all objects of an object set, but also between all object sets. It is indexed with one texture coordinate corresponding to the object ID, the other one to the actual density.

Because we would like to filter linearly along the axis of the actual transfer function, but use point-sampling along the axis of object IDs, we store each transfer function twice at adjacent locations in order to guarantee point-sampling for IDs, while we are using linear interpolation for the entire texture. We have applied this scheme only to 1D transfer functions, but general 2D transfer functions could also be implemented via 3D textures of just a few layers in depth, i.e., the number of different transfer functions.

We are using an extended version of the pixel-resolution filter that we employ for fragment rejection in order to determine which of multiple transfer functions in the same rendering pass a fragment should actually use. Basically, the fragment shader uses multiple RGBA transfer function assignment textures (figure 20, right) for both determining the transfer function and rejecting fragments, instead of a single object set membership texture with only a single color channel. Each one of these textures allows filtering the object ID volume with respect to four object boundaries simultaneously. A single lookup yields binary membership classification of a fragment with respect to four objects. The resulting RGBA membership vectors can then be interpolated directly. The main operation for mapping back the result to an object ID is a simple dot product with a constant vector of object IDs. If the result is the non-existent object ID of zero, the fragment needs to be rejected. The details are described in section 2.3.

This concept can be extended trivially to objects sharing transfer functions by using transfer function IDs instead of object IDs. The following two sections will now describe filtering of object boundaries at sub-voxel precision in more detail.

2.3 Boundary Filtering

One of the most crucial parts of rendering segmented volumes with high quality is that the object boundaries must be calculated during rendering at the pixel resolution of the output image, instead of the voxel resolution of the segmentation volume. Figure 23 (left) shows that simply point-sampling the object ID texture leads to object boundaries that are easily discernible as individual voxels. That is, simply retrieving the object ID for a given fragment from the segmentation volume is trivial, but causes artifacts. Instead, the object ID must be determined via filtering for each fragment individually, thus achieving pixel-resolution boundaries.

Unfortunately, filtering of object boundaries cannot be done directly using the hardware-native linear interpolation, since direct interpolation of numerical object IDs leads to incorrectly interpolated intermediate values when more than two different objects are present. When filtering object IDs, a threshold value s_t must be chosen that determines which object a given fragment belongs to, which is essentially an iso-surfacing problem.

However, this cannot be done if three or more objects are contained in the volume, which is illustrated in the top row of figure 24. In that case, it is not possible to choose a single s_t for the entire volume. The crucial observation to make in order to solve this problem is that the segmentation volume must be filtered as a successive series of binary volumes in order to achieve proper filtering [31], which is shown in the second row of figure 24. Mapping all object IDs of the current object set to 1.0 and all other IDs to 0.0 allows using a global threshold value s_t of 0.5. We of course do not want to store these binary volumes explicitly, but perform this mapping on-the-fly in the fragment shader by indexing the *object set membership texture* that is active in the current rendering pass. Filtering in the other passes simply uses an alternate binary mapping, i.e., other object set membership textures.

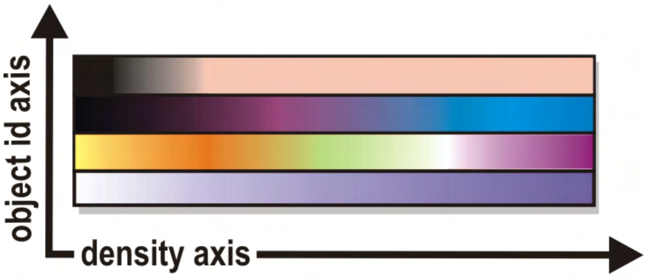


Figure 22: Instead of multiple one-dimensional transfer functions for different objects, we are using a single global two-dimensional transfer function texture. After determining the object ID for the current fragment via filtering, the fragment shader appropriately samples this texture with $(density, object\_id)$ texture coordinates.

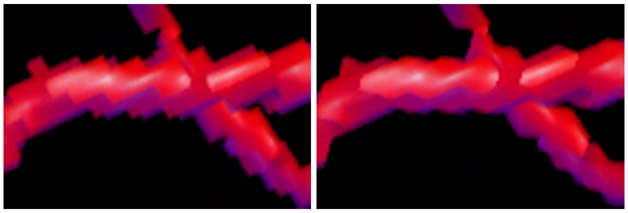


Figure 23: Object boundaries with voxel resolution (left) vs. object boundaries determined per-fragment with linear filtering (right).

One problem with respect to a hardware implementation of this approach is that texture filtering happens before the sampled values can be altered in the fragment shader. Therefore, we perform filtering of object IDs directly in the fragment shader. Note that our approach could in part also be implemented using texture palettes and hardware-native linear interpolation, with the restriction that not more than four transfer functions can be applied in a single rendering pass. However, we have chosen to perform all filtering in the fragment shader in order to create a coherent framework with a potentially unlimited number of transfer functions in a single rendering pass and prepare for the possible use of cubic boundary filtering in the future.

After filtering yields values in the range $[0.0, 1.0]$, we once again come to a binary decision whether a given fragment belongs to the current object set by comparing with a threshold value of 0.5 and rejecting fragments with an interpolated value below this threshold (figure 24, third row).

Actual rejection of fragments is done using the KIL instruction of the hardware fragment shader that is available in the ARB_fragment_program OpenGL extension, for instance. It can also be done by mapping the fragment to RGBA values constituting the identity with respect to the current compositing mode (e.g.,

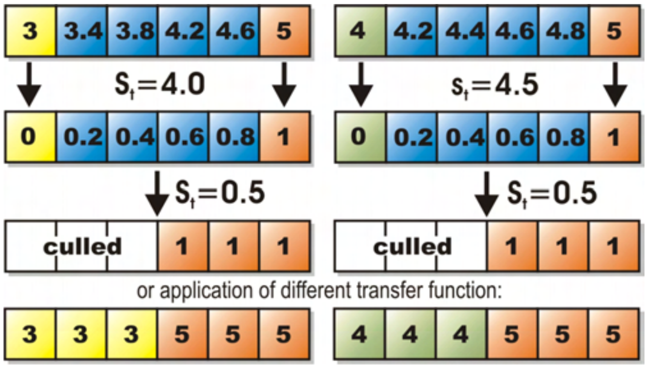


Figure 24: Each fragment must be assigned an exactly defined object ID after filtering. Here, IDs 3, 4, and 5 are interpolated, yielding the values shown in blue. Top row: choosing a single threshold value s_t that works everywhere is not possible for three or more objects. Second row: object IDs must be converted to 0.0 or 1.0 in the fragment shader before interpolation, which allows using a global s_t of 0.5. After thresholding, fragments can be culled accordingly (third row), or mapped back to an object ID in order to apply the corresponding transfer function (fourth row).

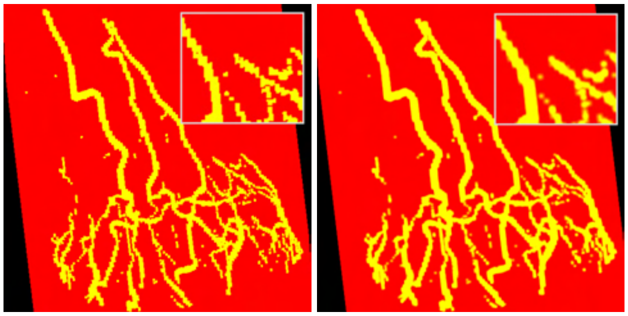


Figure 25: Selecting the transfer function on a per-fragment basis. In the left image, point-sampling of the object ID volume has been used, whereas in the right image procedural linear interpolation in the fragment shader achieves results of much better quality.

an alpha of zero for alpha blending), in order to not alter the frame buffer pixel corresponding to this fragment.

Linear boundary filtering. For object-aligned volume slices, bi-linear interpolation is done by setting the hardware filtering mode for the object ID texture to nearest-neighbor and sampling it four times with offsets of whole texels in order to get access to the four ID values needed for interpolation. Before actual interpolation takes place, the four object IDs are individually mapped to 0.0 or 1.0, respectively, using the current object set membership texture.

We perform the actual interpolation using a variant of texture-based filtering, which proved to be both faster and use fewer instructions than using `LRP` instructions. With this approach, bi-linear weight calculation and interpolation can be reduced to just one texture fetch and one dot product. When intermediate slices are interpolated on-the-fly [27], or view-aligned slices are used, eight instead of four input IDs have to be used in order to perform tri-linear interpolation.

Combination with pre-integration. The combination of pre-integration [5] and high-quality clipping has been described recently [28]. Since our filtering method effectively reduces the segmentation problem to a clipping problem on-the-fly, we are using the same approach after we have mapped object IDs to 0.0 or 1.0, respectively. In this case, the interpolated binary values must be used for adjusting the pre-integration lookup.

Multiple per-object transfer functions in a single rendering pass

In addition to simply determining whether a given fragment belongs to a currently active object or not, which has been described in the previous section, this filtering approach can be extended to the application of multiple transfer functions in a single rendering pass without sacrificing filtering quality. Figure 25 shows the difference in quality for two objects with different transfer functions (one entirely red, the other entirely yellow for illustration purposes).

In general hardware-accelerated volume rendering, the easiest way to apply multiple transfer functions in a single rendering pass would be to use the original volume texture with linear interpolation, and an additional separate point-sampled object ID texture. Although actual volume and ID textures could be combined into a single texture, the use of a separate texture to store the IDs is mandatory in order to prevent that filtering of the actual volume data also reverts back to point-sampling, since a single texture cannot use different filtering modes for different channels and point-sampling is mandatory for the ID texture. The hardware-native linear interpolation cannot be turned on in order to filter object IDs, and thus the resolution of the ID volume is easily discernible if the transfer functions are sufficiently different.

In order to avoid the artifacts related to point-sampling the ID texture, we perform several almost identical filtering steps in the fragment shader, where each of these steps simultaneously filters the object boundaries of four different objects. After the fragment's object ID has been determined via filtering, it can be used to access the global transfer function table as described in section 2.2 and illustrated in figure 22. For multiple simultaneous transfer functions, we do not use object set membership textures but the similar extended concept of *transfer function assignment textures*, which is illustrated in the right image of figure 20.

Each of these textures can be used for filtering the object ID volume with respect to four different object IDs at the same time by using the four channels of an RGBA texture in order to perform four simultaneous binary classification operations. In order to create these textures, each object set membership texture is converted into $\lceil \#objects/4 \rceil$ transfer function assignment textures, where $\#objects$ denotes the number of objects with different transfer functions in a given object set. All values of 1.0 corresponding to the first transfer function are stored into the red channel of this texture, those corresponding to the second transfer function into the green channel, and so on.

In the fragment shader, bi-linear interpolation must index this texture at four different locations given by the object IDs of the four input values to interpolate. This classifies the four input object IDs with respect to four objects with just four 1D texture sampling operations. A single linear interpolation step yields the linear interpolation of these four object classifications, which can then be compared against a threshold of $(0.5, 0.5, 0.5, 0.5)$, also requiring only a single operation for four objects. Interpolation and thresholding yields a vector with at most one component of 1.0, the other components set to 0.0. In order for this to be true, we require that interpolated and thresholded repeated binary classifications never overlap, which is not guaranteed for all types of filter kernels. In the case of bi-linear or tri-linear interpolation, however, overlaps can never occur [31].

The final step that has to be performed is mapping the binary classification to the desired object ID. We do this via a single dot product with a vector containing the four object IDs corresponding to the four channels of the transfer function assignment texture (figure 20, right). By calculating this dot product, we multiply exactly the object ID that should be assigned to the final fragment by 1.0. The other object IDs are multiplied by 0.0 and thus do not change the result. If the result of the dot product is 0.0, the fragment does not belong to any of the objects under consideration and can be culled. Note that exactly for this reason, we do not use object IDs of zero.

For the application of more than four transfer functions in a single rendering pass, the steps outlined above can be executed multiple times in the fragment shader. The results of the individual dot products are simply summed up, once again yielding the ID of the object that the current fragment belongs to.

Note that the calculation of filter weights is only required once, irrespective of the number of simultaneous transfer functions, which is also true for sampling the original object ID textures.

Equation 3 gives the major fragment shader resource requirements of our filtering and binary classification approach for the case of bi-linear interpolation with `LRP` instructions:

$$4\textbf{TEX\_2D} + 4\left\lceil \frac{\#objects}{4} \right\rceil \textbf{TEX\_1D} + 3\left\lceil \frac{\#objects}{4} \right\rceil \textbf{LRP}, \qquad (3)$$

in addition to one dot product and one thresholding operation (e.g., `DP4` and `SGE` instructions, respectively) for every $\lceil \#objects/4 \rceil$ transfer functions evaluated in a single pass.

Similarly to the alternative linear interpolation using texture-based filtering that we have outlined in section 2.3, procedural weight calculation and the `LRP` instructions can once again also be substituted by texture fetches and a few cheaper ALU instructions.

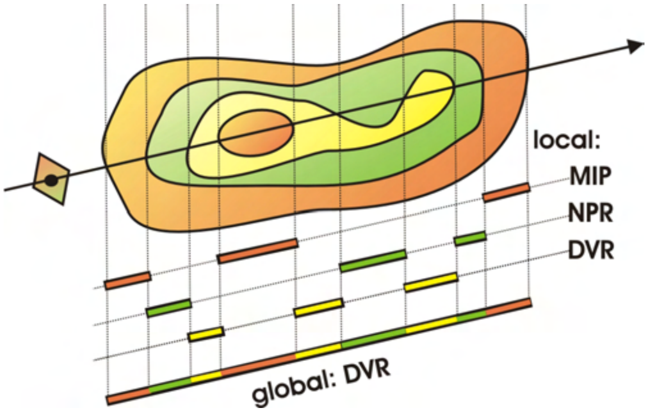


Figure 26: A single ray corresponding to a given image pixel is allowed to pierce objects that use their own object-local compositing mode. The contributions of different objects along a ray are combined with a single global compositing mode. Rendering a segmented data set with these two conceptual levels of compositing (local and global) is known as *two-level volume rendering*.

On the Radeon 9700, we are currently able to combine high-quality shading with up to eight transfer functions in the same fragment shader, i.e., we are using up to two transfer function assignment textures in a single rendering pass.

2.4 Two-Level Volume Rendering

The final component of the framework presented in this chapter with respect to the separation of different objects is the possibility to use individual object-local compositing modes, as well as a single global compositing mode, i.e., *two-level volume rendering* [10, 11]. The local compositing modes that can currently be selected are alpha blending (e.g., for DVR or tone shading), maximum intensity projection (e.g., for MIP or contour enhancement), and isosurface rendering. Global compositing can either be done by alpha blending, MIP, or a simple summation of all contributions.

Although the basic concept of two-level volume rendering is best explained using an image-order approach, i.e., individual rays (figure 26), in the context of texture-based volume rendering we have to implement it in object-order. As described in section 2.2, we are using two separate rendering buffers, a local and a global compositing buffer, respectively. Actual volume slices are only rendered into the local buffer, using the appropriate local compositing mode. When a new fragment has a different local compositing mode than the pixel that is currently stored in the local buffer, that pixel has to be transferred into the global buffer using the global compositing mode. Afterward, these transferred pixels have to be cleared in the local buffer before the corresponding new fragment is rendered. Naturally, it is important that both the detection of a change in compositing mode and the transfer and clear of pixels is done for all pixels simultaneously.

```
TransferLocalBufferIntoGlobalBuffer() {
   ActivateContextGlobalBuffer();
   DepthTest( NOT_EQUAL );
   StencilTest( RENDER_ALWAYS, SET_ONE );
   RenderSliceCompositingIds( DEPTH_BUFFER );
   DepthTest( DISABLE );
   StencilTest( RENDER_WHERE_ONE, SET_ZERO );
   RenderLocalBufferImage( COLOR_BUFFER );
}
```

Table 2: Detecting for all pixels simultaneously where the compositing mode changes from one slice to the next, and transferring those pixels from the local into the global compositing buffer.

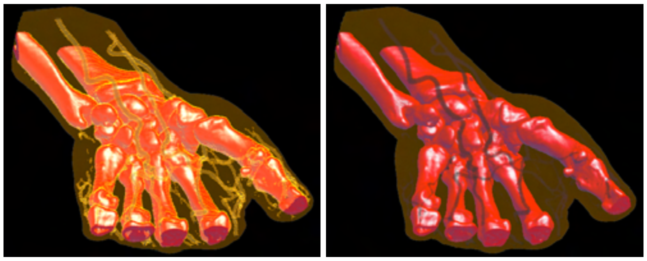


Figure 27: Detecting changes in compositing mode for each individual sample along a ray can be done exactly using two rendering buffers (left), or approximately using only a single buffer (right).

In order to do this, we are using the depth buffer of both the local and the global compositing buffer to track the current local compositing mode of each pixel, and the stencil buffer to selectively enable pixels where the mode changes from one slice to the next. Before actually rendering a slice (see table 1), we render IDs corresponding to the local compositing mode into both the local and the global buffer's depth buffer. During these passes, the stencil buffer is set to one where the ID already stored in the depth buffer (from previous passes) differs from the ID that is currently being rendered. This gives us both an updated ID image in the depth buffer, and a stencil buffer that identifies exactly those pixels where a change in compositing mode has been detected.

We then render the image of the local buffer into the global buffer. Due to the stencil test, pixels will only be rendered where the compositing mode has actually changed. Table 2 gives pseudo code for what is happening in the global buffer. Clearing the just transferred pixels in the local buffer works almost identically. The only difference is that in this case we do not render the image of another buffer, but simply a quad with all pixels set to zero. Due to the stencil test, pixels will only be cleared where the compositing mode has actually changed.

Note that all these additional rendering passes are much faster than the passes actually rendering and shading volume slices. They are independent of the number of objects and use extremely simple fragment shaders. However, the buffer/context switching overhead is quite noticeable, and thus correct separation of compositing modes can be turned off during interaction. Figure 27 shows a comparison between approximate and correct compositing with one and two compositing buffers, respectively. When only a single buffer is used, the compositing mode is simply switched according to each new fragment without avoiding interference with the previous contents of the frame buffer.

The visual difference depends highly on the combination of compositing modes and spatial locations of objects. The example in figure 27 uses MIP and DVR compositing in order to highlight the potential differences. However, using approximate compositing is very useful for faster rendering, and often exhibits little or no loss in quality. Also, it is possible to get an almost seamless performance/quality trade-off between the two, by performing the buffer transfer only every *n* slices instead of every slice. See figures 17, 28, and 29 for two-level volume renderings of segmented volume data.

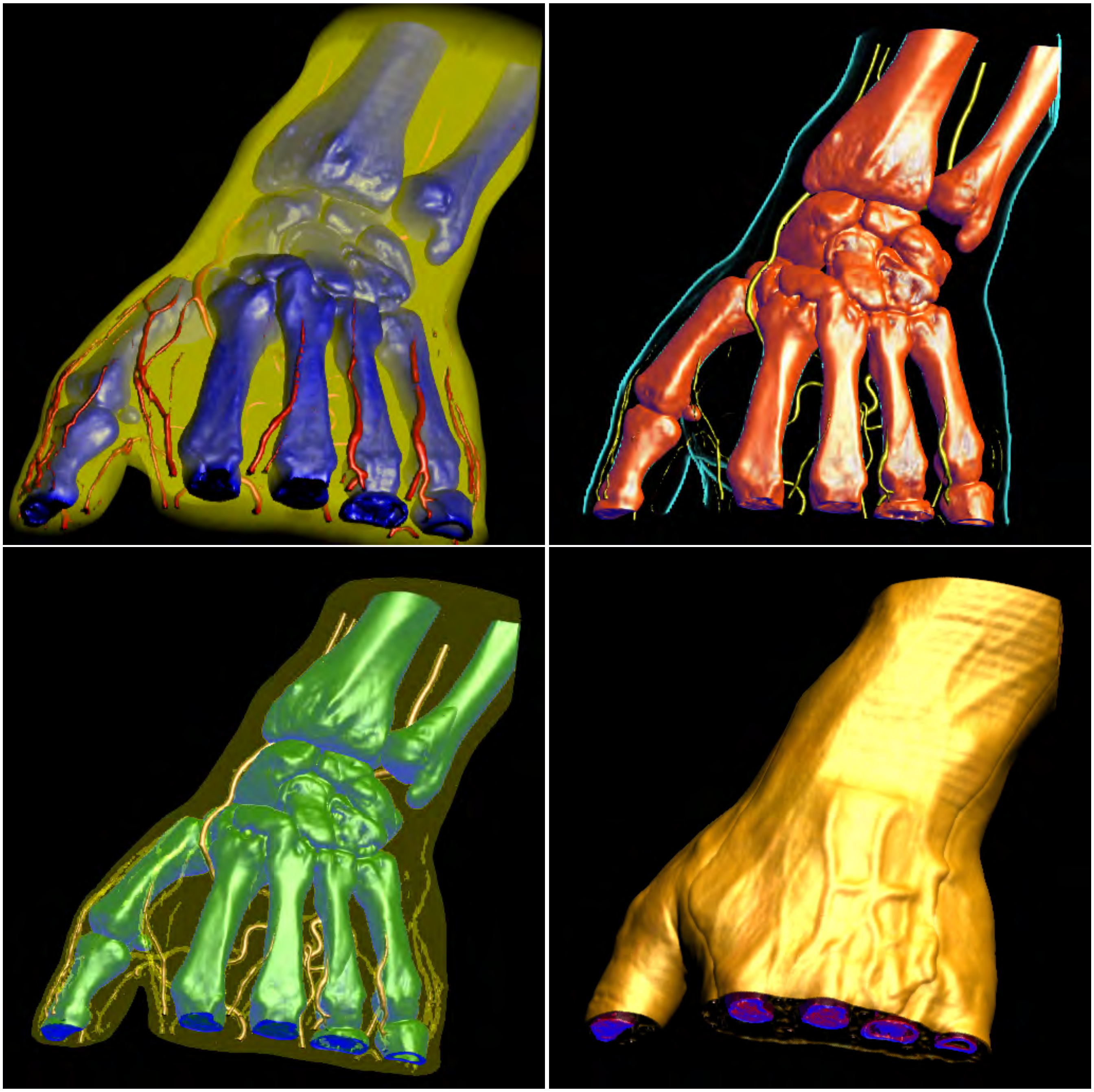


Figure 28: Hand data set (256x128x256) examples of different rendering and compositing modes. (top, left) skin with unshaded DVR, vessels and bones with shaded DVR; (top, right) skin with contour rendering, vessels with shaded DVR, bones with tone shading; (bottom, left) skin with MIP, vessels with shaded DVR, bones with tone shading; (bottom, right) skin with isosurfacing, occluded vessels and bones with shaded DVR.

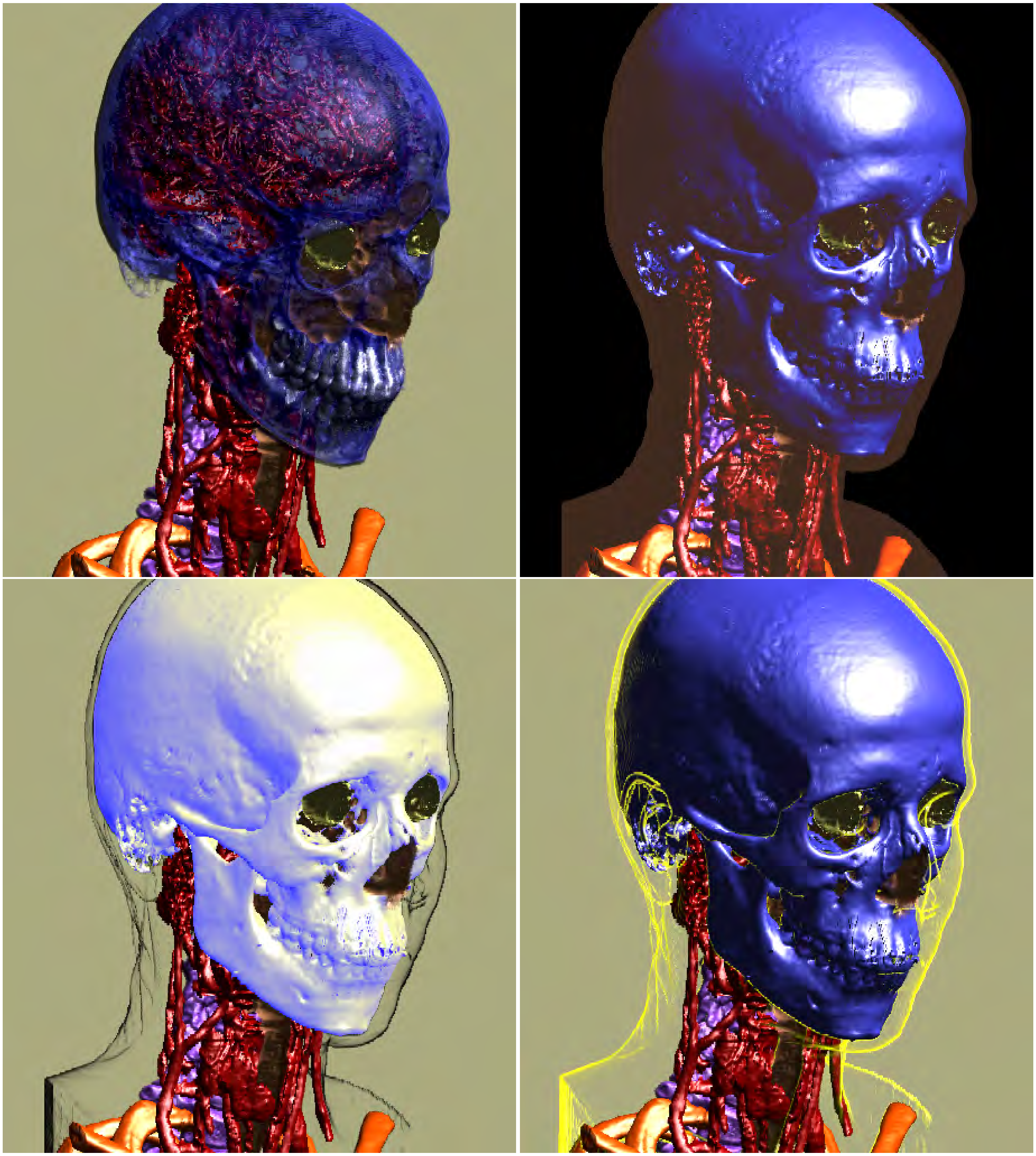


Figure 29: Head and neck data set (256x256x333) examples of different rendering and compositing modes. (top, left) skin disabled, skull with shaded DVR; (top, right) skin with MIP, skull with isosurfacing; (bottom, left) skin with contour rendering, skull with tone shading; (bottom, right) skin with contour rendering, skull with isosurfacing.

References

[1] A. Barr. Ray tracing deformed surfaces. In *Proceedings of SIGGRAPH '86*, pages 287–296, 1986.

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

[3] M. Deering, S. Winner, B. Schediwy, C. Duffy, and N. Hunt. The triangle processor and normal vector shader: a VLSI system for high performance graphics. In *Proceedings of SIGGRAPH '88*, pages 21–30, 1988.

[4] D. Ebert and P. Rheingans. Volume illustration: Non-photorealistic rendering of volume models. In *Proceedings of IEEE Visualization 2000*, pages 195–202, 2000.

[5] K. Engel, M. Kraus, and T. Ertl. High-quality pre-integrated volume rendering using hardware-accelerated pixel shading. In *Proceedings of Graphics Hardware 2001*, pages 9–16, 2001.

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

[7] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proceedings of IEEE Visualization 2003*, pages 301–308, 2003.

[8] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross. Real-time ray-casting and advanced shading of discrete isosurfaces. In *Proceedings of EUROGRAPHICS 2005*, 2005.

[9] P. Hastreiter, H. K. Cakmak, and T. Ertl. Intuitive and interactive manipulation of 3D data sets by integrating texture mapping based volume rendering into the openinventor class hierarchy. In *Bildverarbeitung in der Medizin: Algorithmen, Systeme, Anwendungen*, pages 149–154. Springer Verlag, 1996.

[10] H. Hauser, L. Mroz, G.-I. Bischi, and E. Gröller. Two-level volume rendering - fusing MIP and DVR. In *Proceedings of IEEE Visualization 2000*, pages 211–218, 2000.

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

[12] J. Hladůvka, A. König, and E. Gröller. Curvature-based transfer functions for direct volume rendering. In *Proceedings of Spring Conference on Computer Graphics 2000*, pages 58–65, 2000.

[13] V. Interrante, H. Fuchs, and S. Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *Proceedings of IEEE Visualization '95*, pages 52–59, 1995.

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

[15] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, 2003.

[16] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001*, pages 255–262, 2001.

[17] L. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel. Feature sensitive surface extraction from volume data. In *Proceedings of SIGGRAPH 2001*, pages 57–66, 2001.

[18] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE Visualization 2003*, pages 287–292, 2003.

[19] B. Laramee, B. Jobard, and H. Hauser. Image space based visualization of unsteady flow on surfaces. In *Proceedings of IEEE Visualization 2003*, pages 131–138, 2003.

[20] B. Laramee, J. van Wijk, B. Jobard, and H. Hauser. ISA and IBFVS: Image space based visualization of flow on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2004.

[21] A. Lastra, S. Molnar, M. Olano, and Y. Wang. Real-time programmable shading. In *ACM Symposium on Interactive 3D Graphics*, pages 59–ff., 1995.

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

[23] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH '87*, pages 163–169, 1987.

[24] A. Lu, C. Morris, D. Ebert, P. Rheingans, and C. Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proceedings of IEEE Visualization 2002*, pages 211–218, 2002.

[25] D. Mitchell and A. Netravali. Reconstruction filters in computer graphics. In *Proceedings of SIGGRAPH '88*, pages 221–228, 1988.

[26] T. Möller, K. Müller, Y. Kurzion, R. Machiraju, and R. Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 143–151, 1998.

[27] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage rasterization. In *Proceedings of Graphics Hardware 2000*, pages 109–118, 2000.

[28] S. Röttger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart hardware-accelerated volume rendering. In *Proceedings of VisSym 2003*, pages 231–238, 2003.

[29] T. Saito and T. Takahashi. Comprehensible rendering of 3D shapes. In *Proceedings of SIGGRAPH '90*, pages 197–206, 1990.

[30] C. Sigg and M. Hadwiger. Fast third-order texture filtering. In *GPU Gems 2, Matt Pharr (ed.)*, pages 313–329. Addison-Wesley, 2005.

[31] U. Tiede, T. Schiemann, and K.-H. Höhne. High quality rendering of attributed volume data. In *Proceedings of IEEE Visualization '98*, pages 255–262, 1998.

[32] K. Udupa and G. Herman. *3D Imaging in Medicine*. CRC Press, 1999.

[33] J. van Wijk. Image based flow visualization for curved surfaces. In *Proceedings of IEEE Visualization 2003*, pages 745–754, 2003.

[34] R. Westermann and T. Ertl. Efficiently using graphics hardware in volume rendering applications. In *Proceedings of SIGGRAPH '98*, pages 169–178, 1998.

Smart Visibility in Visualization

Smart Visibility in Visualization

Ivan Viola Meister Eduard Gröller[*]

Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria

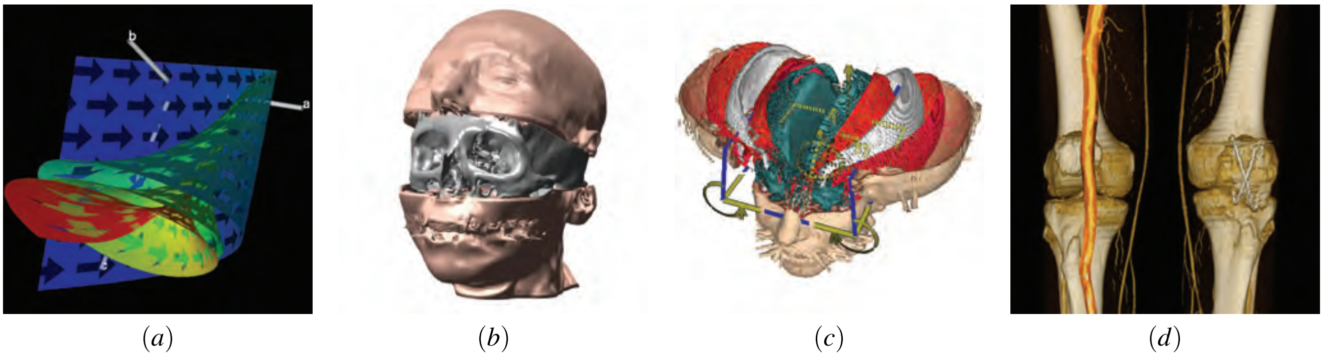


Figure 1: Examples of smart visibility visualizations: cut-away view visualization of complex dynamical systems [17] (a), volumetric splitting [14] (b), browsing in the features through leafer deformation [18] (c), and cutaway views visualization of peripheral arteries [23] (d).

Abstract

In this part of the tutorial we first discuss expressive visualization techniques that provide maximal visual information through dynamic change in visual representation. Such techniques originate from technical illustration and are called cut-away views or ghosted views. We discuss basic principles and techniques for automatic generation of cut-away and ghosted visualizations. One approach is importance-driven feature enhancement, where the visibility of a particular feature is determined according to assigned importance information. The most appropriate level of abstraction is specified automatically to unveil the most important information. Additionally we show the applicability of cut-away views on particular visualization examples. Another approach is context-preserving illustrative volume rendering, which 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 there. Next we discuss a system for direct volume manipulation (such as 3D painting) in combination with cut-away views. Here manipulation metaphores inspired by traditional illustration are discussed. An important aspect for readily understandable visualization is labeling the data with annotations. The combination of automatic label placement with visualized data is presented and new labeling metaphors from the field of information visualization are discussed.

The second category of smart visibility techniques are based on modification of the spatial arrangement of structures. Such techniques are closely related to exploded views, often used for assembly instructions. We discuss visualization techniques that separate context information to unveil the inner focus information by splitting the context into parts and moving them apart. Another visualization technique enables browsing within the data by applying deformations like leafing, peeling, or spreading. In the case of time-varying data we present another visualization technique which is related to exploded views and is denoted as fanning in time.

Keywords: view-dependent visualization, focus+context techniques, level-of-detail techniques, illustrative techniques

1 Introduction

A typical problem in 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.

Widely used technique to resolve the visibility problem is incorporating clipping planes. A clipping plane defines two half-spaces. The context information that spatially belongs to one half-space is visible, while the other is not displayed. This is a very very easy and intuitive way to unveil the most important data. However such approach eliminates less important objects also in those viewing situations, where it would not be necessary, or even worse the spatial arrangement information is lost, because too much of context information has to be removed. Different optical properties and rendering techniques (e.g., contour rendering) in the suppressed half-space ease the problem only to a certain degree.

Effective way to visualize three-dimensional data and resolve the occlusion of the most prominent information can be analogous to technical and medical illustrations [9, 11]. Illustration challenges are very similar in this case. Illustration techniques such as cut-away views, ghosted views, or exploded views effectively unveil most important information by changing the level of visual abstraction or modifying the spatial arrangement of features. In the following we will describe illustrative visualizations that have been inspired by these illustration techniques. Figure 2 shows examples of expressive illustrations that enables to see interesting structures.

[*] {viola | meister}@cg.tuwien.ac.at

Further illustrations featuring expressive techniques can be found on the referenced web-sites [12, 13].

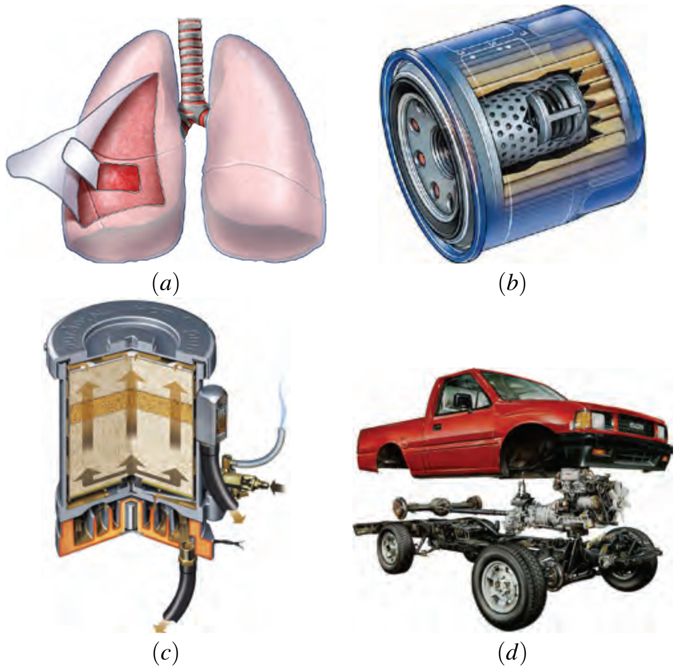


Figure 2: Different types of hand crafted expressive illustrations: cut-away view (a), ghosted view (b), section view (c), and exploded view (d). Technical illustrations are courtesy of Kevin Hulsey Illustration, Inc [13].

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 [9, 11]. An automatic generation of cut-away and ghosted views for polygonal data was introduced by Feiner and Seligmann [8]. 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. [6, 7]. 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 follow 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. [26] 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. In the following Sections 2.1 and 2.2 we focus more on visualization related tasks. First we will discuss an approach for automatic cut-away and ghosted views out of scientific volumetric data [24, 25]. This technique employs additional information about the importance of a particular feature. Afterwards we will show the potential of such expressive views on a set of applications.

2.1 Importance-Driven Feature Enhancement

Traditionally features within the volume dataset are classified by optical properties like color and opacity. With importance-driven feature enhancement we additionally assign one more dimension to features, which describes their importance. Importance encodes which features are the most interesting ones and have the highest priority to be clearly visible. Prior to the final image synthesis, the visibility of important features is estimated. If less important objects are occluding 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 its features more clearly. This allows to see all interesting structures irrespective if they are occluded 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, we use several levels of sparseness for each feature. Levels of sparseness correspond to levels of abstraction, i.e., we do not assign a single optical characteristic, but several characteristics with smooth transitions in between. 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 in front and behind. The level of sparseness thus may continuously vary within a single feature. For different viewpoints the same part of a feature 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, which we call importance compositing. This step evaluates the occlusion according to the viewpoint settings, takes the importance factor of each

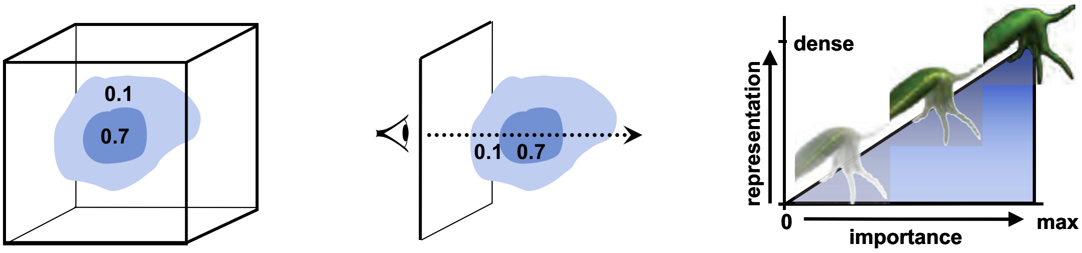


Figure 3: Stages in the pipeline of importance-driven volume rendering: Volumetric features are assigned importance values (left image). The volume is traversed (center) in the importance compositing stage to estimate levels of sparseness (right). These levels are used to enhance or suppress particular parts of the volume. The resulting images then emphasize important features.

feature into account and assigns to each feature a particular level of sparseness. The final synthesis results in images with maximal visual information with respect to the predefined object importance. The interrelationship between object importance, importance compositing, and levels of sparseness is depicted in Figure 3. The importance compositing traverses the whole volume to identify object occlusions and assigns the corresponding level of sparseness to each object. Object importance translates to object visibility in the result image. This causes different rendering settings for the context object (with importance *0.1*) in the area of the image which is covered by the focus object (importance *0.7*).

Figure 4 shows a cut-away view of multi-dimensional volumetric data of hurricane Isabel using importance-driven feature enhancement. The important feature was the hurricane eye selected through a cylindrical proxy geometry. Inside the cylinder the total precipitation mixing ratio is visualized. Thanks 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.

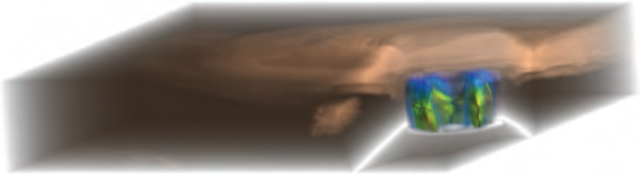


Figure 4: Cut-away visualization of a multidimensional volumetric data of hurricane Isabel.

Figure 5 illustrates a ghosted view of the scalar volumetric data of a Leopard gecko. The small internal structure (in yellow) of the Leopard gecko dataset is the most interesting information and has been pre-segmented. The body is considered as context information. In the area of occlusion the visual representation of the gecko body is reduced to contours to have a clear view on the interesting internal organ.

2.2 Applications of Expressive Visualizations

Expressive visualizations inspired by illustration techniques are useful for various visualization tasks. Straka et al. [23] 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 1 (d).

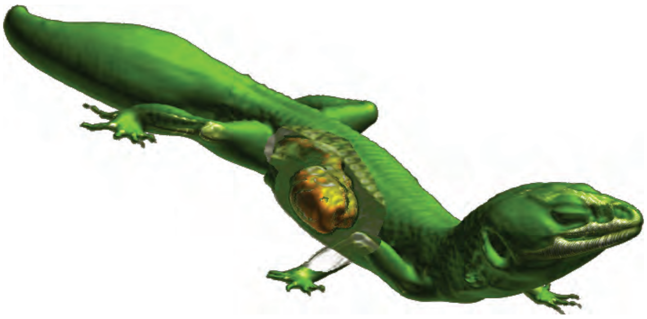


Figure 5: Ghosted visualization using contours in a CT scan of a Leopard gecko.

Krueger et al. [15] incorporate smart visibility to improve the spatial perception of feature arrangement for surgical planning. They present a system for the neck dissection planning, where the lymph nodes are emphasized using ghosted views to easily convey their spatial position. Other features such as muscles or bones are either supressed locally or globally represented in a sparse way to support the understanding of the feature arrangement. The neck dissection planning system is designed for interactive path-planning for minimal invasive interventions. Figure 6 clearly shows all lymph nodes in the neck to enable optimal path planning for the neck dissection.

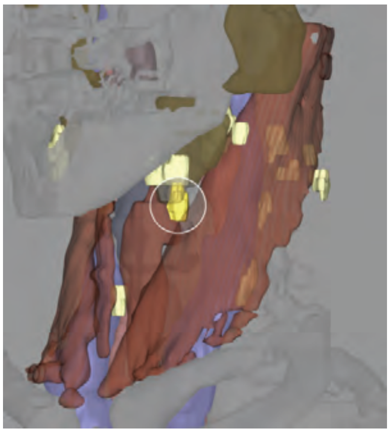


Figure 6: Smart visibility of lymph nodes in the neck. All lymph nodes are clearly visible, the currently analyzed one is additionally emphasized by a circle around it.

An extension to direct volume rendering that focuses on increas-

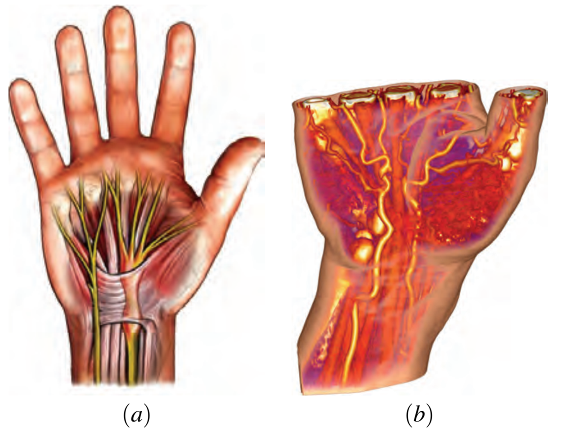


Figure 7: Medical illustration of a human hand (a) compared to the visualization of a human hand using a dynamic opacity approach as a function of the specular highlight level [2] (b).

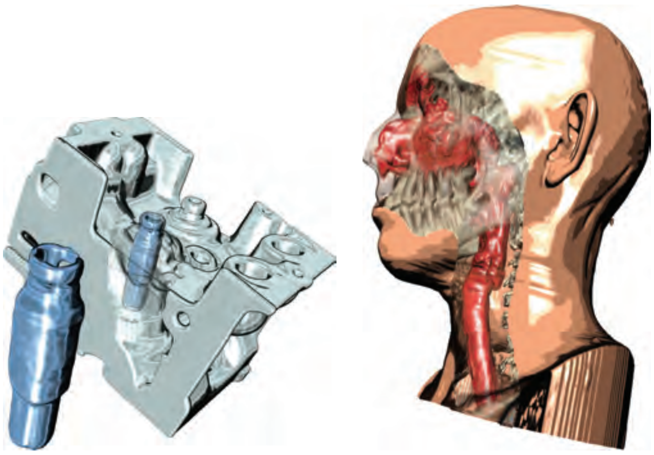


Figure 8: Interactive ghosted visualizations of the engine block and human head datasets [3, 4].

ing the visibility of features has been proposed by Bruckner et al. [2]. 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 there. 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. Their approach is compared to a medical illustration of a human hand in Figure 7.

An interactive tool for cut-away and ghosting visualizations has been recently proposed by Bruckner [3, 4]. The tool is denoted as VolumeShop and it 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 applying 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. 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 ghosted visualizations generated using VolumeShop are shown in Figure 8.

Previous applications of cut-away views are 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. [21] present an interactive drill-like tool for surgical training, which is based on the multi-volume concept. Owada et al. [20] 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 9.

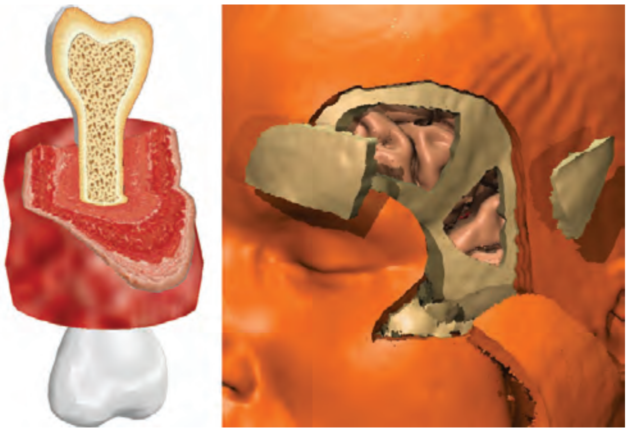


Figure 9: Volume cutting featuring two-dimensional textures for anatomy education [20] (left) and volume cutting with a drill-like tool for surgical education [21] (right).

Visualization of complex dynamical systems can be also enhanced by incorporating cuts into stream surfaces. Streamarrows proposed by Löffelmann et al. [17] 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 1 (a).

3 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 [16]. 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.

3.1 Applications of Expressive Visualization

Information visualization is a field where deformations of the data are rather wide spread. 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.

There are a lot of techniques that incorporate a kind of distortion for important feature emphasis. Typical representatives are magic lenses, fish-eye views, or perspective wall. These techniques allow to *zoom* into the data and to discriminate the focus data from the context data. An example of a magic lens applied to a document is shown in Figure 10 (a).

One technique that relates especially to smart visibility performs viewpoint-dependent distortion of three-dimensional data. This technique highlights data by dedicating more display space to it [5]. Distortions are applied to abstract graphs in order to clearly see interesting graph nodes. All nodes originally occluding the focus node are moved apart to uncover the most relevant information as shown in Figure 10 (b).

Volume splitting is a visualization technique of scientific data that is closely related to exploded views [14, 10]. 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 1 (b).

McGuffin et al. [18] 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 techniques are controlled by pop-up menus and three-dimensional widgets. The interaction technique using leafing deformation is shown in Figure 1 (c).

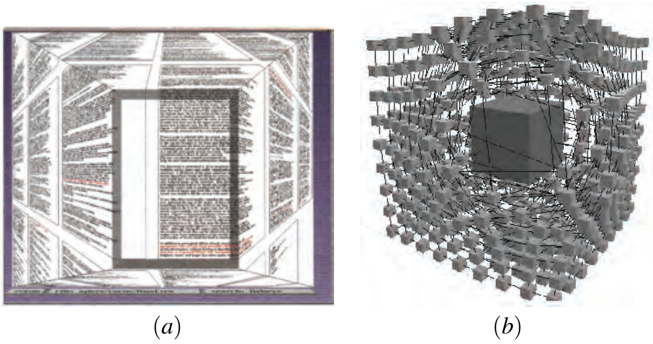


Figure 10: Information visualization examples using the smart visibility metaphor: lens metaphor for easier document browsing [22] (a) and viewpoint-dependent distortions of three-dimensional graphs [5] (b).

Another interesting visualization technique inspired by exploded views is called fanning in time [10]. This technique is different from previously mentioned approaches, because it is a temporal exploded view analogous to temporal exploded views from illustration or multiple exposure photographs. In photography and digital image processing this technique is known as computer enhanced multiple exposure numerical technique (CEMENT). 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 similar to illustrative photographs of a skateboarder performing a certain skateboard trick. Figure 11 illustrates the idea of fanning in time and the correspondence to illustrative photography.

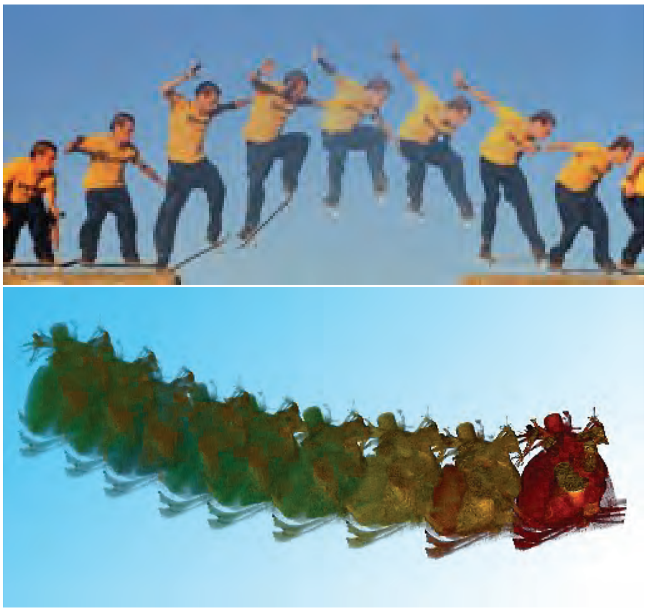


Figure 11: Illustrative photography of a skateboarder performing a jump (top). Photography expressively displays the principle of a particular trick. The photography is courtesy of Trasher magazine. Fanning in time (bottom) shows all time steps of a time-varying dataset of a beating heard [10].

4 Conclusions

In this part of the tutorial we have discussed smart visibility visualization techniques inspired by strategies from high level abstraction approaches in traditional illustration. We have shown that many challenging visualization tasks can be solved by adopting existing techniques from visual arts. Computer generated visualization still can hardly compete with hand crafted illustrations in terms of expressivity, harmony, or aesthetics. On the other hand a considerable advantage to traditional illustration is the aspect of real-time interaction and manipulation with scientific visualizations. The effective combination of expressive visualization techniques with appropriate interaction tools conveys the information of complex scientific data much better than a static image.

5 Acknowledgments

The work presented in this publication is carried out as part of the exvisation project (www.cg.tuwien.ac.at/research/vis/exvisation) supported by the Austrian Science Fund (FWF) grant no. P18322.

References

[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*, 2003.

[2] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving volume rendering. In *Proceedings of EuroVis'05*, pages 69–76, 2005.

[3] S. Bruckner and M. E. Gröller. VolumeShop: An interactive system for direct volume illustration. In *Proceedings of IEEE Visualization'05 (to appear)*, 2005.

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

[5] 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.

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

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

[8] 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.

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

[10] 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.

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

[12] Howell MediGraphicsweb page, http://www.medigraphics.com/, 2005.

[13] K. Hulsey technical illustration web page, http://www.khulsey.com/, 2005.

[14] 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.

[15] A. Krüger, C. Tietjen, J. Hintze, B. Preim, I. Hertel, and G. Strauß. Interactive visualization for neck dissection planning. In *Proceedings of EuroVis'05*, pages –, 2005.

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

[17] 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.

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

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

[20] 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.

[21] 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.

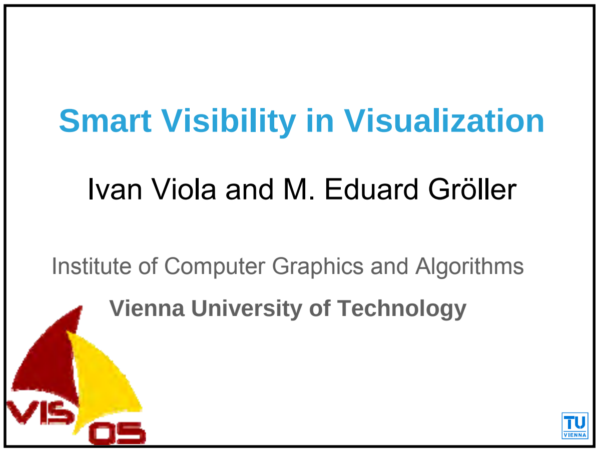
[22] G. Robertson and J. Mackinlay. The document lens. In *Proceedings of ACM Symposium on User Interface Software and Technology*, pages 101–108, 1993.

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

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

[25] 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.

[26] 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.

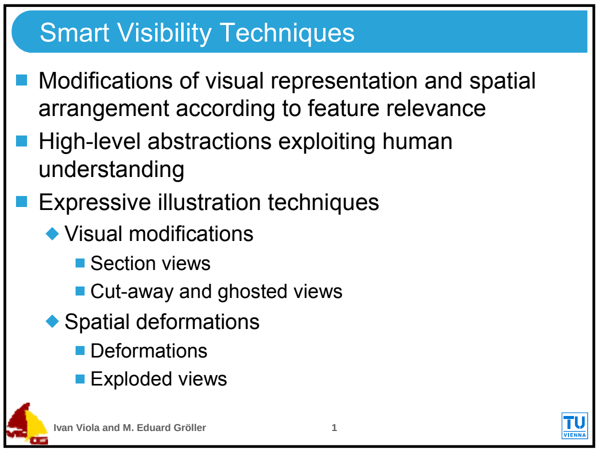


Smart Visibility in Visualization

Ivan Viola and M. Eduard Gröller

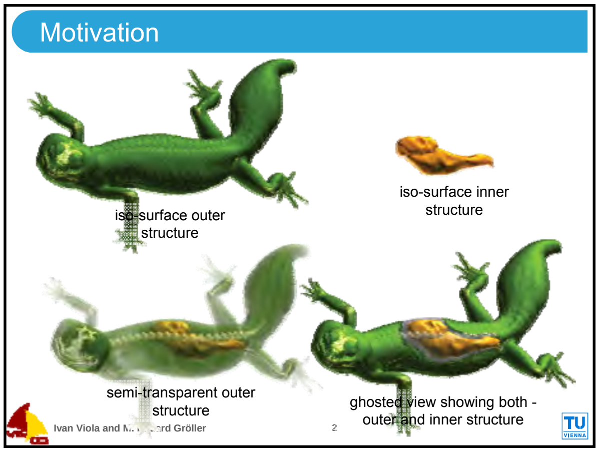
Institute of Computer Graphics and Algorithms

Vienna University of Technology



Smart Visibility Techniques

- Modifications of visual representation and spatial arrangement according to feature relevance
- High-level abstractions exploiting human understanding
- Expressive illustration techniques
  - Visual modifications
    - Section views
    - Cut-away and ghosted views
  - Spatial deformations
    - Deformations
    - Exploded views



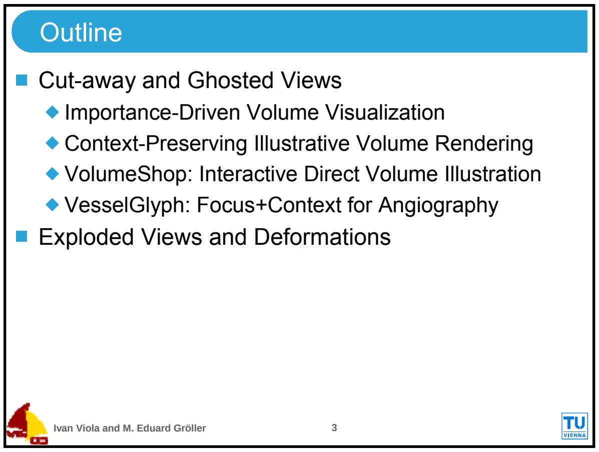
Motivation

iso-surface outer structure

iso-surface inner structure

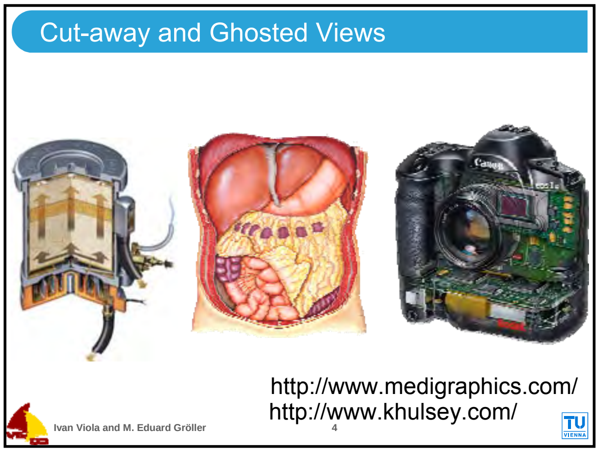
semi-transparent outer structure

ghosted view showing both - outer and inner structure



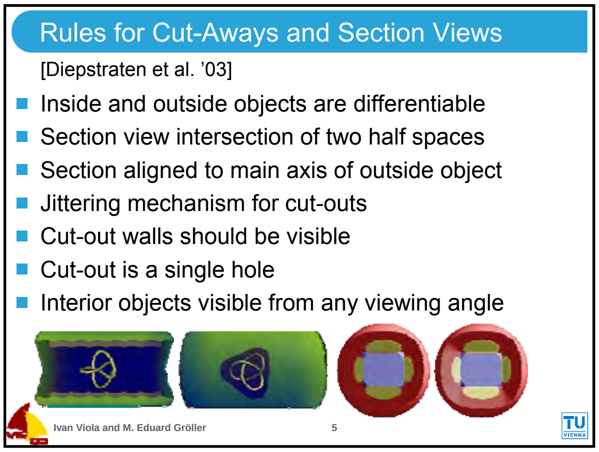
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations



Cut-away and Ghosted Views

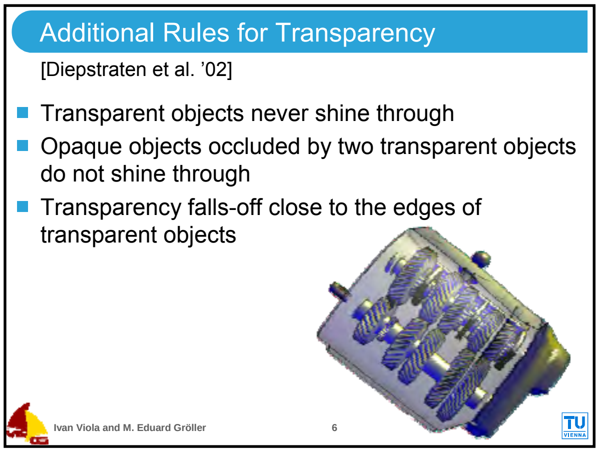
http://www.medigraphics.com/
http://www.khulsey.com/



Rules for Cut-Aways and Section Views

[Diepstraten et al. ’03]

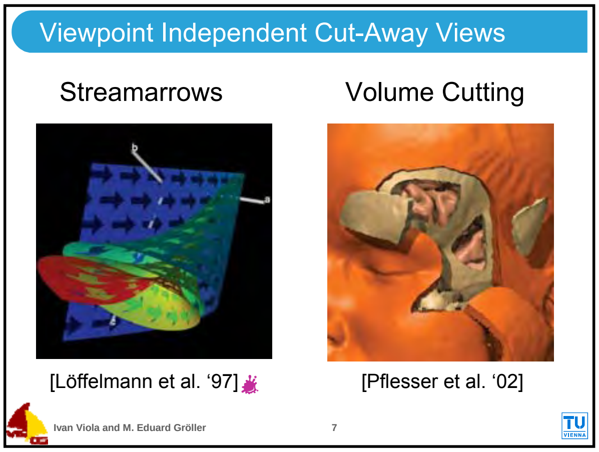
- Inside and outside objects are differentiable
- Section view intersection of two half spaces
- Section aligned to main axis of outside object
- Jittering mechanism for cut-outs
- Cut-out walls should be visible
- Cut-out is a single hole
- Interior objects visible from any viewing angle



Additional Rules for Transparency

[Diepstraten et al. '02]

- Transparent objects never shine through
- Opaque objects occluded by two transparent objects do not shine through
- Transparency falls-off close to the edges of transparent objects



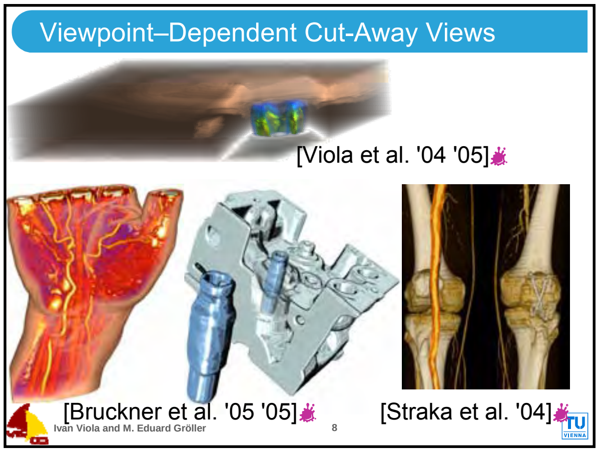
Viewpoint Independent Cut-Away Views

Streamarrows

[Löffelmann et al. '97]

Volume Cutting

[Pflesser et al. '02]

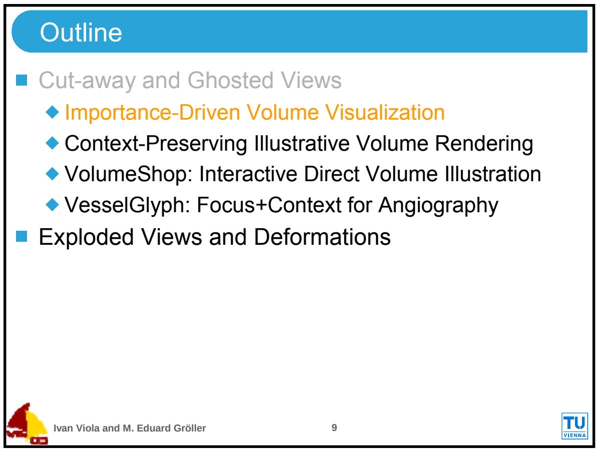


Viewpoint–Dependent Cut-Away Views

[Viola et al. '04 '05]

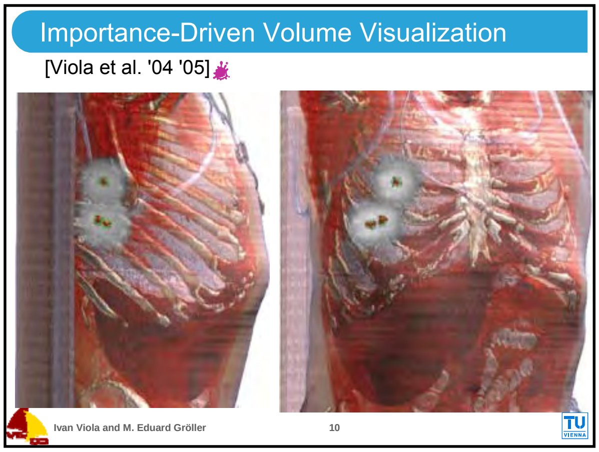
[Bruckner et al. '05 '05]

[Straka et al. '04]



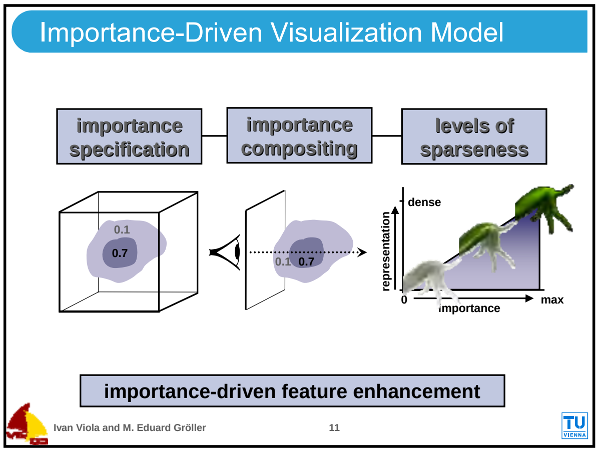
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations



Importance-Driven Volume Visualization

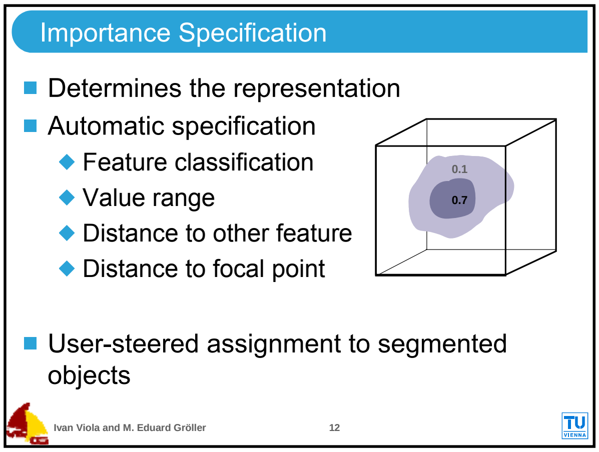
[Viola et al. '04 '05]



Importance-Driven Visualization Model

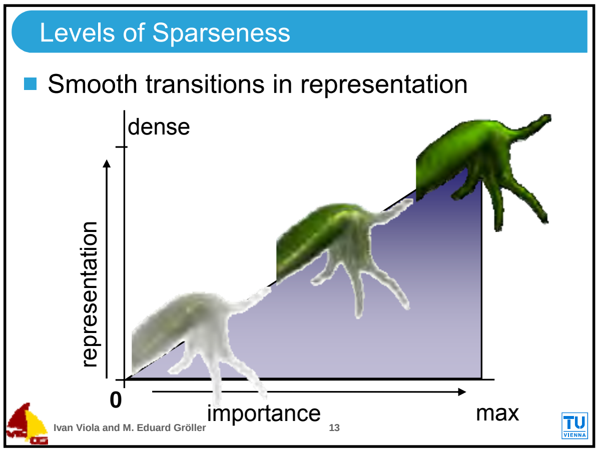
importance specification — importance compositing — levels of sparseness

importance-driven feature enhancement



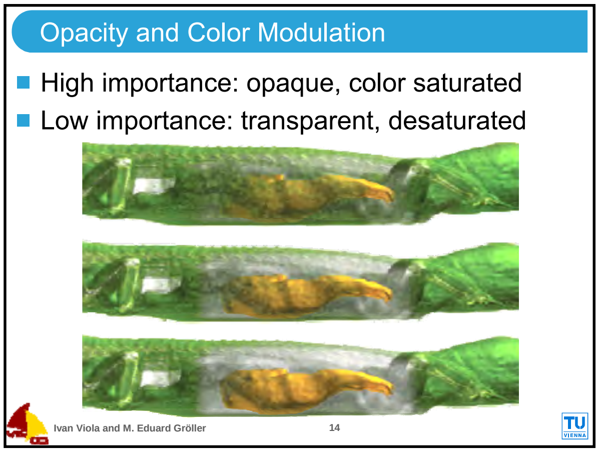
Importance Specification

- Determines the representation
- Automatic specification
  - Feature classification
  - Value range
  - Distance to other feature
  - Distance to focal point
- User-steered assignment to segmented objects



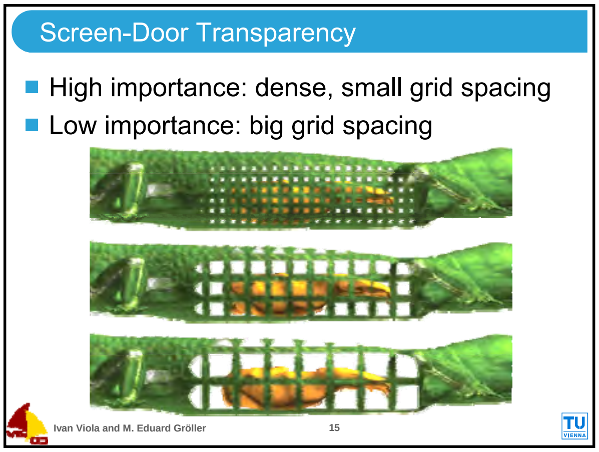
Levels of Sparseness

- Smooth transitions in representation



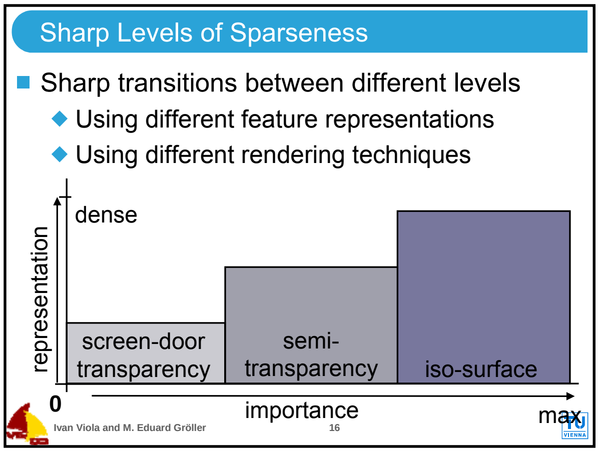
Opacity and Color Modulation

- High importance: opaque, color saturated
- Low importance: transparent, desaturated



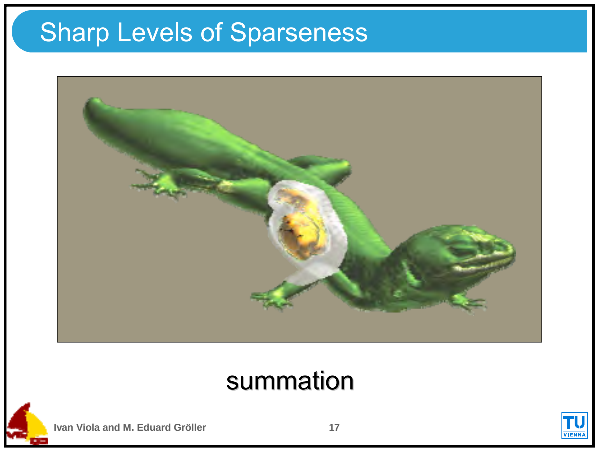
Screen-Door Transparency

- High importance: dense, small grid spacing
- Low importance: big grid spacing



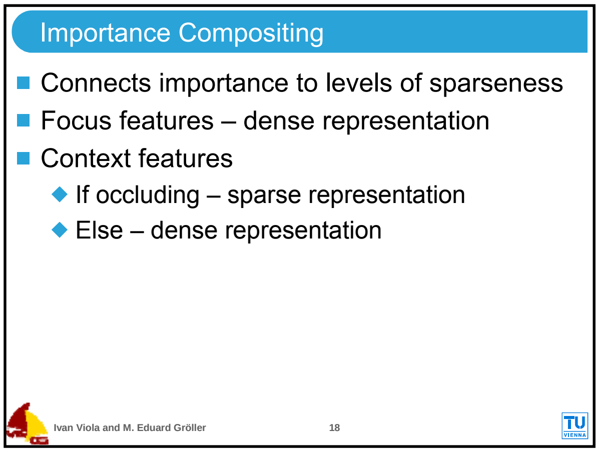
Sharp Levels of Sparseness

- Sharp transitions between different levels
  - Using different feature representations
  - Using different rendering techniques



Sharp Levels of Sparseness

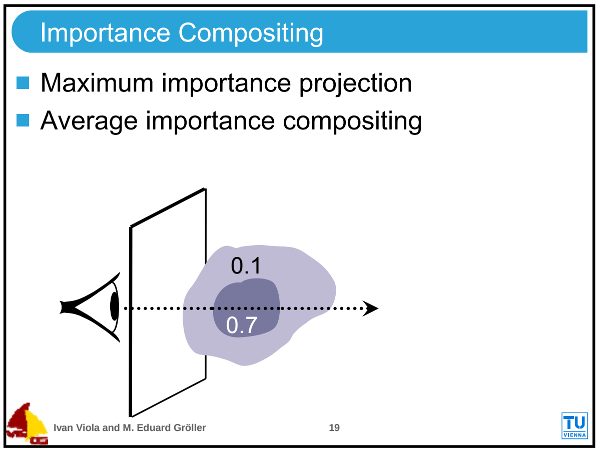
summation



Importance Compositing

- Connects importance to levels of sparseness
- Focus features – dense representation
- Context features
  - If occluding – sparse representation
  - Else – dense representation

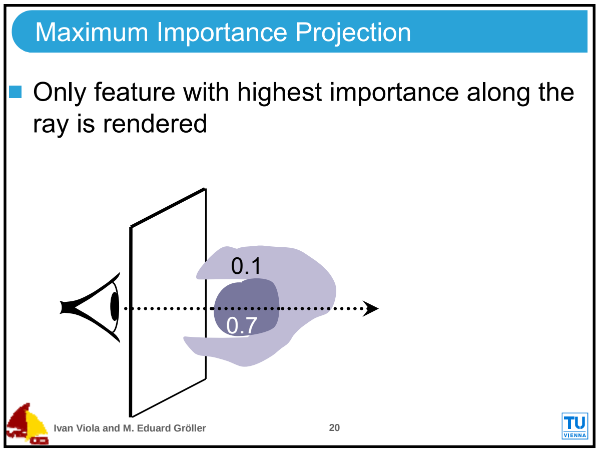
Ivan Viola and M. Eduard Gröller 18



Importance Compositing

- Maximum importance projection
- Average importance compositing

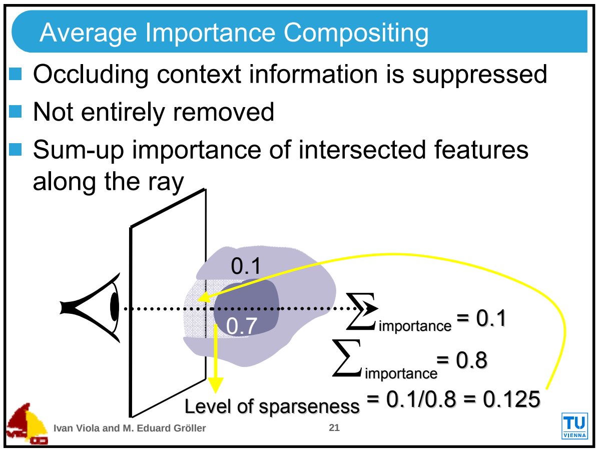
Ivan Viola and M. Eduard Gröller 19



Maximum Importance Projection

- Only feature with highest importance along the ray is rendered

Ivan Viola and M. Eduard Gröller 20



Average Importance Compositing

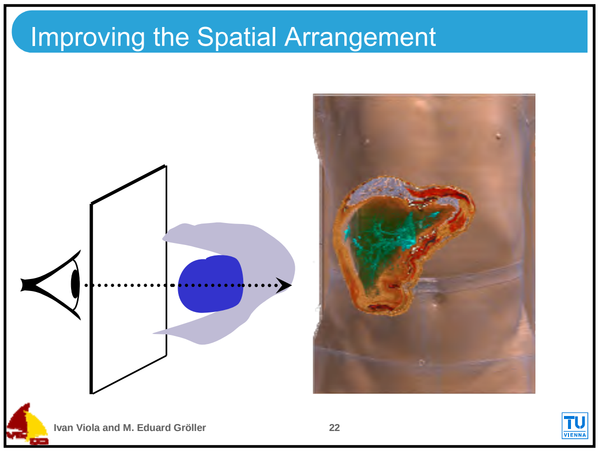
- Occluding context information is suppressed
- Not entirely removed
- Sum-up importance of intersected features along the ray

$\sum_{importance} = 0.1$

$\sum_{importance} = 0.8$

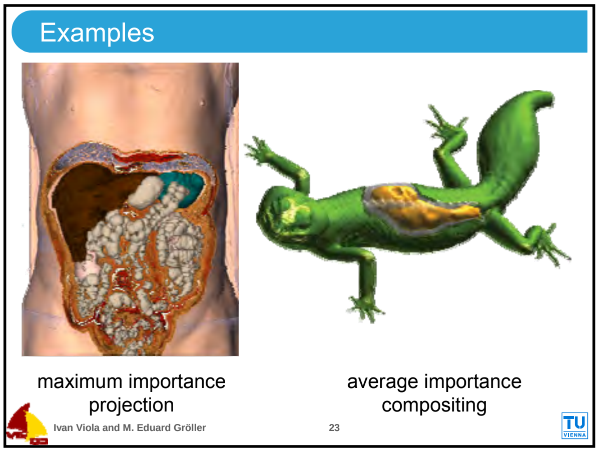
Level of sparseness = 0.1/0.8 = 0.125

Ivan Viola and M. Eduard Gröller 21



Improving the Spatial Arrangement

Ivan Viola and M. Eduard Gröller 22

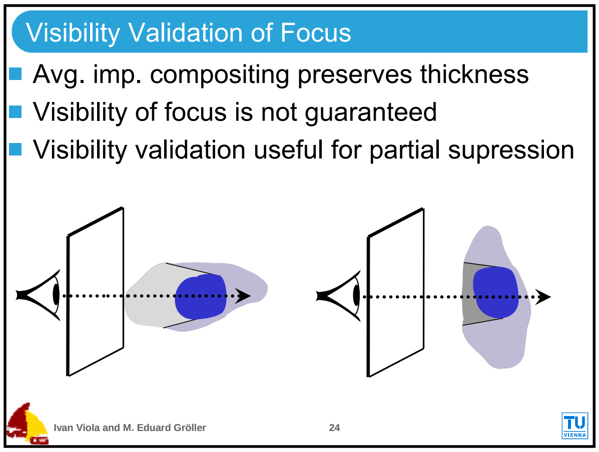


Examples

maximum importance projection

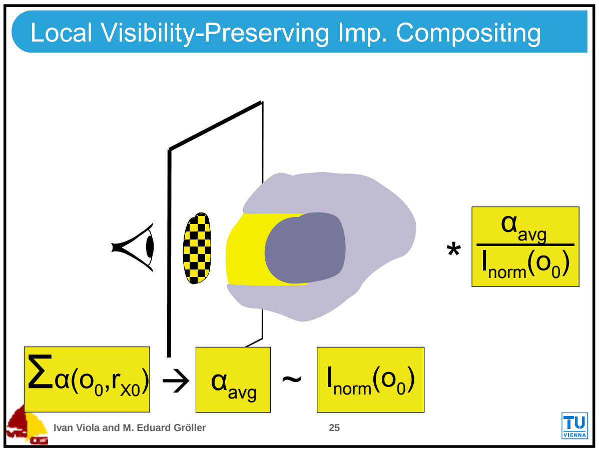
average importance compositing

Ivan Viola and M. Eduard Gröller 23



Visibility Validation of Focus

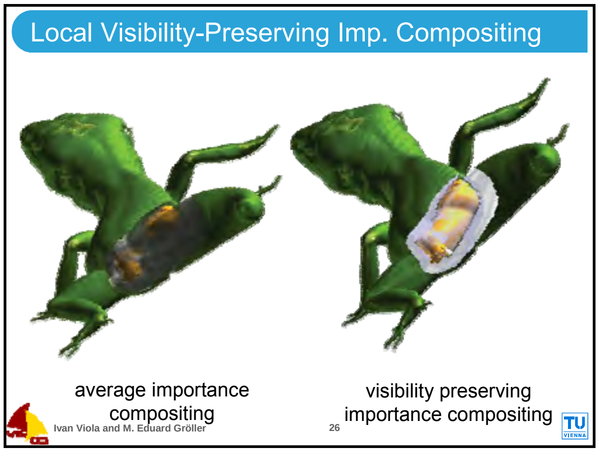
- Avg. imp. compositing preserves thickness
- Visibility of focus is not guaranteed
- Visibility validation useful for partial supression



Local Visibility-Preserving Imp. Compositing

$* \frac{\alpha_{avg}}{I_{norm}(o_0)}$

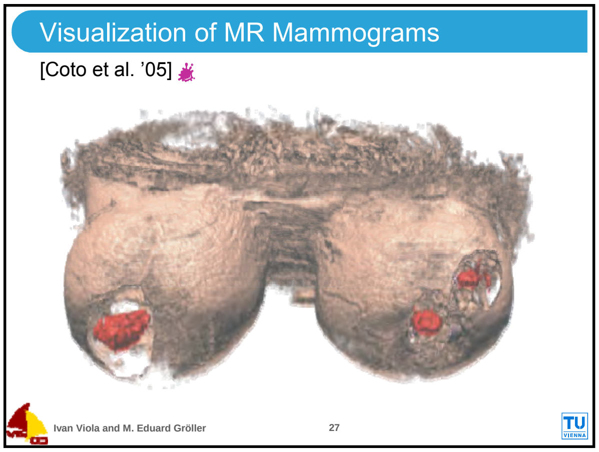
$\sum \alpha(o_0, r_{x0}) \rightarrow \alpha_{avg} \sim I_{norm}(o_0)$



Local Visibility-Preserving Imp. Compositing

average importance compositing

visibility preserving importance compositing

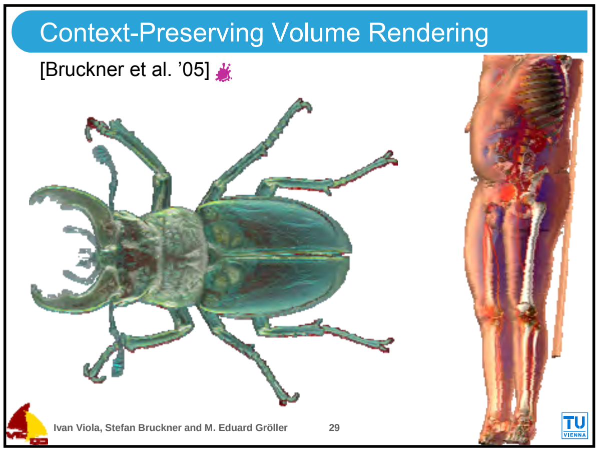


Visualization of MR Mammograms

[Coto et al. ’05]

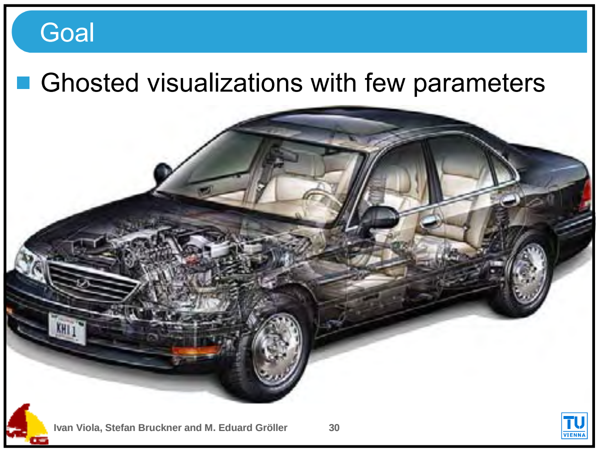
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations



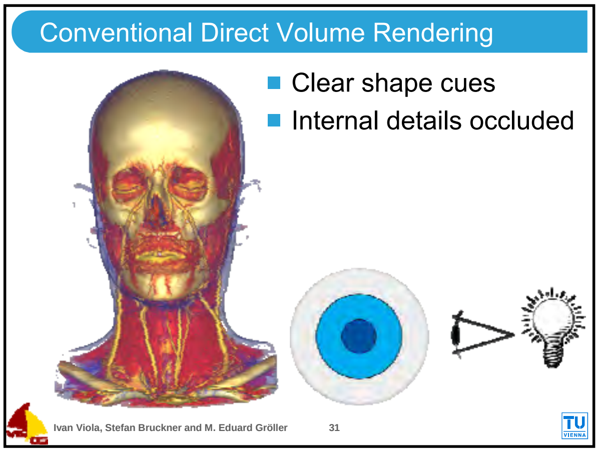
Context-Preserving Volume Rendering

[Bruckner et al. ’05]



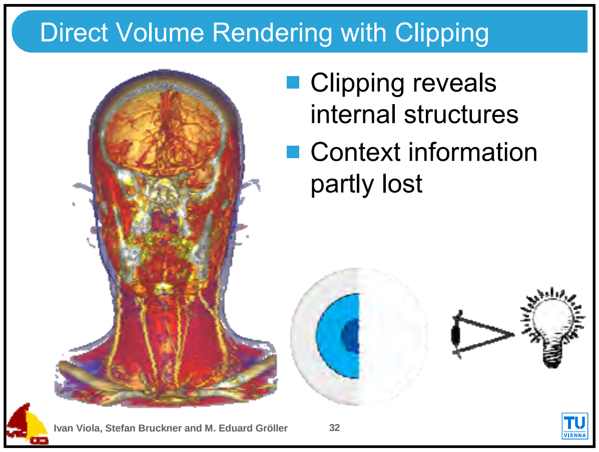
Goal

- Ghosted visualizations with few parameters



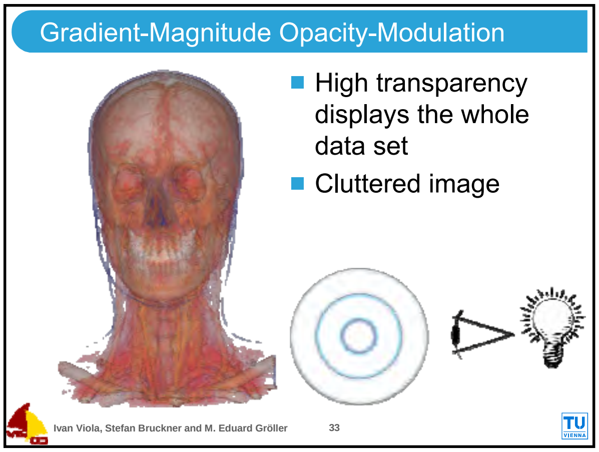
Conventional Direct Volume Rendering

- Clear shape cues
- Internal details occluded



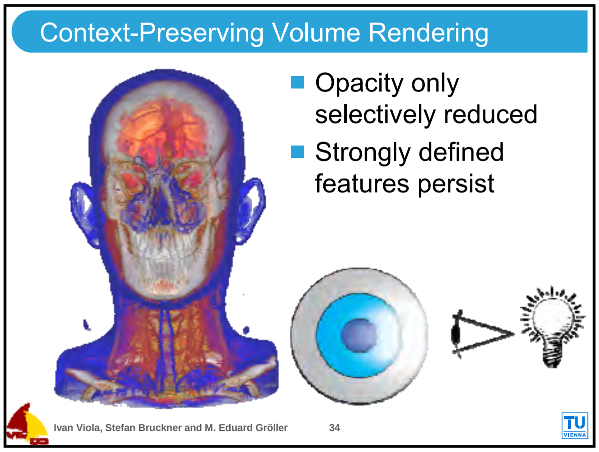
Direct Volume Rendering with Clipping

- Clipping reveals internal structures
- Context information partly lost



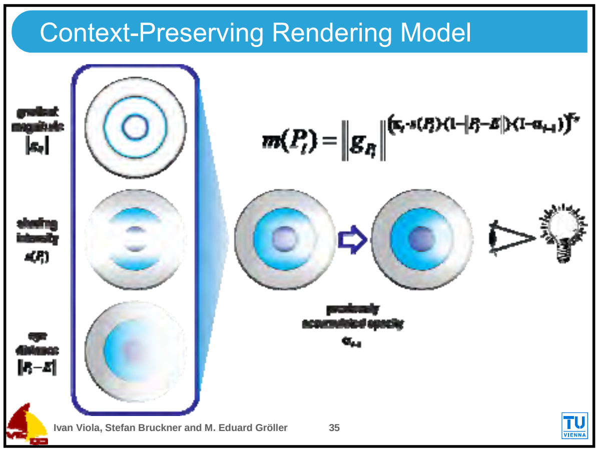
Gradient-Magnitude Opacity-Modulation

- High transparency displays the whole data set
- Cluttered image



Context-Preserving Volume Rendering

- Opacity only selectively reduced
- Strongly defined features persist



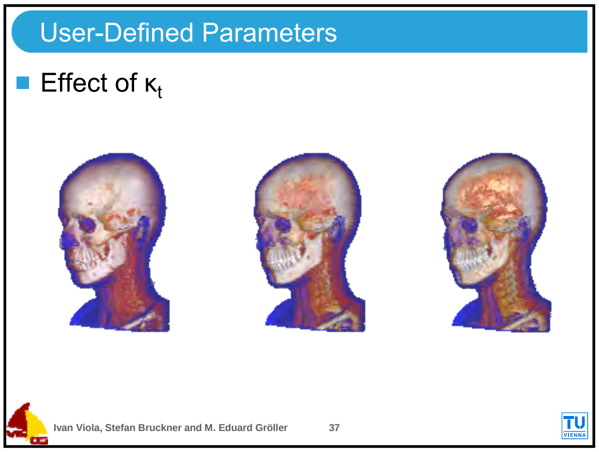
Context-Preserving Rendering Model

$$m(P_i) = \|g_{P_i}\|^{(\kappa_t \cdot s(P_i)(1-\|P_i - E\|)(1-\alpha_{i-1}))^{\kappa_s}}$$

gradient magnitude $|g_{P_i}|$
shading intensity $s(P_i)$
eye distance $|P_i - E|$
previously accumulated opacity α_{i-1}

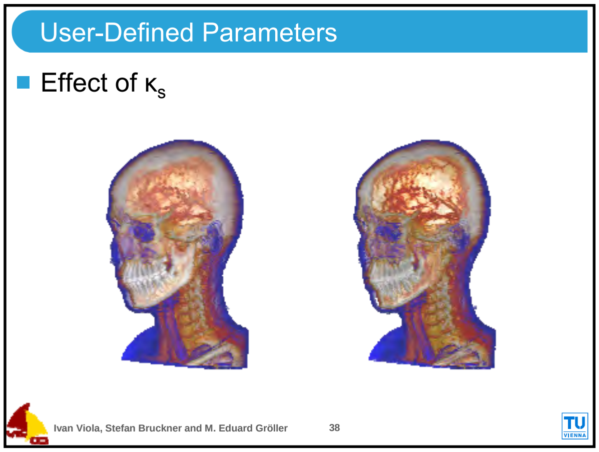
User-Defined Parameters

- κ_t controls *depth of cut*
  - Higher values → remove more occluding structures
  - Zero → results in conventional direct volume rendering
- κ_s controls *sharpness of cut*
  - Higher values → less smooth transition in opacity
  - Zero → pure gradient-magnitude opacity modulation



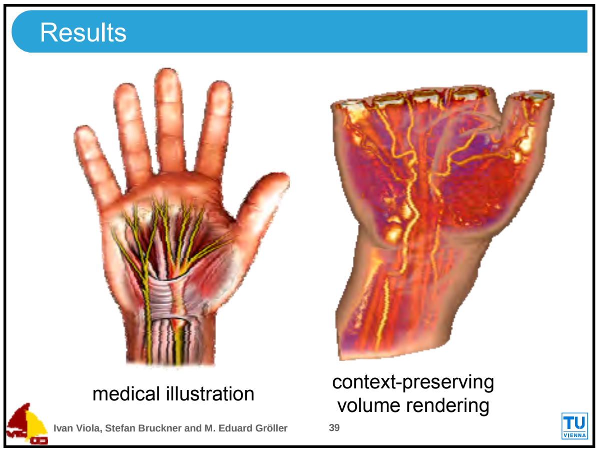
User-Defined Parameters

- Effect of κ_t



User-Defined Parameters

- Effect of κ_s



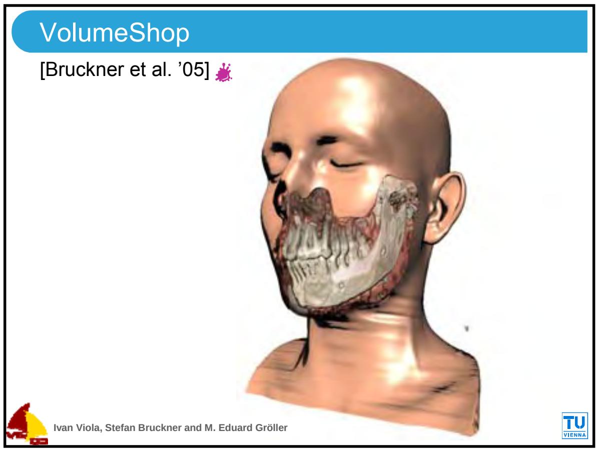
Results

medical illustration

context-preserving volume rendering

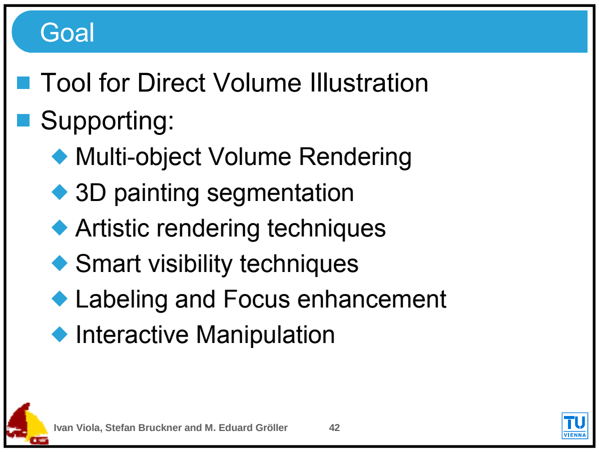
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations



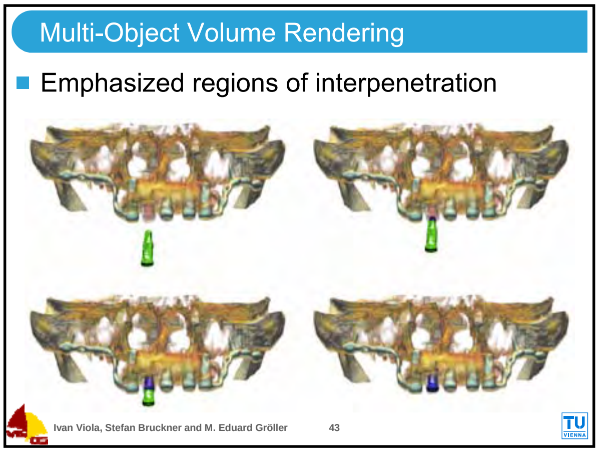
VolumeShop

[Bruckner et al. '05]



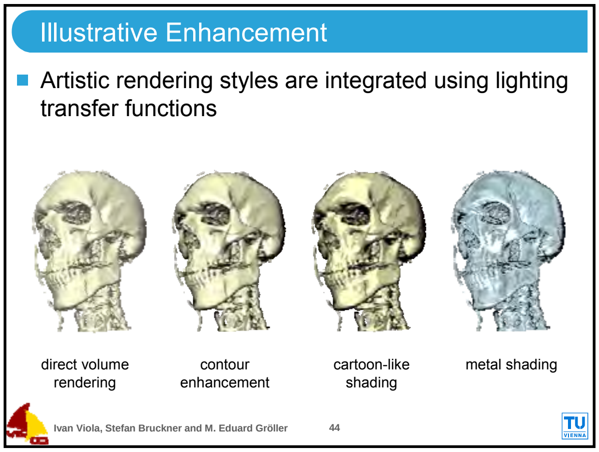
Goal

- Tool for Direct Volume Illustration
- Supporting:
  - Multi-object Volume Rendering
  - 3D painting segmentation
  - Artistic rendering techniques
  - Smart visibility techniques
  - Labeling and Focus enhancement
  - Interactive Manipulation



Multi-Object Volume Rendering

- Emphasized regions of interpenetration



Illustrative Enhancement

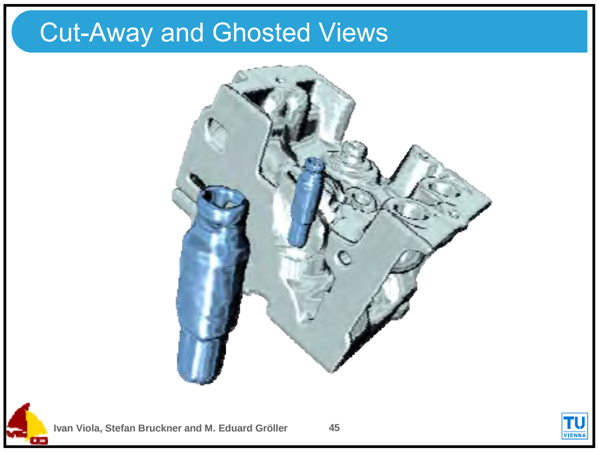
- Artistic rendering styles are integrated using lighting transfer functions

direct volume rendering

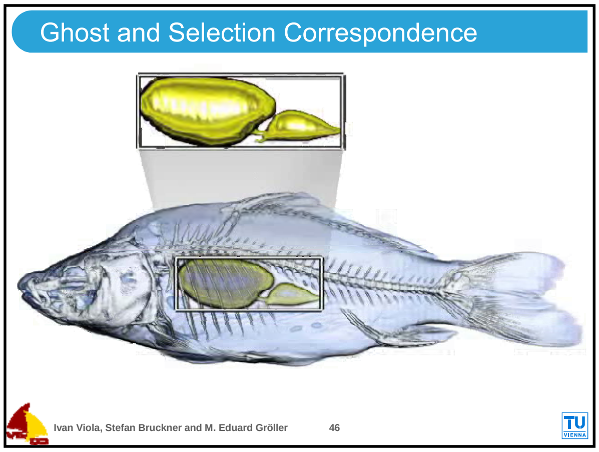
contour enhancement

cartoon-like shading

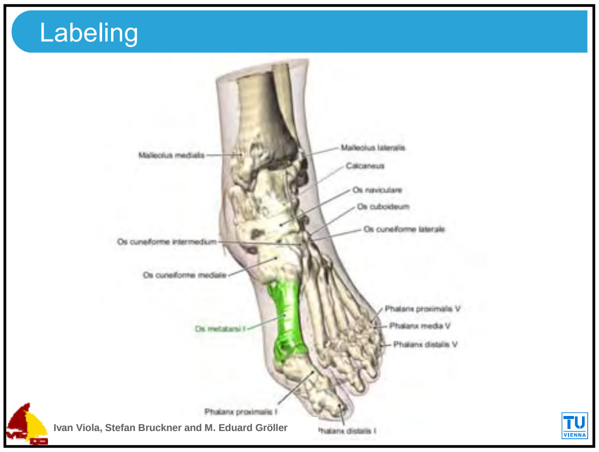
metal shading



Cut-Away and Ghosted Views



Ghost and Selection Correspondence



Labeling

Presentation and Availability

Friday 28 Oct. 9:30 Illustrative Visualization I
VolumeShop: An Interactive System for Direct Volume Illustration

Stefan Bruckner, M. Eduard Gröller

www.cg.tuwien.ac.at/volumeshop

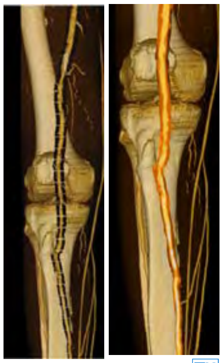
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations

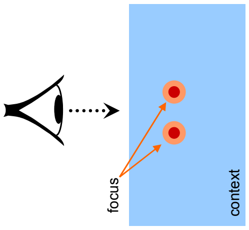
VesselGlyph

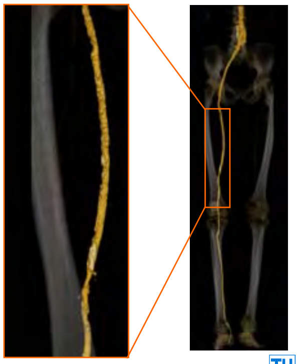
[Straka et al. '04]

- Focus+Context for angiographic data
- Combination of rendering techniques
  - Volume rendering
  - Curved-planar reformation
  - Maximum intensity projection
- Displays unoccluded objects in correct anatomic context

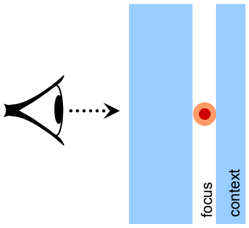


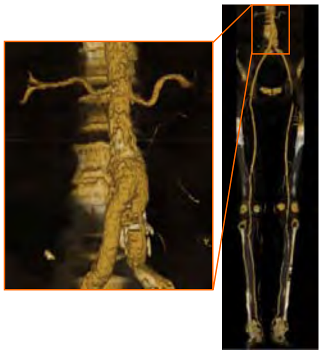
Tubular VesselGlyph



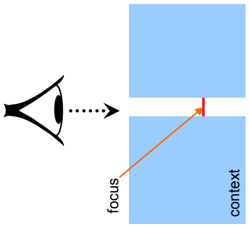


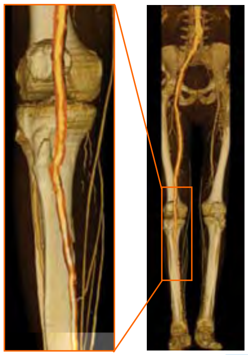
Thick-Slab VesselGlyph

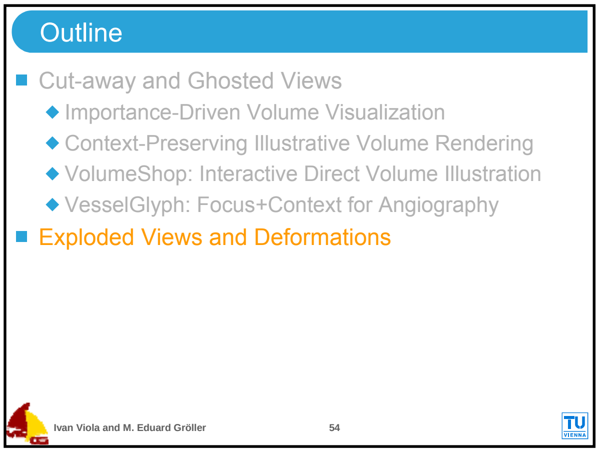




CPR-in-DVR VesselGlyph

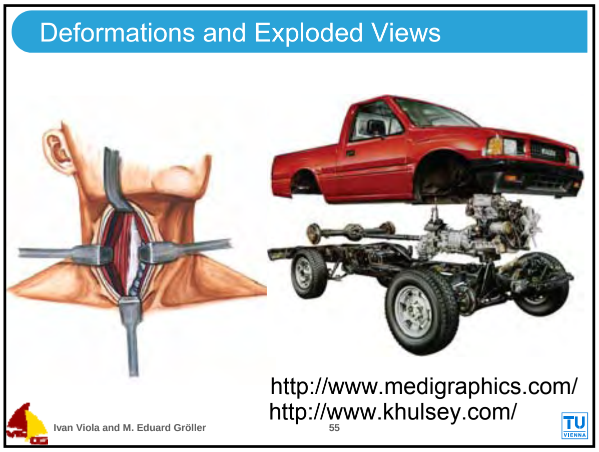






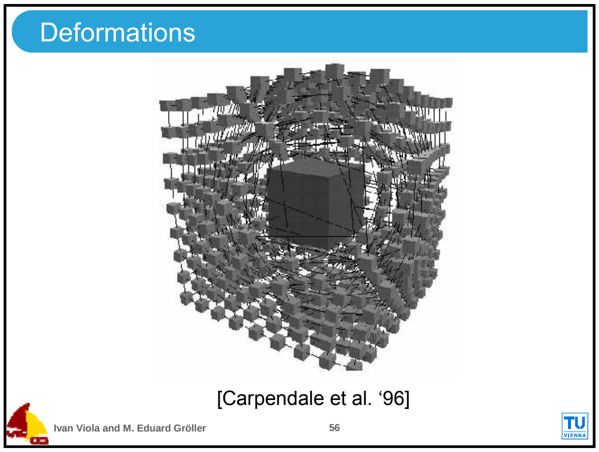
Outline

- Cut-away and Ghosted Views
  - Importance-Driven Volume Visualization
  - Context-Preserving Illustrative Volume Rendering
  - VolumeShop: Interactive Direct Volume Illustration
  - VesselGlyph: Focus+Context for Angiography
- Exploded Views and Deformations



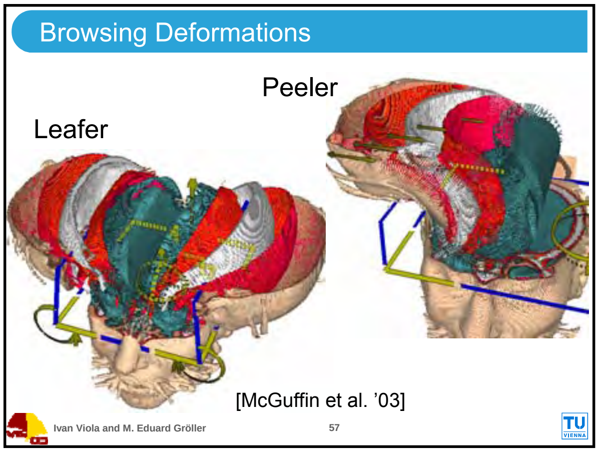
Deformations and Exploded Views

http://www.medigraphics.com/
http://www.khulsey.com/



Deformations

[Carpendale et al. ‘96]

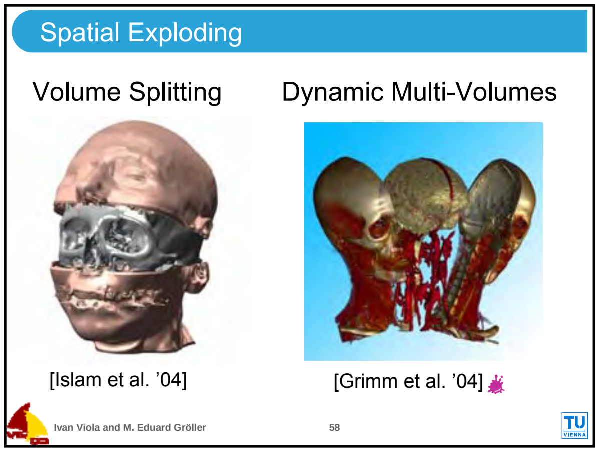


Browsing Deformations

Peeler

Leafer

[McGuffin et al. ’03]



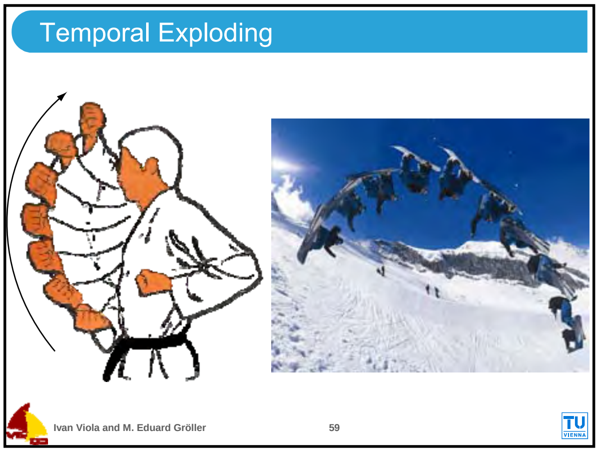
Spatial Exploding

Volume Splitting

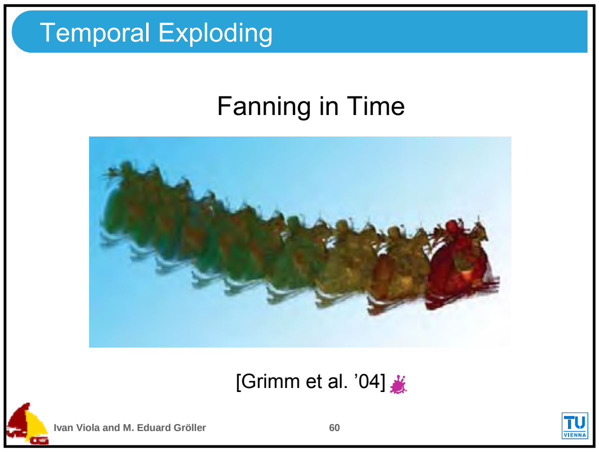
Dynamic Multi-Volumes

[Islam et al. ’04]

[Grimm et al. ’04]



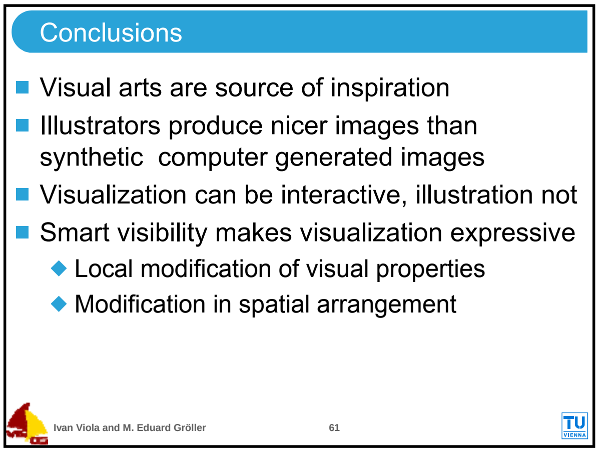
Temporal Exploding



Temporal Exploding

Fanning in Time

[Grimm et al. ’04]



Conclusions

- Visual arts are source of inspiration
- Illustrators produce nicer images than synthetic computer generated images
- Visualization can be interactive, illustration not
- Smart visibility makes visualization expressive
  - Local modification of visual properties
  - Modification in spatial arrangement

Thank you!

Thanks to: Stefan Bruckner, Matúš Straka

Importance-Driven Feature Enhancement in Volume Visualization

Ivan Viola, Armin Kanitsar, and M. Eduard Gröller, *Member, IEEE*

(Invited Paper)

Abstract— **This paper presents importance-driven feature enhancement as a technique for the automatic generation of cut-away and ghosted views out of volumetric data. The presented focus+context approach removes or suppresses less important parts of a scene to reveal more important underlying information. However, less important parts are fully visible in those regions, where important visual information is not lost, i.e., more relevant features are not occluded.**

Features within the volumetric data are first 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 (*importance compositing*).

The paper includes an extended discussion on several possible schemes for *levels of sparseness* specification. Furthermore different approaches to *importance compositing* are treated.

Index Terms— **view-dependent visualization, volume rendering, focus+context techniques, level-of-detail techniques, illustrative techniques**

I. Introduction

THE relevance of volume visualization in medical applications has been increasing over the last years. Three-dimensional visualization is becoming an essential tool for medical diagnosis and operation planning. Due to the rapid development of high-precision imaging modalities the amount of data is steadily increasing. The amount of relevant information is often relatively small as compared to the overall amount of acquired data. Therefore these small, interesting features have to be visually emphasized. Examples are tumors in the kidneys, lesions inside the liver, and lung nodules. 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 it is a focus+context task.

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. 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 these settings are applied globally. Beside this the 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 *importance-driven volume rendering* (IDVR) [28]. 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. With IDVR the interesting structures are clearly visible from different viewing angles. Occluding objects are rendered more sparsely or suppressed entirely.

The main contribution of this paper is importance-driven feature enhancement as an approach to automatic focus+context volume rendering. The proposed method overcomes the problem of occlusions within the volume,

The authors are with the Institute of Computer Graphics and Algorithms at the Vienna University of Technology, Austria. E-mail: {viola|kanitsar|meister}@cg.tuwien.ac.at

which happens when using any kind of view-independent classification. As opposed to previous approaches, the optical properties of the proposed technique are not constant for an entire object. Depending on the viewing situation, the estimated *level of sparseness* varies dynamically. 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 an anatomical illustration of the human abdomen [12] and a result of our technique is presented. 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.

The paper is organized as follows: Section II describes previous work related to importance-driven volume rendering. Section III explains the basic idea of the proposed model. Sections IV and V discuss the principal components of the model, i.e., *importance compositing* and *levels of sparseness*, and depict their impact on the resulting visualizations. In Section VI we describe the test datasets and show various rendering results applicable for medical and flow visualization. Finally we draw conclusions, summarize the paper in Section VII and propose future work in Section VIII.

II. Related Work

Scientific work related to our model can be divided into several categories. First, methods that use advanced and semi-automatic transfer function specification for feature enhancement are discussed. Our work enables automatic focus+context visualization, where the viewpoint information is taken into account. We therefore point out some previous focus+context approaches. Then, various rendering techniques for *levels of sparseness* specification are reviewed. Finally former work on incorporating cut-away concepts in visualization is mentioned.

Feature Classification: A typical feature classification in volume visualization is done through *transfer function* specification [18]. The transfer function with density as single input parameter is also denoted as one-dimensional transfer function. In recent years the idea of multi-dimensional transfer functions has been introduced. The multi-dimensional concept incorporates first and second derivatives of the density into the transfer function design [11], [17]. It is possible to assign optical properties based on gradient and curvature values, so for example object boundaries are classified differently than homogeneous regions. Taking into consideration first and second derivatives enables the semi-automatic generation of transfer functions [16]. An interesting approach was presented by Hauser and Mlejnek [9] for multi-dimensional 3D flow data. They use the *degree-of-interest* function that maps the user interest to optical properties. These concepts, however, define the representation globally, i.e., the visibility of important features is not guaranteed. The lack of adaptation of optical properties to the viewpoint settings is the main drawback of view-independent classifications.

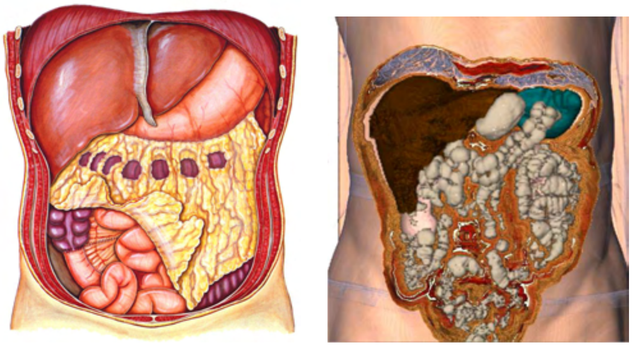


Fig. 1. Comparison between an artistic medical illustration of the abdomen (left) and our method (right).

Focus+Context Rendering: Visualization tasks frequently use the focus+context metaphor to clearly differentiate very relevant information from the context. Viewpoint-dependent distortion of three-dimensional data [3], for example, highlights data by dedicating more display space to it. Distortions are applied to abstract graphs in order to clearly see interesting graph nodes. An interesting idea is also to include the *distance to focal point* into the volume rendering pipeline [29]. The optical properties are changing according to the distance to the focal point. Using this technique several expressive focus+context effects can be achieved. Focus+context approaches use viewpoint-independent classification and therefore they have the same limitations as the feature classification methods discussed above.

Gaze-directed volume rendering [19] was an early approach in volume visualization, where the observer's viewing direction was taken into consideration. The motivation in this case was to increase the rendering performance instead of increasing 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.

Sparse Representation: The graphics community has been inspired by artists to represent features sparsely in order to exploit the human imagination. The display of contours is a popular method to thinly represent context information in volume visualization [4], [24]. Outlines are often sufficient to roughly understand the shape and can be combined with other rendering techniques such as direct volume rendering or maximum intensity projection [10]. To make a contour representation more expressive *suggestive contours* can be used [5]. The suggestive contours technique combines contours rendered from a particular viewpoint with contours from other viewpoints close to the current view. Also *pen-and-ink* techniques convey good shape information. Pen-and-ink styles in combination with traditional volume rendering have already been applied for focus+context rendering in volume visualization [26]. This is up to a certain degree similar to the combination of curvature-directed strokes with iso-surface rendering [14]. This approach was proposed for rendering structures that are nested within other objects. The interior structures are rendered fully opaque, while the enclosing objects are represented by a set of curvature-directed lines. Using stippling techniques for volume visualization is another example of inspiration from traditional illustration [21]. The visibility of interior structures can also be modified by dynamic changes in transparency of the outer shape. Recent work proposes to map transparency to the level of specular highlight [2]. This allows to see *inside* the volume in the areas of highlights. Dynamic transparency is also used in the user interface design [8].

Cut-Away Views: Cut-away illustrations are another way to represent nested objects. The popularity of this technique is demonstrated by the fact that it can be found in almost all books with technical or medical illustrations [13]. In volume visualization this technique is also known as *volume cutting* [23]. Automatic generation of cut-away images has been already researched in computer graphics [6], [7]. Straka et al. [25] are applying a cut-away technique for CT-Angiography. For visualizing complex dynamical systems, streamarrows were proposed by Löffelmann et al. [20]. 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.

Importance-driven volume rendering (IDVR) has been firstly introduced in our previous work [28]. We have presented a generalized model for view-dependent focus+context tasks in volume visualization. This paper extends the discussion on the key elements of the model: *object importance*, *levels of sparseness*, and *importance compositing*. Furthermore suggestions on feature definition are discussed to overcome the limited applicability to pre-segmented data. IDVR turns out to be helpful not only for changing viewpoints, but also in case of dynamic features.

III. Importance-Driven Volume Rendering

In volume visualization we are often dealing with the problem that interesting structures are partly or completely occluded by surrounding tissue. This is hard to resolve by traditional view-independent approaches, such as transfer function specification. We propose a viewpoint-dependent model that removes unwanted occlusions automatically and maximizes the information content in the final image.

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. In medical visualization features are often classified as particular organs. Such objects are defined by a segmentation process. Another way of feature classification can be through a spatial relationship within the volume or eventually 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. A detailed treatment of feature definition is outside the scope of this paper.

Traditionally features within the volume dataset are classified by optical properties such as color and opacity. We additionally assign 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 occluding 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 its features more clearly. All interesting structures are visible irrespective if they are covered 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, we use several ***levels of sparseness*** for each object. We do not assign a single optical characteristic, but several characteristics

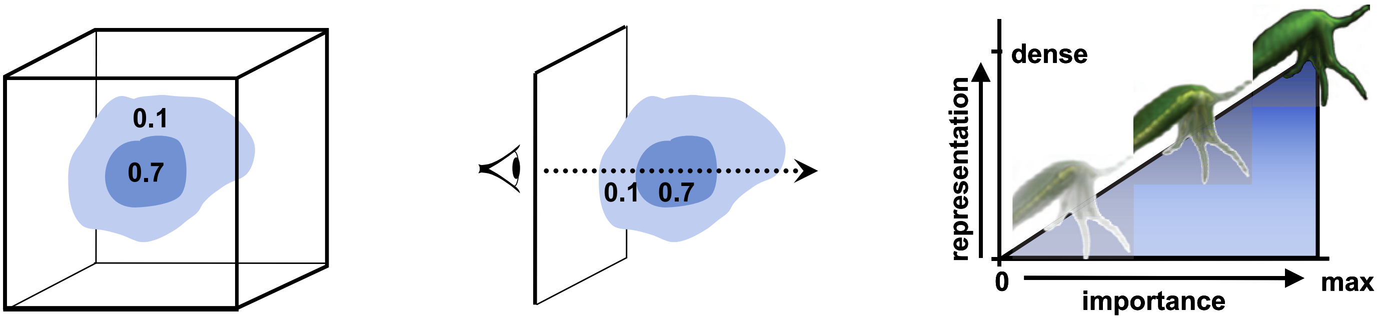


Fig. 2. Basic elements of importance-driven volume rendering: Volumetric features are classified by *importance* values (left). The volume is traversed in the *importance compositing* step (middle). Then *levels of sparseness* are chosen to enhance or suppress particular parts of the volume (right).

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 thus may continuously vary within a single object. Also depending on the viewpoint 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 the occlusion, 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.

The relationship between the above mentioned components is depicted in Figure 2. *Importance compositing* is done similar to the direct volume rendering (DVR) approach. For each ray the compositing step evaluates object occlusions and assigns the corresponding *level of sparseness* to each object. 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 2) in the area of the image which is covered by the focus object (importance 0.7).

The difference between traditional volume rendering and importance-driven volume rendering is clearly visible in Figure 3. The goal is to emphasize the inner organ as focus object. In the traditional approach it is necessary to reduce the opacity of occluding objects globally. Importance-driven rendering assigns a higher sparseness factor only to the area where occlusion occurs.

IV. 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 we will discuss three methods of importance compositing, which are inspired by compositing optical properties through ray casting of volume data.

A. Maximum Importance Projection

Maximum intensity projection (MIP) [22] is a simple and fast volume rendering approach. It is applicable for sparse data where important information has high intensity values such as contrast-media enhanced blood vessels. With MIP compositing reduces to selecting the highest intensity value along a ray. Intensities are encoded as gray values to produce the final image.

Analogous to MIP we propose 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,

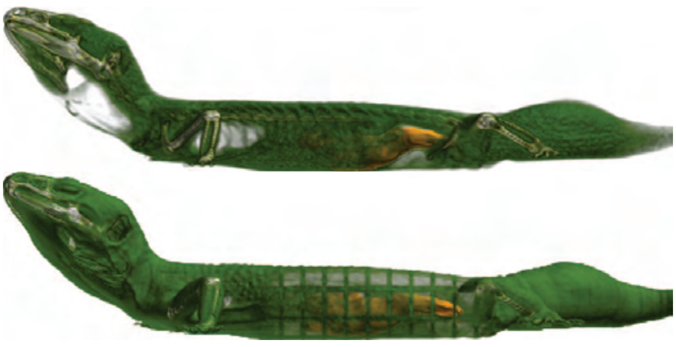


Fig. 3. Comparison between traditional volume rendering (top) and importance-driven volume rendering (bottom).

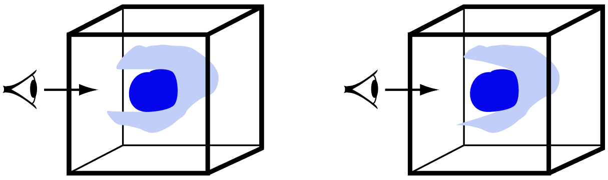


Fig. 4. Maximum Importance Projection. Illustration of cylindrical (left) and conical countersink (right).

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 we alleviate 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 countersink clipping geometry. Figure 4 illustrates the difference between the cylindrical and 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 5 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.

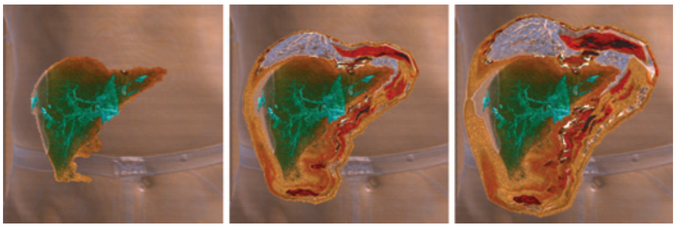


Fig. 5. Maximum importance projection (MImP). Cylindrical MImP (left) and conical MImP with different slope factors (middle and right).

The countersink geometry, respectively the ray starting points are computed from the footprint of the focus object. The footprint contains depth information of the focus object's *last hit* for each ray along the viewing direction. This information is used for performing the cut-out. For cylindrical MImP we simply skip ray samples that belong to the context object, until the focus object's last hit depth is reached. For the conical MImP we need to enlarge the footprint to build the conical shape. This is realized using image processing operators on the depth image, where the intensity encodes the depth of the entry point. The depth-footprint is processed by a 2D chamfer distance transform [1]. First e_{max} the highest depth value of the footprint is calculated. The starting points e_i of the rays that pierce the countersink are calculated from e_{max}, the slope s_c of the countersink, and distance d_i as shown in Equation 1:

$$e_i = e_{max} - d_i * s_c \tag{1}$$

d_i denotes the image space distance from pixel i to the footprint.

To correctly simulate the cut-out it is necessary to change the gradient vector of the ray entry points at the countersink geometry. Two components of the gradient are estimated from the gradient information of the 2D distance field. The z component is constant, i.e, the *slope* s_c of the countersink frustum.

B. 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 from 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:

$$S_o = \frac{I_o}{\sum_{i=1}^{n_r} I_i} \tag{2}$$

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 see a rough structure of the 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. At each pixel position we perform a lookup to each object footprint. 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 2).

The final image synthesis using AImC is an extension to 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 not readily apparent. In order to improve spatial perception we propose two methods to perform final importance-driven image synthesis using 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 the spatial perception we generate smooth transitions between different levels of sparseness. Before the levels of sparseness are computed for each object, we apply image processing operators 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 2. For each pixel we compute the footprint-weighted importance sum of all contributing objects. 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 suppressed.

Object-space AImC: To avoid suppression of context behind the focus object, we propose a more costly object-space approach. 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 6. 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.

The image synthesis of the object space approach is analogous to the conical MImP. In the case of the conical MImP the countersink geometry is used to estimate the starting position of the ray in order to perform the cutout. In object-space AImC the countersink 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 countersink region or not. The context outside the countersink is depicted with a more dense representation and inside a more sparse form is chosen.

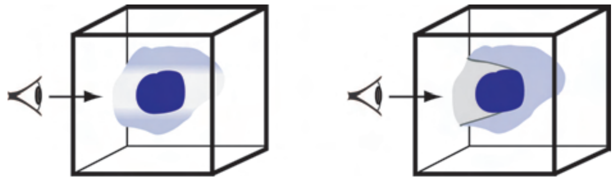


Fig. 6. Average importance compositing (AImC). The illustration depicts the difference between the image-space (left) and object space approach (right).

Results of each approach are shown in Figure 7. The images show the same dataset under different viewing angles. The top row illustrates the image-space approach and the bottom row the object-space approach.

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. The following Sub-section describes how to automatically overcome the problem a of varying thickness of the occluding object.

C. Visibility Preserving Importance Compositing

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 may be *thin* and samples can be represented more densely. Under a different viewing angle the context object may be *thicker* in front of the focus object. Therefore samples that belong to this area should be more transparent. This is illustrated in Figure 8. The suppression of the context varies according to the *thickness* of the context, therefore the visibility of the focus remains constant.

The values for levels of sparseness using VPImC are estimated in the same way as with AImC (Equation 2). In AImC the levels of sparseness are selected for each sample during ray-traversal. With VPImC we select an appropriate level of sparseness *after* the ray-traversal stage. The level of sparseness is determined as follows: The level of sparseness for the context object in front of the focus object shall be S_o. In VPImC the goal is to adjust the *average* accumulated opacity of 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

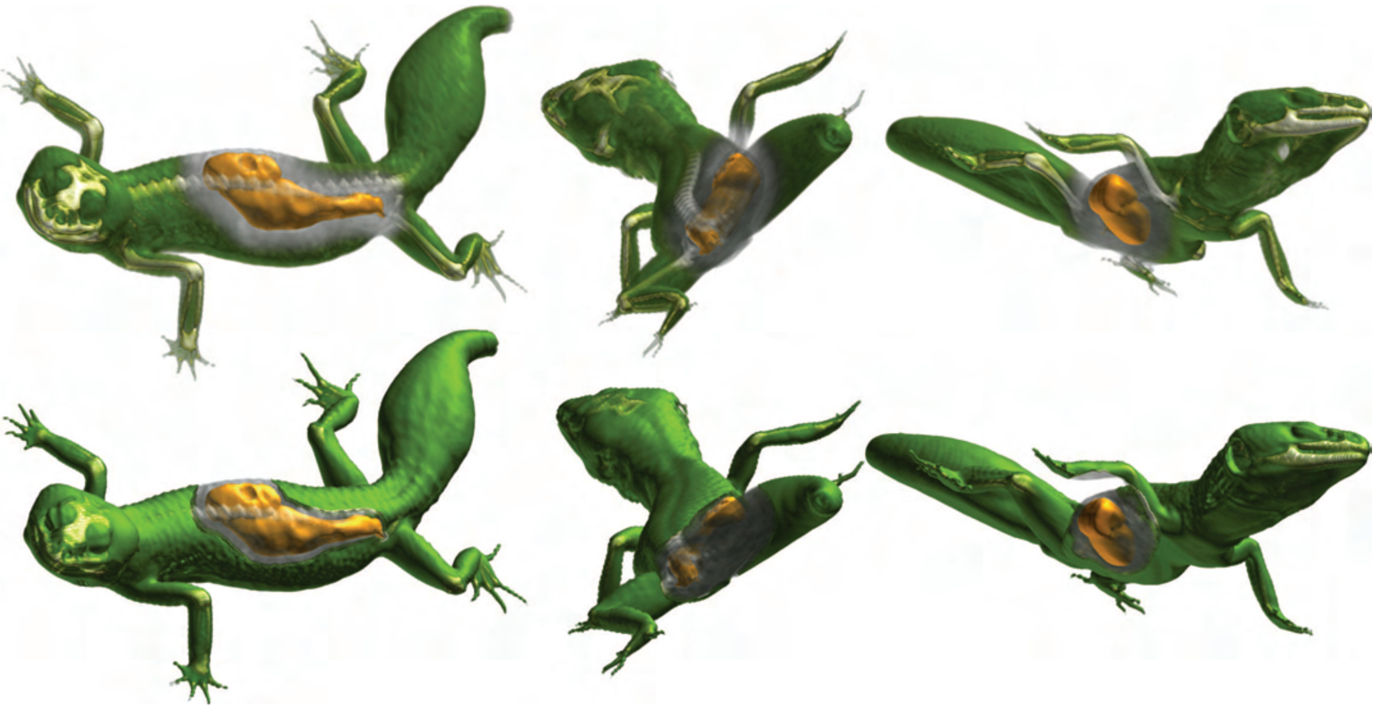


Fig. 7. Average importance compositing (AImC) is shown under different viewpoint settings in combination with modulating optical properties. The upper row shows image based AImC. The bottom row shows the object space approach with the same viewpoint settings.

compute the average per-ray opacity α_{avg}. To preserve the constant visibility of the focus object the average per-ray opacity has to be equal to S_o. Therefore for each ray, the accumulated opacity value is corrected. This is expressed by the Equation 3:

$$\alpha_{accum\_new}(x) = \alpha_{accum}(x)\frac{S_o}{\alpha_{avg}} \tag{3}$$

where α_{accum_new} is the modified accumulated opacity of the suppressed context part, α_{accum} 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 two level volume rendering [10] (2lVR). 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. Finally a combination of both context parts takes place in the *global compositing step*. A more detailed discussion on 2lVR follows in Sub-section V-D.

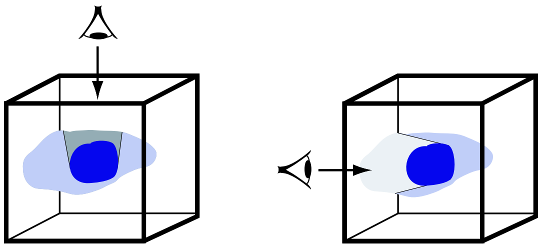


Fig. 8. Principle of visibility preserving importance compositing. Constant visibility allows a denser thin context (left) and requires a sparser thick context (right).

Figure 9 shows the results of the compositing under different viewpoint settings. It offers a comparison to the importance compositing technique shown in Figure 7 (described in Section IV-B). Especially in the middle images of Figure 7 a large occluding context region considerably reduces the visibility of the focus object. This is not the case in Figure 9 (middle image.)

V. Levels of Sparseness

Importance compositing determines for each part of an object its visibility in the rendered image. This is achieved by determining the appropriate level of sparseness. In the following four types of levels of sparseness are described. Three of those are depicted in Figure 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.

A. Color and Opacity Modulation

A direct control of optical properties is the first approach to modify the visual prominence of a particular

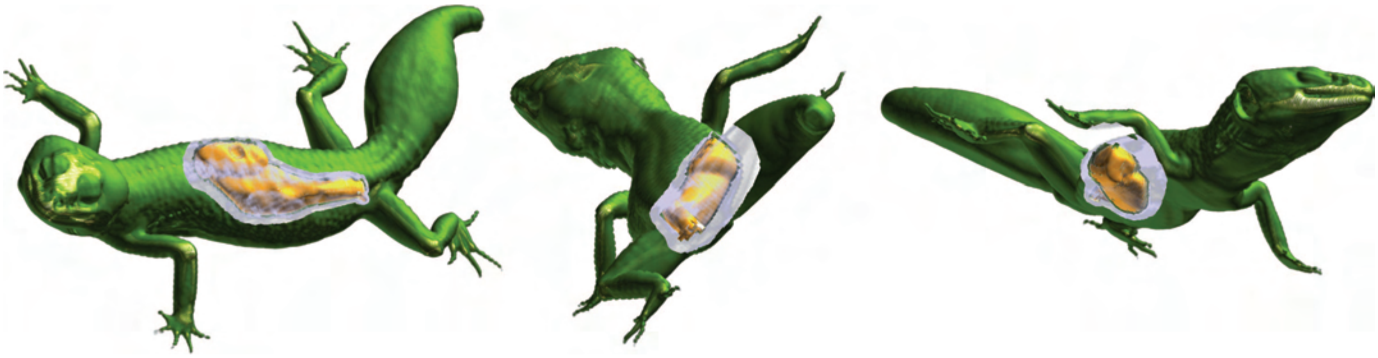


Fig. 9. Visibility preserving importance compositing (VPImC). The dataset is shown under different viewpoint settings with constant visibility of the focus object. Focus visibility is independent from the thickness of the occluding context part.

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 the level 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 10 (top).

B. 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. We use an analogous idea 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 10 (middle).

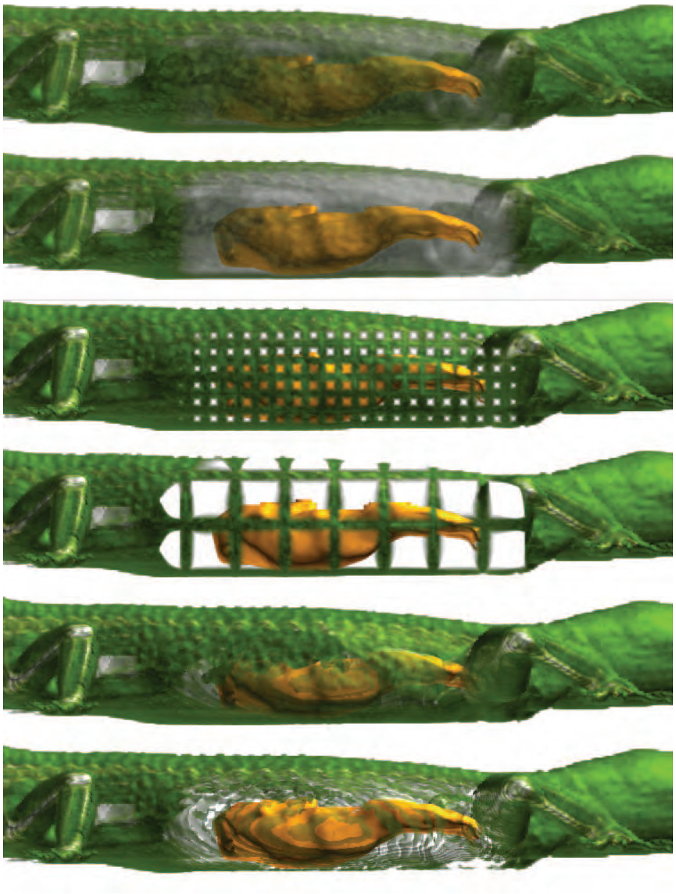


Fig. 10. Changing *levels of sparseness*. Top two rows: opacity modulation and color saturation modulation. Middle two rows: screen-door transparency. Bottom two rows: volume thinning. Images display *levels of sparseness* for the body with factors 0.75 and 0.25.

C. 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. Reducing the sparseness factor according to gradient magnitude has the effect that the volume is con-

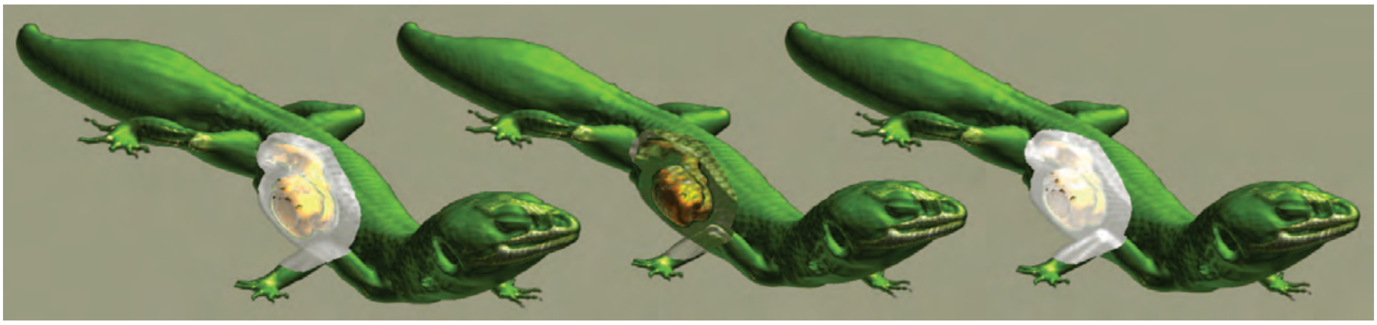


Fig. 11. Sparse and dense rendering styles: The occluding context object is rendered using summation (left), illustrative contour enhancement (middle), and maximum intensity projection (right).

tinuously 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 10 (bottom) illustrates visibility reduction through volume thinning.

D. 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 [10]. The difference between levels of sparseness based on object representations and rendering styles is shown in Figure 12.

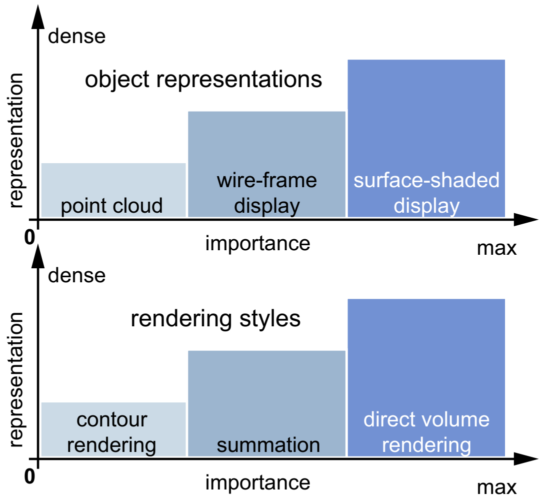


Fig. 12. Two types of *levels of sparseness*: based on object representations (top) and based on different rendering techniques (bottom).

Two level volume rendering (2lVR) is a technique to combine different volume rendering techniques. Well-known rendering techniques are direct volume rendering (DVR), MIP, summation (similar to X-Ray imaging), or illustrative rendering with contour enhancement [4], [24]. 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.

Results of using different rendering techniques as levels of sparseness are shown in Figure 11. Where no occlusion occurs the context information is rendered using standard 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.

VI. Results

We show results of our method on three datasets. The *Leopard Gecko* dataset is of resolution $512 \times 512 \times 87$. The *Monster Study* dataset has been downsampled to the half of its full resolution, i.e., $256 \times 256 \times 610$. Both datasets are using pre-segmented objects. The third dataset is a time-varying simulation of the hurricane Isabel. It is a three-dimensional flow dataset with multiple simulated properties including cloud moisture, precipitation, pressure, and temperature. This dataset is not pre-segmented, only the position of the hurricane eye

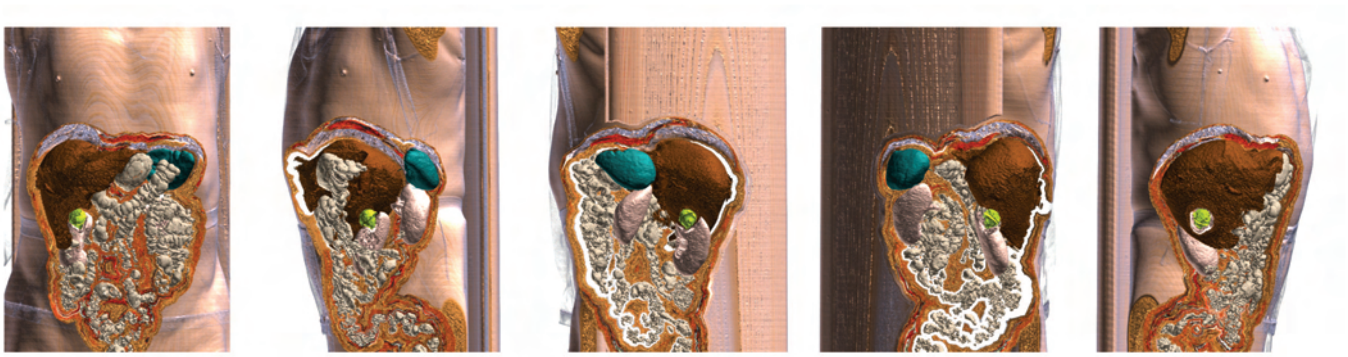


Fig. 13. The Monster Study dataset rendered using conical MImP. The highest importance is assigned to the tumor object (yellow). The organs of the abdomen are assigned a medium importance. The rest of the dataset has the lowest importance.

in object space is predefined. The simulation consists of 48 time steps.

Figure 13 shows the conical MImP of multiple abdomen organs 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. The organs of the abdomen have the same importance and therefore 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 relationship to other objects. In contrast to traditional approaches the occlusion problem is solved automatically.

Another example of a conical MImP is shown in Figure 14. It shows 6 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 countersink geometry.

This example also shows how to combine multiple scalar volumes using importance-driven volume rendering. 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. Thanks 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 volume rendering guarantees to visualize the important feature irrespective of viewpoint and feature position and shape.

The performance of the current implementation is not interactive. The goal of the implementation was to do a proof of concept rather than performance optimizations. The model was integrated as a plugin into the J-Vision [15] medical workstation.

To fully appreciate the strengths of importance-driven volume rendering viewpoint changes or dynamic scenes are essential. This is best illustrated with animation sequences, which are available at `http://www.cg.tuwien.ac.at/research/vis/adapt/2004_idvr/`.

VII. Summary and Conclusions

In this paper we have investigated importance-driven volume rendering as a view and feature dependent approach for automatic focus+context volume visualization. A new factor to the traditional volume rendering pipeline is introduced, i.e., the importance dimension. According to the importance and viewpoint settings each object is rendered in order to maximize the visual information. This method allows to see structures within the volume as dense as possible. A sparse representation is chosen only if other, more important structures are occluded.

Importance compositing defines how the occluding context information should be visualized. It can be simply cut away (MImP), or displayed using sparse representations. This approach can preserve either the thickness of the context object (AImC) or the visibility of the focus object (VPImC).

We have discussed four schemes for levels of sparseness. Levels of sparseness control the optical properties or the amount of visible elements of the volume. Smooth opacity changes work well in combination with desaturation. The amount of visible volume elements can be

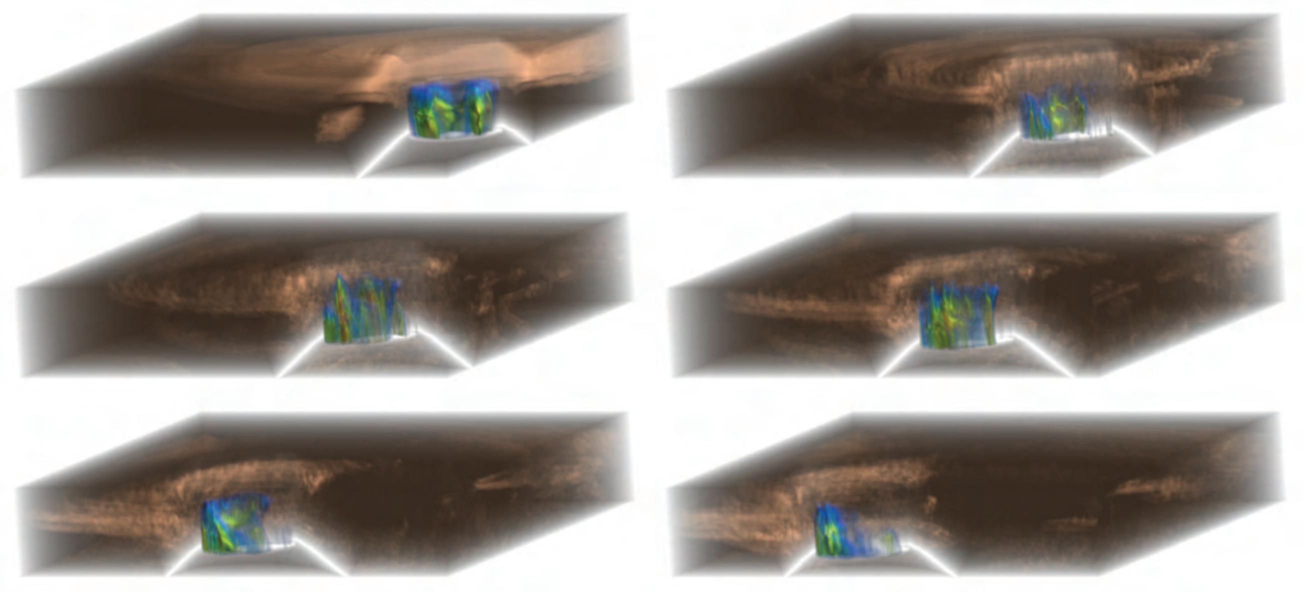


Fig. 14. Visualizing the simulation data of hurricane Isabel. Two different properties are visualized: total cloud and total precipitation mixing ratios. The interesting feature is the precipitation close to the eye of the hurricane. The context feature is the cloud mixing ratio. Images are showing 6 out of 48 time steps from left to right, top to bottom.

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 representation from most dense to most sparse ones. Another approach defines levels of sparseness through different rendering techniques.

VIII. Future Work

The paper opens multiple opportunities for possible research areas. An open issue is, how to do the feature selection and importance assignment automatically. Various automatic feature detection approaches can be integrated into the model to select the important features without additional user interaction.

The paper has presented various levels of sparseness schemes. The continuous transition from dense to sparse representations for volumetric data is a wide area of research. In polygonal rendering levels of sparseness are often used. The most sparse representation is a set of points, another representation is a wireframe display, and the most dense display is a surface representation. Volume graphics does not yet have such a variety, which shows the need for research in this direction.

The third factor of importance-driven volume rendering is importance compositing. The paper presents simple compositing schemes derived from ray-casting approaches. The next step are compositing schemes that incorporate first- and second order derivatives to preserve object boundaries. The parts with high first derivatives values can then be considered as more important and a dense representation is chosen there.

The conical MImP and other object-space importance compositing approaches are implementing the cut-out illustration technique to improve perception of the spatial relationships. More elaborate approaches for intelligent automatic cut-out generations need to be researched. In cut-away views sometimes the borderline of the cut-out regions is emphasized (e.g., through thick lines, or zig-zag lines). Automatically emphasizing these transition zones is also an open problem.

Each viewpoint brings out only a fraction of the entire information contained in the data set. How to estimate viewpoint entropy and how to automatically determine optimal viewpoints [27] is another, not yet researched, area in volume visualization.

IX. Acknowledgments

The work presented in this publication 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. See http://www.cg.tuwien.ac.at/research/vis/adapt for further information on this project.

The *Monster Study* dataset is courtesy of Tiani Medgraph and the *Leopard Gecko* is courtesy of University of the Veterinary Medicine Vienna. The hurricane Isabel data is courtesy of the National Center for Atmospheric Research in the United States. The medical illustration of the human abdomen is courtesy of Howell MediGraphics [12].

References

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

[2] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving volume rendering. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2005.

[3] M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings of IEEE Symposium on Information Visualization '96*, pages 46–53, 1996.

[4] 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.

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

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

[7] 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.

[8] C. Gutwin, J. Dyck, and Ch. Fedak. The effects of dynamic transparency on targeting performance. In *Proceedings of Graphics Interface '03*, pages 105–112, 2003.

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

[10] 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.

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

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

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

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

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

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

[17] J. Kniss, G. Kindlmann, and Ch. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization '01*, pages 255–262, 2001.

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

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

[20] 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.

[21] A. Lu, C. Morris, D. Ebert, P. Rheingans, and C. Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proceedings of IEEE Visualization '02*, pages 211–218, 2002.

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

[23] B. Pflesser, U. Tiede, and K. H. Höhne. Towards realistic visualization for surgery rehearsal. In *Computer Vision, Virtual Reality and Robotics in Medicine, Proc. CVRMed '95*, pages 487–491, 1995.

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

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

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

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

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

[29] 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.

Ivan Viola graduated in 2002 from the Vienna University of Technology, Austria, as a Dipl.-Ing. (MSc) in the field of computer graphics and visualization. Since then he is a PhD student and research associate in the ADAPT research project in the field of medical visualization and real-time volume visualization. His interests are efficient visualization in terms of quality, performance, and visual information.

Armin Kanitsar is key developer in the 3D department of Tiani Medgraph and research associate at the Institute of Computer Graphics and Algorithms (ICGA), Vienna University of Technology. He received his PhD in 2004 for his thesis 'Curved Planar Reformation for Vessel Visualization'. His main research interests include volume visualization, medical visualization, and image processing.

M. Eduard Gröller is associate professor at the Institute of Computer Graphics and Algorithms (ICGA), Vienna University of Technology. In 1993 he received his PhD from the same university. His research interests include computer graphics, flow visualization, volume visualization, and medical visualization. He is heading the visualization group at ICGA.

EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005)
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

Illustrative Context-Preserving Volume Rendering

Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M. Eduard Gröller

{bruckner | grimm | kanitsar | groeller}@cg.tuwien.ac.at
Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria

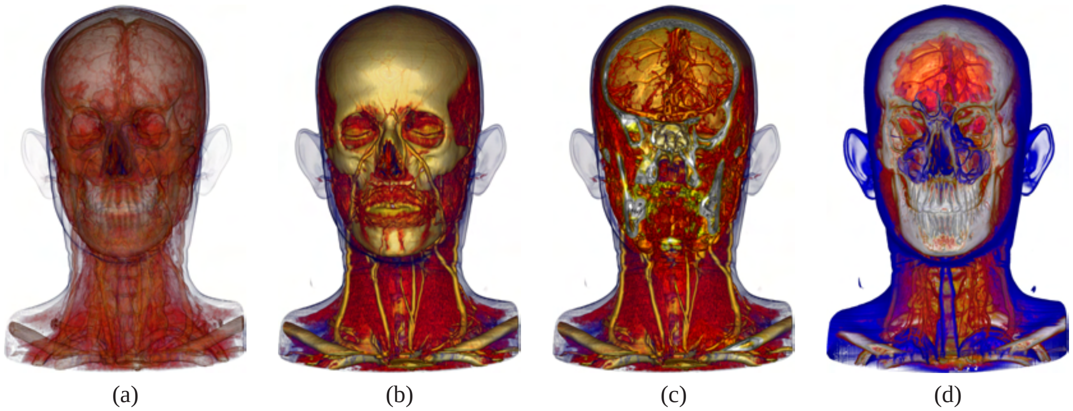


Figure 1: *Contrast-enhanced CT angiography data set.* ***(a)*** *Gradient-magnitude opacity-modulation.* ***(b)*** *Direct volume rendering.* ***(c)*** *Direct volume rendering with cutting plane.* ***(d)*** *Context-preserving volume rendering.*

Abstract

In volume rendering it is very difficult to simultaneously visualize interior and exterior structures while preserving clear shape cues. Very transparent transfer functions produce cluttered images with many overlapping structures, while clipping techniques completely remove possibly important context information. In this paper we present a new model for volume rendering, inspired by techniques from illustration that provides a means of interactively inspecting the interior of a volumetric data set in a feature-driven way which retains context information. The context-preserving volume rendering model uses a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity to selectively reduce the opacity in less important data regions. It is controlled by two user-specified parameters. This new method represents an alternative to conventional clipping techniques, shares their easy and intuitive user control, but does not suffer from the drawback of missing context information.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Theoretically, in direct volume rendering every single sample contributes to the final image allowing simultaneous visualization of surfaces and internal structures. However, in practice it is rather difficult and time-demanding to specify an appropriate transfer function. Due to the exponential attenuation, objects occluded by other semi-transparent objects are difficult to recognize. Furthermore, reducing opacities also results in reduced shading contributions per sample which causes a loss of shape cues, as shown in Figure 1 (a). One approach to overcome this difficulty is to use steep transfer functions, i.e., those with rapidly increasing opac-

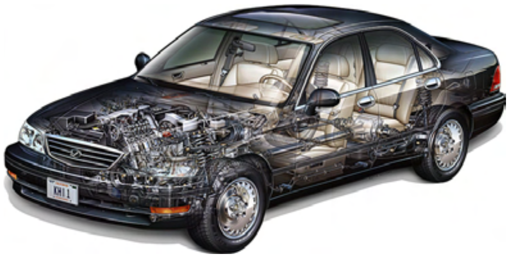


Image courtesy of Kevin Hulsey Illustration
Copyright ©2004 Kevin Hulsey Illustration, Inc. All rights reserved.
http://www.khulsey.com

Figure 2: *Technical illustration using ghosting to display interior structures.*

ities at particular value ranges of interest. This may increase depth cues by creating the appearance of surfaces within the volume, but it does so by hiding all information in some regions of the volume, sacrificing a key advantage of volume rendering. Figure 1 (b) shows an example.

Thus, volume clipping usually plays a decisive role in understanding 3D volumetric data sets. It allows us to cut away selected parts of the volume, based on the position of voxels in the data set. Very often, clipping is the only way to uncover important, otherwise hidden, details of a data set. However, as these clipping operations are generally not data-aware they do not take into account features of the volume. Consequently, clipping can also remove important context information leading to confusing and partly misleading result images as displayed in Figure 1 (c).

In order to resolve these issues, we propose to only suppress regions which do not contain strong features when browsing through the volume. Our idea is based on the observation that large regions of high lighting intensity usually correspond to rather homogenous areas which do not contain characteristic features. While the *position* and *shape* of specular highlights, for example, give good cues for perceiving the curvature of a surface, the area inside the highlight could also be used to display other information. Thus, we propose to make this area transparent allowing the user to see the interior of the volume.

In illustration, when artists want to visualize specific internal structures as well as the exterior, the use of *ghosting* is common where less significant items are reduced to an illusion of transparency. For example, rather flat surfaces are faded from opaque to transparent to reveal the interior of an object. Detailed structures are still displayed with high opacity, as shown in the technical illustration in Figure 2. The goal of this illustration technique is to provide enough hints to enable viewers to mentally complete the partly removed structures.

Using our idea of lighting-driven feature classification, we can easily mimic this artistic concept in an illustrative volume rendering model. Our approach allows context-preserving clipping by adjusting the model parameters. The approach of clipping planes is extended to allow feature-aware clipping. Figure 1 (d) shows a result achieved with out method - the blood vessels inside the head are revealed while preserving context information.

2. Related Work

We draw our inspiration mainly from three areas of research in volume visualization: volume clipping, illustrative volume rendering, and transfer functions. The term illustrative volume rendering refers to approaches dealing with feature enhancement, alternative shading and compositing models, and related matters, which are usually called non-photorealistic volume rendering methods. We believe that the adjective illustrative more accurately describes the nature of many of these techniques.

Volume Clipping. There are approaches which try to remedy the deficiencies of simple clipping planes by using more complex clipping geometry. Weiskopf et al. [WEE02, WEE03] presented techniques for interactively applying arbitrary convex and concave clipping objects. Konrad-Verse et al. [KVPL04] use a mesh which can be flexibly deformed by the user with an adjustable sphere of influence. Zhou et al. [ZDT04] propose the use of distance to emphasize and de-emphasize different regions. They use the distance from the eye point to directly modulate the opacity at each sample position. Thus, their approach can be seen as a generalization of binary clipping. Volume sculpting, proposed by Wang and Kaufman [WK95], enables interactive carving of volumetric data. An automated way of performing clipping operations has been presented by Viola et al. [VKG04]. Inspired by cutaway views, which are commonly used in technical illustrations, they apply different compositing strategies to prevent an object from being occluded by a less important object.

Illustrative Volume Rendering. A common illustrative method is gradient-magnitude opacity-modulation. Levoy [Lev88] proposed to modulate the opacity at a sample position using the magnitude of the local gradient. This is an effective way to enhance surfaces in volume rendering, as homogeneous regions are suppressed. Based on this idea, Ebert and Rheingans [ER00, RE01] presented several illustrative techniques which enhance features and add depth and orientation cues. They also propose to locally apply these methods for regional enhancement. Using similar methods, Lu et al. [LME*02] developed an interactive direct volume illustration system that simulates traditional stipple drawing. Csébfalvi et al. [CMH*01] visualize object contours based on the magnitude of local gradients as well as on the angle between viewing direction and gradient vector using depth-shaded maximum intensity projection. The concept of two-level volume rendering, proposed by Hauser et al. [HMBG01], allows focus+context visualization of volume data. Different rendering styles, such as direct volume rendering and maximum intensity projection, are used to emphasize objects of interest while still displaying the remaining data as context.

Transfer Functions. Multi-dimensional transfer functions have been proposed to extend the classification space and thus allow better selection of features. These transfer functions take into account derivative information. For example, Kindlmann et al. [KWTM03] use curvature information to achieve illustrative effects, such as ridge and valley enhancement. Lum and Ma [LM04] assign colors and opacities as well as parameters of the illumination model through a transfer function look up. They apply a two-dimensional transfer function to emphasize material boundaries using illumination. While there are many advantages of multi-dimensional approaches, they also have some issues: it is, for example, difficult to develop a simple and intuitive user interface which allows the specification of multi-dimensional transfer functions. One possible solution was presented by Kniss et al. [KKH01] who introduce probing and classification widgets.

The main contribution of this paper is a new illustrative volume rendering model which incorporates the functionality of clipping planes in a feature-aware manner. It includes concepts from artistic illustration to enhance the information content of the image. Cumbersome transfer function specification is simplified and no segmentation is required as features are classified implicitly. Thus, our approach is especially well-suited for interactive exploration.

3. The Context-Preserving Volume Rendering Model

Lighting plays a critically important role in illustrating surfaces. In particular, lighting variations provide visual cues regarding surface orientation. This is acknowledged by Lum and Ma [LM04], who highlight material boundaries by using two-dimensional lighting transfer functions. In contrast, in our approach the lighting intensity serves as an input to a function which varies the opacity based on this information, i.e., we use the result of the shading intensity function to classify features.

3.1. Background

We assume a continuous volumetric scalar field $f(P_i)$. A sample at position P_i is denoted by f_{P_i}. We denote the gradient at position P_i by $g_{P_i} = \nabla f(P_i)$. We use $\hat{g}_{P_i}$ for the normalized gradient and $\|g_{P_i}\|$ for the gradient magnitude normalized to the interval $[0..1]$, where zero corresponds to the lowest gradient magnitude and one corresponds to the highest gradient magnitude in the data set.

A discrete approximation of the volume rendering integral along a viewing ray uses the front-to-back formulation of the over operator to compute the opacity α_i and the color c_i at each step along the ray:

$$\begin{aligned} \alpha_i &= \alpha_{i-1} + \alpha(P_i) \cdot (1 - \alpha_{i-1}) \\ c_i &= c_{i-1} + c(P_i) \cdot \alpha(P_i) \cdot (1 - \alpha_{i-1}) \end{aligned} \tag{1}$$

$\alpha(P_i)$ and $c(P_i)$ are the opacity and color contributions at position P_i, α_{i-1} and c_{i-1} are the previously accumulated values for opacity and color.

For conventional direct volume rendering using shading, $\alpha(P_i)$ and $c(P_i)$ are defined as follows:

$$\begin{aligned} \alpha(P_i) &= \alpha_{tf}(f_{P_i}) \\ c(P_i) &= c_{tf}(f_{P_i}) \cdot s(P_i) \end{aligned} \tag{2}$$

α_{tf} and c_{tf} are the opacity and color transfer functions; they map an opacity and color to each scalar value in the volumetric function. $s(P_i)$ is the value of the shading intensity at the sample position.

For the Phong-Blinn model using one directional light source, $s(P_i)$ is:

$$s(P_i) = c_d \cdot \left\|\hat{L} \cdot \hat{g}_{P_i}\right\| + c_s \cdot \left(\left\|\hat{H} \cdot \hat{g}_{P_i}\right\|\right)^{c_e} + c_a \tag{3}$$

c_d, c_s, and c_a are the diffuse, specular, and ambient lighting coefficients, respectively, and c_e is the specular exponent. $\hat{L}$ is the normalized light vector and $\hat{H}$ is the normalized half-way vector. For conventional direct volume rendering, the opacity at point P_i is only determined by the scalar value f_{P_i} and the opacity transfer function. Gradient-magnitude opacity-modulation additionally scales the opacity by the gradient magnitude, causing an enhanced display of boundaries. Thus, for gradient-magnitude opacity-modulation $\alpha(P_i)$ is defined in the following way:

$$\alpha(P_i) = \alpha_{tf}(f_{P_i}) \cdot \|g_{P_i}\| \tag{4}$$

3.2. Model

In direct volume rendering, the illumination intensity normally does not influence the opacity at a sample position. Large regions of highly illuminated material normally correspond to rather flat surfaces that are oriented towards the light source. Our idea is to reduce the opacity in these regions. Regions, on the other hand, which receive less lighting, for example contours, will remain visible.

Therefore we choose to use the result of the shading intensity function $s(P_i)$ for opacity modulation. Furthermore, as we want to mimic - to a certain extent - the look and feel of a clipping plane, we also want to take the distance to the eye point into account. For viewing rays having already accumulated a lot of opacity, we want to reduce the attenuation by our model. The total effect should also take the gradient magnitude into account.

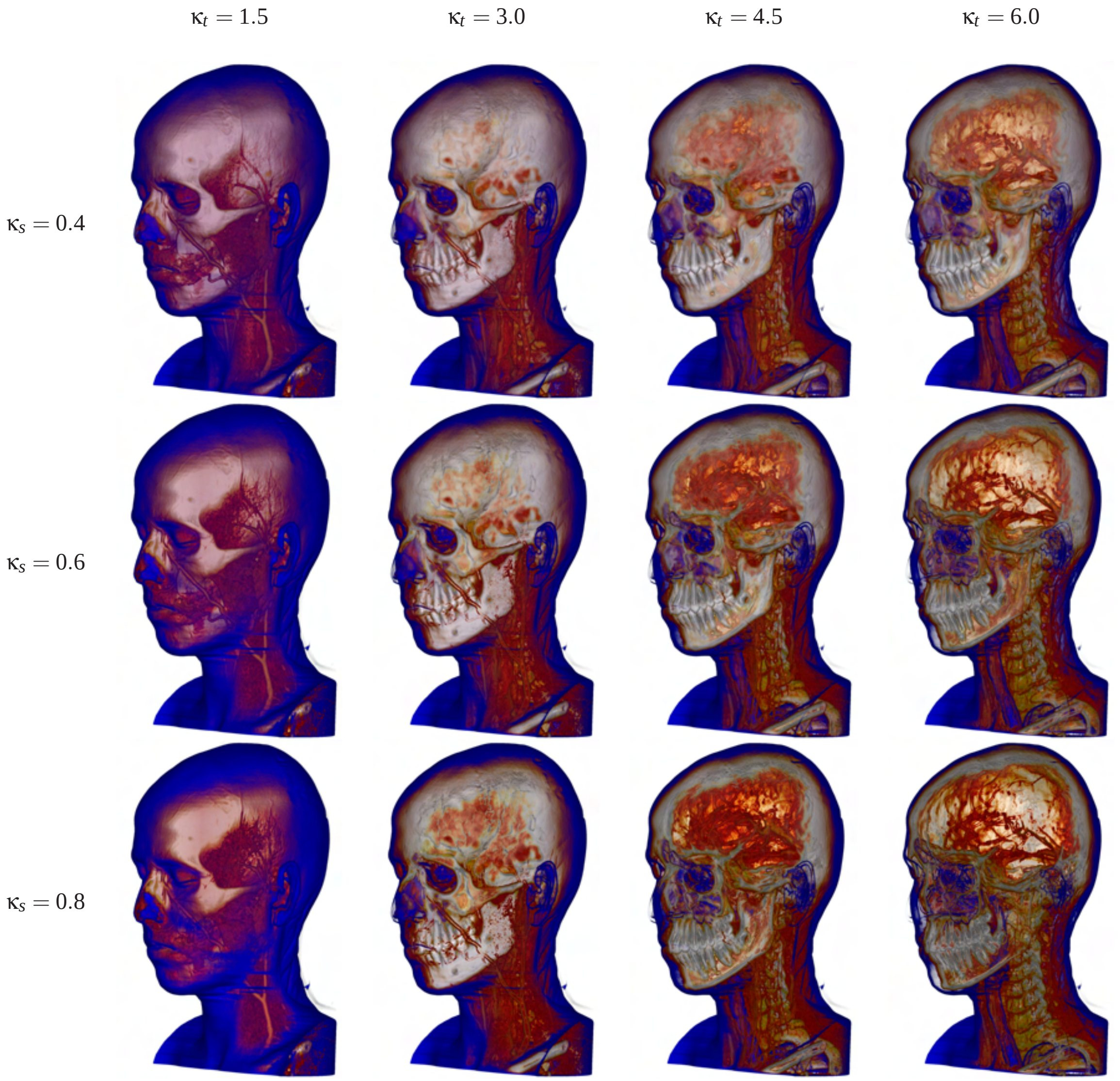


Figure 3: *Context-preserving volume rendering of a contrast-enhanced CT angiography data set using different values for κ_t and κ_s. Columns have the same κ_t value and rows have the same κ_s value.*

These requirements lead us to the following equation for the opacity at each sample position P_i:

$$\alpha(P_i) = \alpha_{tf}(f_{P_i}) \cdot \|g_{P_i}\|^{(\kappa_t \cdot s(P_i) \cdot (1-\|P_i - E\|) \cdot (1-\alpha_{i-1}))^{\kappa_s}} \tag{5}$$

$\|g_{P_i}\|$ is the gradient magnitude and $s(P_i)$ is the shading intensity at the current sample position. A high value of $s(P_i)$ indicates a highlight region and decreases opacity. The term $\|P_i - E\|$ is the distance of the current sample position to the eye point, normalized to the range $[0..1]$, where zero corresponds to the sample position closest to the eye point and one corresponds to the sample position farthest from the eye point. Thus, the effect of our model will decrease as distance increases. Due to the term $1 - \alpha_{i-1}$ structures located behind semi-transparent regions will appear more opaque. The influence of the product of these three components is controlled by the two user-specified parameters κ_t and κ_s. The parameter κ_t roughly corresponds - due to the position-dependent term $1 - \|P_i - E\|$ - to the depth of a clipping plane, i.e., higher values reveal more of the interior of the volume. This is the one parameter the user will modify interactively to explore the data set. The effect of modifying κ_s is less pronounced - its purpose is to allow control of the sharpness of the cut. Higher values will result in very sharp cuts, while lower values produce smoother transitions.

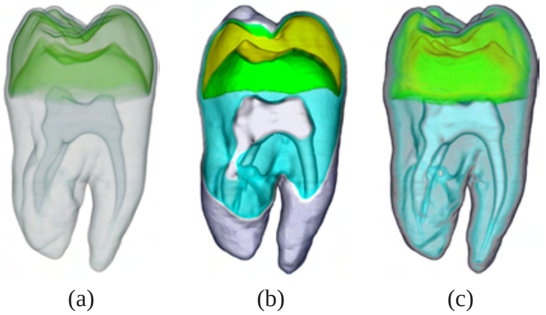

(a) (b) (c)

Figure 4: *CT scan of a tooth rendered with three different techniques.* ***(a)*** *Gradient-magnitude opacity-modulation.* ***(b)*** *Direct volume rendering with clipping plane.* ***(c)*** *Context-preserving volume rendering.*

Figure 3 shows results for different settings of κ_t and κ_s. It can be seen that as κ_t increases more of the interior of the head is revealed - structures on the outside are more and more reduced to the strongest features. An increase in κ_s causes a sharper transition between attenuated and visible regions. A further property of this model is that it is a unifying extension of both direct volume rendering and gradient-magnitude opacity-modulation. If κ_t is set to zero, the opacity remains unmodified and normal direct volume rendering is performed. Likewise, when κ_s is set to zero, the opacity is directly modulated by the gradient magnitude.

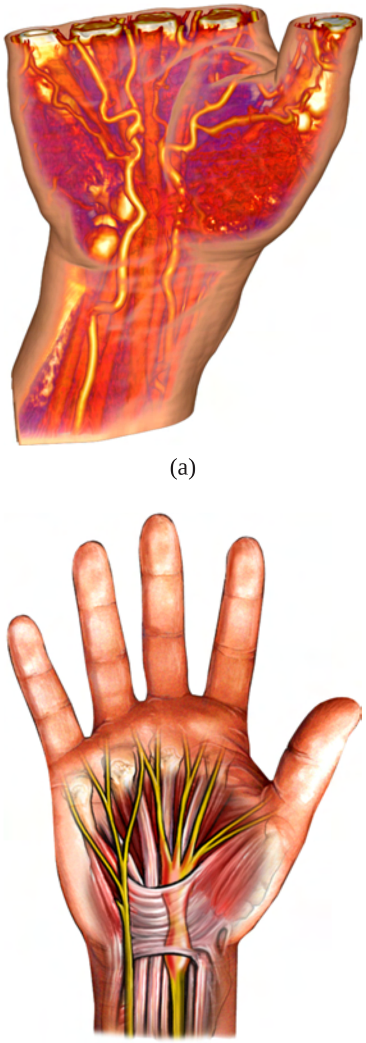

(a)

Image courtesy of Nucleus Medical Art
Copyright ©2004 Nucleus Medical Art, Inc. All rights reserved.
http://www.nucleusinc.com

(b)

Figure 5: *Comparing context-preserving volume rendering to illustration.* ***(a)*** *Context-preserving volume rendering of a hand data set.* ***(b)*** *Medical illustration using ghosting.*

4. Results

We experimented with the presented model using a wide variety of volumetric data sets. We have found that our approach makes transfer function specification much easier, as there is no need to pay special attention to opacity. Normally, tedious tuning is required to set the right opacity in order to provide good visibility for all structures of interest. Using the context-preserving volume rendering model, we just assign colors to the structures and use the parameters κ_t and κ_s to achieve an insightful rendering. Opacity in the transfer function is just used to suppress certain regions, such as background. This contrasts the usual direct volume rendering approach, where opacity specification is vital in order to achieve the desired visual result. In many cases, however, good results are difficult and laborious to achieve. For example, for structures sharing the same value range, as it is often the case with contrast-enhanced CT scans, it is impossible to assign different opacities using a one-dimensional transfer function. If one object is occluding the other, setting a high opacity will cause the occluded object to be completely hidden. Using high transparency, on the other hand, will make both objects hardly recognizable. Our method inherently solves this issue, as it bases opacity not only on data values, but also includes a location-dependent term. Thus, a key advantage of our approach is that it reduces the complexity of transfer function specification. In the following, we present some results achieved with our model in combination with a one-dimensional color transfer function. No segmentation was applied.

Figure 4 shows the tooth data set rendered with gradient-magnitude opacity-modulation, direct volume rendering using a clipping plane, and context-preserving volume rendering using the same transfer function. Gradient-magnitude opacity-modulation shows the whole data set but the over-

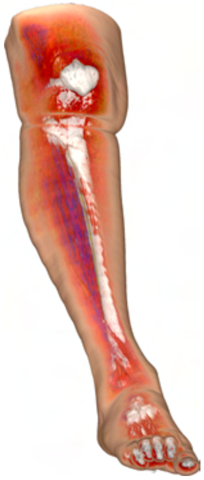


Figure 6: *Contrast-enhanced CT scan of a leg rendered using context-preserving volume rendering.*

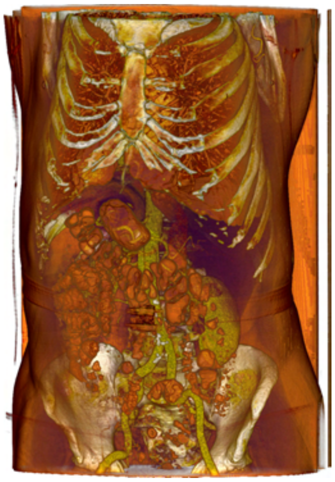


Figure 7: *Contrast-enhanced CT scan of a torso rendered using context-preserving volume rendering.*

lapping transparent structures make the interpretation of the image a difficult task. On the other hand, it is very difficult to place a clipping plane in a way that it does not cut away features of interest. Using context-preserving volume rendering, the clipping depth adapts to features characterized by varying lighting intensity or high gradient magnitude.

Figure 5 and Figure 6 show CT scans of a hand and a leg rendered using our model. These images have a strong resemblance to medical illustrations using the *ghosting* technique, as can be seen by comparing Figure 5 (a) and (b). By preserving certain characteristic features, such as creases on the skin, and gradually fading from opaque to transparent, the human mind is able to reconstruct the whole object from just a few hints while inspecting the detailed internal structures. This fact is commonly exploited by illustrators for static images. For interactive exploration the effect becomes even more pronounced and causes a very strong impression of depth.

Finally, Figure 7 shows a CT scan of a human torso. While the image shows many of the features contained in the data set, no depth ambiguities occur as the opacity is selectively varied.

As some of the strengths of our model are most visible in animated viewing, several supplementary video sequences are available for download at: `http://www.cg.tuwien.ac.at/research/vis/adapt/2004_cpvr`

5. Discussion

The context-preserving volume rendering model presents an alternative to conventional clipping techniques. It provides a simple interface for examining the interior of volumetric data sets. In particular, it is well-suited for medical data which commonly have a layered structure. Our method provides a mechanism to investigate structures of interest that are located inside a larger object with similar value ranges, as it is often the case with contrast-enhanced CT data. Landmark features of the data set are preserved. Our approach does not require any form of pre-processing, such as segmentation. The two parameters κ_t and κ_s allow intuitive control over the visualization: κ_t is used to interactively browse through the volume, similar to the variation of the depth of a clipping plane; κ_s normally remains fixed during this process and is only later adjusted to achieve a visually pleasing result.

While we have found that these parameters provide sufficient control, a possible extension is to make them data dependent, i.e., they are defined by specifying a transfer function. This increases the flexibility of the method, but also raises the burden on the user, as transfer function specification is a complex task. Thus, we propose a hybrid solution between both approaches. We keep the global constants κ_t and κ_s, but their values are modulated by a simple function of the scalar value. In Equation 5, κ_t is replaced by $\kappa_t \cdot \lambda_t(f_{P_i})$ and κ_s is replaced by $\kappa_s \cdot \lambda_s(f_{P_i})$. Both λ_t and λ_s are real-valued functions in the range $[0..1]$. For example, the user can specify zero for λ_t to make some regions impenetrable. Likewise, setting λ_s to zero for certain values ensures pure gradient-magnitude opacity-modulation. If one of these functions has a value of one, the corresponding global parameter remains unchanged. Figure 8 (a) shows the visible human male CT data set rendered using just the global parameters, while in Figure 8 (b) bone is made impenetrable by setting λ_t to zero for the corresponding values. Further degrees of freedom of our method are provided by its close connection to the illumination model. By changing the direction of a directional light source, for example, features can be

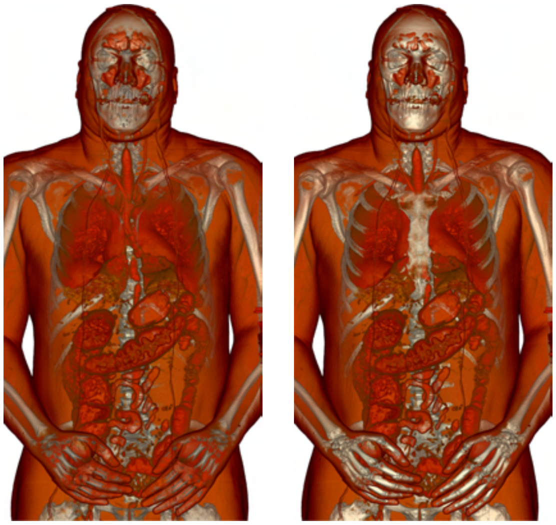


Figure 8: *Context-preserving volume rendering of the visible human male CT data set.* ***(a)*** *Only global parameter settings are used.* ***(b)*** *Bone is made impenetrable by using data-dependent parameters.*

interactively highlighted or suppressed, based on their orientation. Modifying the diffuse and specular factors will result in a variation of directional dependency, while adjusting the ambient component has a global influence. As means of changing these illumination properties are included in every volume visualization system, this additional flexibility will not increase the complexity of the user interface.

Our method does not pose any restrictions on the type of transfer function used. The model could be applied to modulate the opacity retrieved from a multi-dimensional transfer function without changes. Likewise, the modulation functions λ_t and λ_s could be multi-dimensional. For reasons of simplicity, we have only considered simple one-dimensional transfer functions in this paper.

6. Implementation

The implementation of context-preserving volume rendering is straight-forward, as it only requires a simple addition in the compositing routine of an existing volume rendering algorithm. The model only uses quantities which are commonly available in every volume renderer, such as gradient direction and magnitude and the depth along a viewing ray. We have integrated this method into a high-quality software volume ray casting system for large data [GBKG04a, GBKG04b]. It could also be used in an implementation using graphics hardware, such as GPU-based ray casting presented by Krüger and Westermann [KW03].

In our implementation, the most costly part of the context-preserving volume rendering model are the exponentiations. However, as with the specular term of the Phong illumination model, it is sufficient to approximate the exponentiation with a function that evokes a similar visual impression. Schlick [Sch94] proposed to use the following function:

$$x^n \approx \frac{x}{n - nx + x} \tag{6}$$

Thus, we can approximate the two exponentiations used by our model in the following way:

$$x^{(a \cdot y)^b} \approx \frac{x \cdot (b + a \cdot y - a \cdot b \cdot y)}{a \cdot y + b \cdot (a \cdot x \cdot y)} \tag{7}$$

This can be efficiently implemented using just 4 multiplications, 2 additions, 1 subtraction, and 1 division. In our implementation, this optimization reduced the cost from a 15% to a 5% increase compared to normal direct volume rendering.

7. Conclusions

The focus of our research was to develop an effective alternative to conventional clipping techniques for volume visualization. It preserves context information when inspecting the interior of an object. The context-preserving volume rendering model is inspired by artistic illustration. The opacity of a sample is modulated by a function of shading intensity, gradient magnitude, distance to the eye point, and previously accumulated opacity. The model is controlled by two parameters which provide a simple user-interface and have intuitive meaning. Context-preserving volume rendering is a unifying extension of both direct volume rendering and gradient-magnitude opacity-modulation and allows a smooth transition between these techniques. The approach simplifies transfer function specification, as the user only needs to specify constant opacity for structures of interest. Variation of the parameters of the model can then be used to interactively explore the volumetric data set. The achieved results have a strong resemblance to artistic illustrations and, thus, despite their increased information content, are very easy to interpret. Furthermore, as this approach adds little complexity to conventional direct volume rendering, it is well-suited for interactive viewing and exploration. For future work, it might be interesting to alter further properties (e.g., color or compositing mode) in addition to opacity using the presented concept.

We believe that research dealing with the inclusion of low-level and high-level concepts from artistic illustration is very beneficial for the field of volume visualization. The continuation of this research will include the further investigation of the presented approach as well as other illustrative volume rendering techniques in the context of an automated intent-based system for volume illustration [SF91].

8. Acknowledgements

The presented work has been funded by the ADAPT project (FFF-804544). ADAPT is supported by *Tiani Medgraph*, Austria (`http://www.tiani.com`), and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria. See `http://www.cg.tuwien.ac.at/research/vis/adapt` for further information on this project.

References

[CMH*01] CSÉBFALVI B., MROZ L., HAUSER H., KÖNIG A., GRÖLLER M. E.: Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum 20*, 3 (2001), 452–460.

[ER00] EBERT D. S., RHEINGANS P.: Volume illustration: non-photorealistic rendering of volume models. In *Proceedings of IEEE Visualization 2000* (2000), pp. 195–202.

[GBKG04a] GRIMM S., BRUCKNER S., KANITSAR A., GRÖLLER E.: Memory efficient acceleration structures and techniques for CPU-based volume raycasting of large data. In *Proceedings of the IEEE/SIGGRAPH Symposium on Volume Visualization and Graphics 2004* (2004), pp. 1–8.

[GBKG04b] GRIMM S., BRUCKNER S., KANITSAR A., GRÖLLER M. E.: A refined data addressing and processing scheme to accelerate volume raycasting. *Computers & Graphics 28*, 5 (2004), 719–729.

[HMBG01] HAUSER H., MROZ L., BISCHI G. I., GRÖLLER M. E.: Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics 7*, 3 (2001), 242–252.

[KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001* (2001), pp. 255–262.

[KVPL04] KONRAD-VERSE O., PREIM B., LITTMANN A.: Virtual resection with a deformable cutting plane. In *Proceedings of Simulation und Visualisierung 2004* (2004), pp. 203–214.

[KW03] KRÜGER J., WESTERMANN R.: Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE Visualization 2003* (2003), pp. 287–292.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MÖLLER T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003* (2003), pp. 513–520.

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Computer Graphics and Applications 8*, 3 (1988), 29–37.

[LM04] LUM E. B., MA K.-L.: Lighting transfer functions using gradient aligned sampling. In *Proceedings of IEEE Visualization 2004* (2004), pp. 289–296.

[LME*02] LU A., MORRIS C. J., EBERT D. S., RHEINGANS P., HANSEN C.: Non-photorealistic volume rendering using stippling techniques. In *Proceedings of IEEE Visualization 2002* (2002), pp. 211–218.

[RE01] RHEINGANS P., EBERT D. S.: Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics 7*, 3 (2001), 253–264.

[Sch94] SCHLICK C.: A fast alternative to Phong's specular model. In *Graphics gems IV*, Heckbert P., (Ed.). Academic Press, 1994, pp. 385–387.

[SF91] SELIGMANN D. D., FEINER S. K.: Automated generation of intent-based 3d illustrations. In *Proceedings of ACM Siggraph 1991* (1991), pp. 123–132.

[VKG04] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-driven volume rendering. In *Proceedings of IEEE Visualization 2004* (2004), pp. 139–145.

[WEE02] WEISKOPF D., ENGEL K., ERTL T.: Volume clipping via per-fragment operations in texture-based volume visualization. In *Procceedings of IEEE Visualization 2002* (2002), pp. 93–100.

[WEE03] WEISKOPF D., ENGEL K., ERTL T.: Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 298–312.

[WK95] WANG S. W., KAUFMAN A. E.: Volume sculpting. In *Proceedings of the Symposium on Interactive 3D Graphics 1995* (1995), pp. 151–156.

[ZDT04] ZHOU J., DÖRING A., TÖNNIES K. D.: Distance based enhancement for focal region based volume rendering. In *Proceedings of Bildverarbeitung für die Medizin 2004* (2004), pp. 199–203.

VolumeShop: An Interactive System for Direct Volume Illustration

Stefan Bruckner M. Eduard Gröller*

Institute of Computer Graphics and Algorithms
Vienna University of Technology, Austria

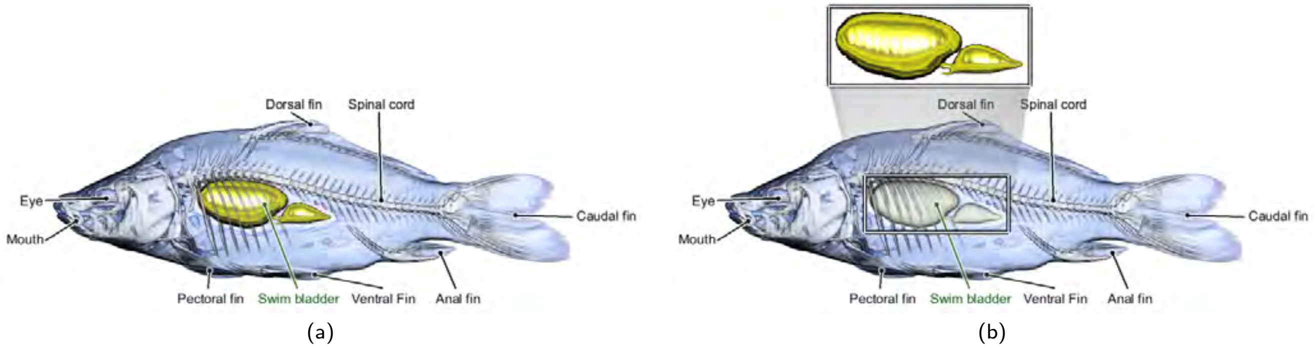


Figure 1: Annotated direct volume illustrations of a carp. (a) The swim bladder is highlighted using cutaways and ghosting. (b) The swim bladder is displayed enlarged.

Abstract

Illustrations play a major role in the education process. Whether used to teach a surgical or radiologic procedure, to illustrate normal or aberrant anatomy, or to explain the functioning of a technical device, illustration significantly impacts learning. Although many specimens are readily available as volumetric data sets, particularly in medicine, illustrations are commonly produced manually as static images in a time-consuming process. Our goal is to create a fully dynamic three-dimensional illustration environment which directly operates on volume data. Single images have the aesthetic appeal of traditional illustrations, but can be interactively altered and explored. In this paper we present methods to realize such a system which combines artistic visual styles and expressive visualization techniques. We introduce a novel concept for direct multi-object volume visualization which allows control of the appearance of inter-penetrating objects via two-dimensional transfer functions. Furthermore, a unifying approach to efficiently integrate many non-photorealistic rendering models is presented. We discuss several illustrative concepts which can be realized by combining cutaways, ghosting, and selective deformation. Finally, we also propose a simple interface to specify objects of interest through three-dimensional volumetric painting. All presented methods are integrated into VolumeShop, an interactive hardware-accelerated application for direct volume illustration.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.3 [Computer Graphics]: Picture/Image Generation—Viewing algorithms

Keywords: illustrative visualization, volume rendering, focus+context techniques

1 Introduction

A considerable amount of research has been devoted to developing, improving and examining direct volume rendering algorithms for visualization of scientific data. It has been shown that volume rendering can be successfully used to explore and analyze volumetric data sets in medicine, biology, engineering, and many other fields. In recent years, non-photorealistic or illustrative methods employed to enhance and emphasize specific features have gained popularity. Although we base our paper on this large body of research, our focus is somewhat different. Instead of using these techniques to improve the visualization of volume data for common applications such as diagnosis, we want to combine existing and new methods to directly generate illustrations, such as those found in medical textbooks, from volumetric data.

Illustrations are an essential tool in communicating complex relationships and procedures in science and technology. However, the time needed to complete an illustration is considerable and varies widely depending on the experience and speed of the illustrator and the complexity of the content. The more complicated the subject matter, the longer it will take the illustrator to research and solve a complex visual problem. Different illustration methods and styles can also have a significant impact on the time involved in the creation of an illustration. Therefore, illustrators are increasingly using computer technology to solve some of these problems. This, however, is mostly restricted to combining several manually created parts of an illustration using image processing software.

Volume rendering has gained some attention in the illustration community. For example, Corl et al. [4] describe the use of volume rendering to produce images as reference material for the manual generation of medical illustrations. We aim to take this development one step further. Our goal is to create a fully dynamic three-dimensional volume-based illustration environment where static images have the aesthetic appeal of traditional illustrations. The advantages of such a system are manifold: Firstly, the whole process of creating an illustration is accelerated. Different illustration methods and techniques can be explored interactively. It is easy to change the rendering style of a whole illustration - a process that

*e-mail: {bruckner | groeller}@cg.tuwien.ac.at

would otherwise require a complete redrawing. Moreover, the research process is greatly simplified. Provided that the object to be depicted is available as a volumetric data set, it can be displayed with high accuracy. Based on this data, the illustrator can select which features he wants to emphasize or present in a less detailed way. Illustration templates can be stored and reapplied to other data sets. This allows for the fast generation of customized illustrations which depict, for instance, a specific pathology. Finally, the illustration becomes more than a mere image. Interactive illustrations can be designed where a user can select different objects of interest and change the viewpoint.

This paper is subdivided as follows: In Section 2 we discuss related work. Section 3 gives a conceptual overview of our approach. In Sections 4, 5, and 6, we cover in detail the three fundamental building blocks of our direct volume illustration system, multi-object volume rendering, illustrative enhancement, and selective illustration, respectively. Section 7 discusses strategies and results for an efficient implementation of the presented concepts. The paper is concluded in Section 8.

2 Related Work

Non-photorealistic or illustrative rendering methods are a very active field of research. In volume visualization, Levoy [14] was the first to propose modulation of opacity using the magnitude of the local gradient. This is an effective way to enhance surfaces in volume rendering, as homogeneous regions are suppressed. Based on this idea, Rheingans and Ebert [19] present several illustrative techniques which enhance features and add depth and orientation cues. They also propose to locally apply these methods for regional enhancement. Using similar methods, Lu et al. [15] developed an interactive volume illustration system that simulates traditional stipple drawing. Csébfalvi et al. [5] visualize object contours based on the magnitude of local gradients as well as on the angle between viewing direction and gradient vector using depth-shaded maximum intensity projection. Lum and Ma [16] present a hardware-accelerated approach for high-quality non-photorealistic rendering of volume data. Exploring the variety of traditional illustration styles, selective emphasis of certain structures is an important technique. The concept of two-level volume rendering, proposed by Hauser et al. [8], allows focus+context visualization of volume data. Different rendering styles, such as direct volume rendering and maximum intensity projection, are used to emphasize objects of interest while still displaying the remaining data as context. Methods for combining multiple volume data sets have been investigated in the context of multi-modal data. For instance, Cai and Sakas [2] discuss different methods for data intermixing in volume rendering. Wilson et al. [25] propose a hardware-accelerated algorithm for multi-volume visualization. Leu and Chen [13] present a system for modeling scenes consisting of multiple volumetric objects which is restricted to non-intersecting volumes. The approach by Grimm et al. [7] uses alternating sampling for combining multiple volumes in dynamic scenes. An automated way of performing clipping operations has been presented by Viola et al. [23]. Inspired by cut-away views, which are commonly used in technical illustrations, they apply different compositing strategies to prevent an object from being occluded by a less important object. Konrad-Verse et al. [10] perform clipping using a mesh which can be flexibly deformed by the user with an adjustable sphere of influence. Zhou et al. [27] propose the use of distance to emphasize and de-emphasize different regions. Lum and Ma [17] use two-dimensional scalar-based lighting transfer functions to enhance material boundaries using illumination. Volume sculpting, proposed by Wang and Kaufman [24], enables interactive carving of volumetric data. Islam et al. [9] discuss methods for spatial and temporal splitting of volume data sets.

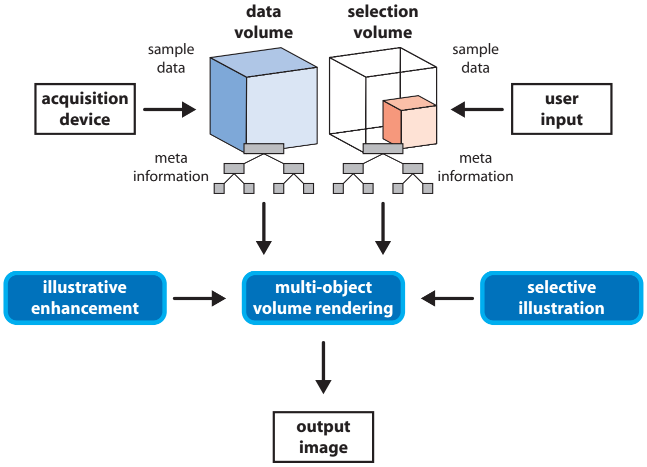


Figure 2: Conceptual overview of our direct volume illustration environment.

3 Overview

The architecture of VolumeShop, our direct volume illustration system, discriminates between two basic types of volumes: data volumes and selection volumes. A data volume stores the actual scalar field, for example acquired by a CT scanner. A selection volume specifies a particular structure of interest in a corresponding data volume. It stores real values in the range [0,1] where zero means ”not selected” and one means ”fully selected”. While both multiple data and selection volumes can be defined, only one pair is active at a time. Both volumes are stored in a bricked memory layout using reference counting, i.e., they are subdivided into small cubes which are accessed using an index data structure. Redundant information is not duplicated, thus, if two bricks contain the same data, they are stored in memory only once. The copy-on-write idiom is used for handling modifications. This is most useful for the selection volume due to its sparse nature. Furthermore, several pieces of meta information (e.g., min-max octrees, bounding boxes, and transformations) are stored for both volumes and updated on modification. This allows, for instance, the quick extraction of tight bounding volumes, which are used to skip empty space during rendering. At the heart of the system lies a multi-object volume rendering algorithm which is responsible for the concurrent visualization of multiple user-defined volumetric objects. It makes use of illustrative enhancement methods and selective illustration techniques defining the visual appearance of objects. A conceptual overview of our system is given is Figure 2. In the following sections, we will describe each of these components in detail.

4 Multi-Object Volume Rendering

When illustrating a volumetric data set, we want to enable interactive selection and emphasis of specific features. The user should be able to specify a region of interest which can be highlighted and transformed, similar to common image editing applications. We also want to permit arbitrary intersections between objects and control how the intersection regions are visualized.

Our approach identifies three different objects for the interaction with a volumetric data set: a *selection* is a user-defined focus region, the *ghost* corresponds to the selection at its original location, and the *background* is the remaining volumetric object. A transformation T can be applied to the selection, e.g., the user can move, rotate, or scale this object. While the concept of background and selection is used in nearly every graphical user interface, ghosts nor-

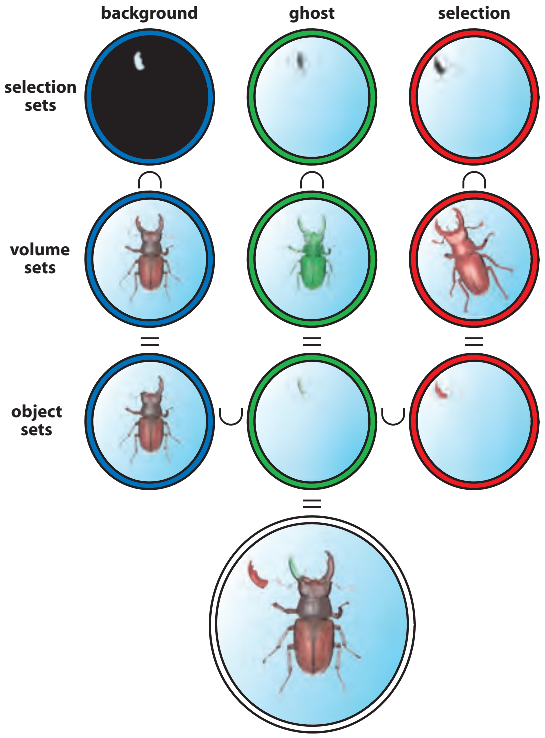


Figure 3: Overview of the basic multi-object combination process for background, ghost, and selection: the intersection between selection sets and volume sets results in object sets which are then combined.

mally exist, if at all, only implicitly. In the context of illustration, however, such an explicit definition of a ghost object is advantageous.

We assume a scalar-valued volumetric function f_V and a selection function f_S, which are defined for every point p in space. The selection function f_S has a value in $[0,1]$ which indicates the degree of selection. Based on this degree of selection we define three fuzzy *selection sets* S_S, S_G, and S_B (see Figure 3, first row) with their respective membership functions μ_S, μ_G, and μ_B:

$$\begin{aligned} \mu_{S_S}(p) &= f_S(T(p)) \\ \mu_{S_G}(p) &= f_S(p) \\ \mu_{S_B}(p) &= 1 - f_S(p) \end{aligned} \tag{1}$$

where T is the transformation that has been applied to the selection.

To control the appearance of our three objects, i.e., selection, ghost, and background, we define color and opacity transfer functions based on the values of f_V, which we denote c_S, α_S, c_G, α_G, and, c_B, α_B. We use the opacity transfer functions to define the membership functions of three *volume sets*, V_S, V_G, and V_B (see Figure 3, second row):

$$\begin{aligned} \mu_{V_S}(p) &= \alpha_S(f_V(T(p))) \\ \mu_{V_G}(p) &= \alpha_G(f_V(p)) \\ \mu_{V_B}(p) &= \alpha_B(f_V(p)) \end{aligned} \tag{2}$$

For each of our three objects we can now define an *object set* as the intersection between the corresponding selection and volume set (see Figure 3, third row):

$$\begin{aligned} S &= S_S \cap V_S \\ G &= S_G \cap V_G \\ B &= S_B \cap V_B \end{aligned} \tag{3}$$

These sets correspond to our basic objects selection, ghost, and background. Thus, in the following we will use these terms to refer to the respective object sets and vice versa. For volume rendering, we now need a way to determine the color and opacity at a point p in space depending on its grade of membership in these sets. We assume n sets $X_1, X_2, \ldots, X_n$ and their corresponding color transfer functions $c_1, c_2, \ldots, c_n$. We can then define the color at a point p as a weighted sum using the respective membership functions as weights:

$$c(p) = \frac{\sum_{i=1}^{n} c_i(p) \cdot \mu_i(p)}{\sum_{i=1}^{n} \mu_i(p)} \tag{4}$$

As the membership functions of our sets are based on the opacity and the degree of selection, we define the opacity at p as the grade of membership in the union of all sets:

$$\alpha(p) = \mu_{X_1 \cup X_1 \cup \ldots \cup X_n}(p) \tag{5}$$

Using Equations 4 and 5 for our three sets S, G, and B and the color transfer functions c_S, c_G, and c_B leads to a meaningful combination of colors and opacities when used in direct volume rendering. However, we want to provide the user with additional control over the appearance of regions of intersection. Frequently, for example, illustrators emphasize inter-penetrating objects when they are important for the intent of the illustration.

To achieve this we first need to identify potential regions of intersection. According to our definitions $B \cap G = \emptyset$, i.e., background and ghost never intersect. The selection, however, can intersect either the background, the ghost, or both. Thus, we direct our attention to the sets $GS = G \cap S$ and $BS = B \cap S$. For every point which is a member of one of these sets, we want to be able to specify its appearance using special intersection transfer functions for color and opacity. Thus, we define two new sets V_{GS} and V_{BS} with the following membership functions:

$$\begin{aligned} \mu_{V_{GS}}(p) &= \alpha_{GS}(f_V(p), f_V(T(p))) \\ \mu_{V_{BS}}(p) &= \alpha_{BS}(f_V(p), f_V(T(p))) \end{aligned} \tag{6}$$

The intersection transfer functions are two-dimensional. Their arguments correspond to the value of volumetric function f_V at point p and at $T(p)$, the value of the function at p transformed by the selection transformation T. Based on these two sets, we now define two alternative sets $\widehat{GS}$ and $\widehat{BS}$ for the regions of intersection:

$$\begin{aligned} \mu_{\widehat{GS}}(p) &= \begin{cases} 0 & \mu_{GS}(p) = 0 \\ \mu_{S_G \cap S_S \cap V_{GS}}(p) & otherwise \end{cases} \\ \mu_{\widehat{BS}}(p) &= \begin{cases} 0 & \mu_{BS}(p) = 0 \\ \mu_{S_B \cap S_S \cap V_{BS}}(p) & otherwise \end{cases} \end{aligned} \tag{7}$$

To evaluate the combined color and opacity at a point p in space, we use Equation 4 and 5 with the sets $S - (\widehat{GS} \cup \widehat{BS})$, $G - \widehat{GS}$, $B - \widehat{BS}$, $\widehat{GS}$, and $\widehat{BS}$ and the respective color transfer functions c_S, c_G, c_B, c_{GS}, and c_{BS}. We use the standard definitions for fuzzy set operators where the minimum operator is used for the intersection and the maximum operator is used for the union of two fuzzy sets [26].

The intersection transfer functions can be used to control the color and opacity in the region of intersection between two objects based on the scalar values of both objects. In our implementation we provide a default setting which is an opacity-weighted average between the one-dimensional color transfer functions of the two respective objects (background and selection, or ghost and selection). Further, we provide presets where the opacity is computed from the one-dimensional opacity transfer functions by one of the compositing operators derived by Porter and Duff [18]. The color can be

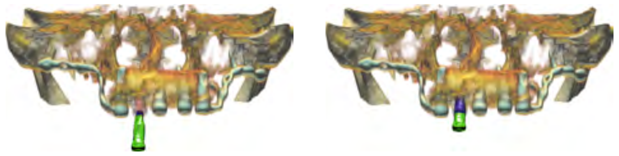


Figure 4: Using intersection transfer functions to illustrate implant placement in the maxilla. As the selection (green) is moved into the ghost (faint red), the intersection transfer function causes it to be displayed in blue.

specified arbitrarily. Additionally, the user can paint on the two-dimensional function using a gaussian brush to highlight specific scalar ranges. Figure 4 shows an example where the ghost/selection intersection transfer function is used to illustrate the placement of an implant in the maxilla. This kind of emphasis is not only useful for the final illustration, but can act as a kind of implicit visual collision detection during its design.

While we use the concept presented in this section for concurrent visualization of multiple objects derived from the same data set, this restriction is not necessary - objects could also be derived from multiple data sets. The approach could be straight-forwardly used for general multi-volume visualization. However, we note that the use of intersection transfer functions might not be feasible in a setup consisting of a large number of objects. Increasing the number of objects will quickly lead to a combinatorial explosion in the number of possible regions of intersection. In such a case the objects for which such a fine-grained control is required should be limited by application-specific constraints.

5 Illustrative Enhancement

Illustration is closely related to non-photorealistic rendering methods, many of which attempt to mimic artistic styles and techniques. In this section we present a simple approach which integrates several presented models and is thus well-suited for a volume illustration system. Most illumination models use information about the angle between normal, light vector and viewing vector to determine the lighting intensity. In volume rendering, the directional derivative of the volumetric function, the gradient, is commonly used to approximate the surface normal. Additionally, the gradient magnitude is used to characterize the ”surfaceness” of a point; high gradient magnitudes correspond to surface-like structures while low gradient magnitudes identify rather homogeneous regions. Numerous distinct approaches have been presented that use these quantities in different combinations to achieve a wide variety of effects. Our goal is to present a computationally inexpensive method which integrates many of these models.

We define a two-dimensional lighting transfer function. The arguments of this function are the dot product between the normalized gradient $\hat{N}$ and the normalized light vector $\hat{L}$ and the dot product between the normalized gradient and the normalized half-way vector $\hat{H}$, where $\hat{H}$ is the normalized sum of $\hat{L}$ and the normalized view vector $\hat{V}$. A two-dimensional lookup table stores the ambient, diffuse, and specular lighting contributions for every $\hat{N}\cdot\hat{L}$ and $\hat{N}\cdot\hat{H}$ pair. Additionally, a fourth component used for opacity enhancement is stored. Shading is then performed by using these four values in the following way to compute the shaded color c_s and shaded opacity α_s:

$$\begin{aligned} c_s &= (s_a(\hat{N}\cdot\hat{L},\hat{N}\cdot\hat{H}) + s_d(\hat{N}\cdot\hat{L},\hat{N}\cdot\hat{H}))\cdot c_u + s_s(\hat{N}\cdot\hat{L},\hat{N}\cdot\hat{H}) \\ \alpha_s &= (\min(1, s_\alpha(\hat{N}\cdot\hat{L},\hat{N}\cdot\hat{H}) + (1-|N|)))^{-1}\cdot\alpha_u \end{aligned} \tag{8}$$

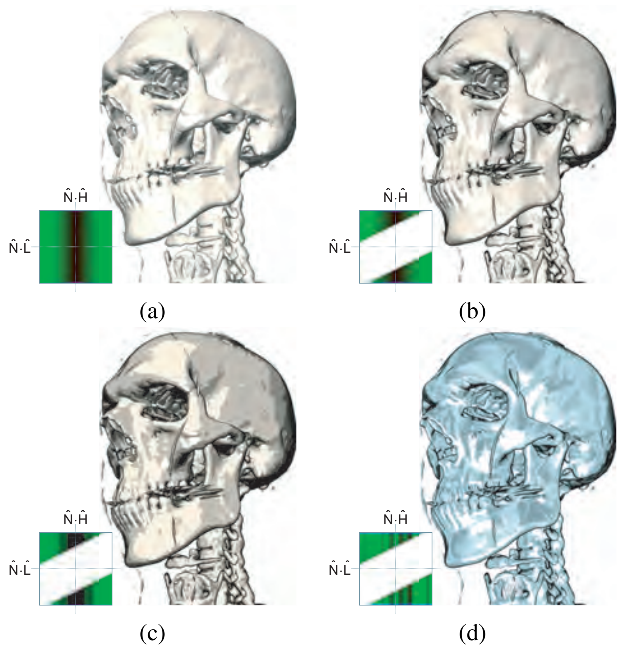


Figure 5: The same data set rendered with four different lighting transfer functions (the lighting transfer function for each image are displayed in the lower left corner - ambient, diffuse, specular, and opacity enhancement components are encoded in the red, green, blue, and alpha channel, respectively). (a) Standard Phong-Blinn lighting. (b) Phong-Blinn lighting with contour enhancement. (c) Cartoon shading with contour enhancement. (d) Metal shading with contour enhancement.

where c_u and α_u are the unshaded color and opacity, and s_a, s_d, and s_s are the shading transfer function components for ambient, diffuse, and specular lighting contributions. The opacity enhancement component of the transfer function denoted by s_α is combined with the gradient magnitude $|N|$ to modulate the unshaded opacity value (we assume that the gradients have been scaled such that $|N|$ is between zero and one).

We use the terms ”ambient”, ”diffuse”, and ”specular” to illustrate the simple correspondence in case of Phong-Blinn lighting. However, the semantics of these components are defined by the model used for generation of the lighting transfer function. Thus, a lighting transfer function might use these terms to achieve effects completely unrelated to ambient, diffuse, and specular lighting contributions. In a similar matter, when examining Equation 8 it can be seen that the ambient and diffuse components could be combined without loss. We only choose to keep them separate for the sake of consistency and simplicity.

It is straight-forward to use this kind of lighting transfer function for common Phong-Blinn lighting. However, many other models can also be specified in this way and evaluated at constant costs. For example, contour lines are commonly realized by using a dark color where the dot product between gradient and view vector $\hat{N}\cdot\hat{V}$ approaches zero, i.e., these two vectors are nearly orthogonal. If we have $\hat{N}\cdot\hat{L}$ and $\hat{N}\cdot\hat{H}$ with $\hat{H} = \widehat{L+V}$, then $\hat{N}\cdot\hat{V} = 2(\hat{N}\cdot\hat{H}) - \hat{N}\cdot\hat{L}$. We can thus create a lighting transfer function where we set ambient, diffuse and specular components to zero where $\hat{N}\cdot\hat{L} \approx 2(\hat{N}\cdot\hat{H})$. One advantage of this approach is that artifacts normally introduced by using a threshold to identify contour lines can be remedied by smoothing them in the lighting transfer function (e.g., using a gaussian) with no additional costs during rendering. Using the opacity enhancement component of the lighting transfer function also allows for a meaningful combination of contour enhancement and transparency: the opacity of contour regions is increased, but only where the gradient magnitude is high. Without taking the gradient magnitude into account opacity enhanced contour lines would lead to a cluttered image in translucent views. This is due to rapidly varying gradient directions in nearly homogeneous regions. Pure gradient-magnitude opacity-enhancement without directional dependence just requires a constant s_α. Other methods, such as cartoon shading [3] or metal shading [6] can be realized straight-forwardly and combined with effects like contour enhancement. Figure 5 shows an image rendered using four different lighting transfer functions.

6 Selective Illustration

In this section we present techniques for selective illustration. Selective illustration techniques are methods which aim to emphasize specific user-defined features in a data set using visual conventions commonly employed by human illustrators. They are closely related to focus+context approaches frequently found in information visualization. The general idea is to highlight the region of interest (focus) without completely removing other information important for orientation (context).

6.1 Volume Painting

Volume Segmentation, i.e., the identification of individual objects in volumetric data sets is an area of extensive research, especially in medical imaging applications. Approaches range from very general methods to algorithms specifically designed to identify certain structures. An important criterion is the exactness of the segmentation, i.e., the ratio between correctly and incorrectly classified voxels. In practice, due to limited information, this criterion is difficult to measure. For volume illustration, however, voxel-exact classification of features is not necessarily of primary concern. Rather, it is important that the illustrator can quickly and easily add and remove structures of interest to and from the selection. Furthermore, as our approach is based on a fuzzy selection function, this fuzzyness should be also supported by the selection definition method. For this reason, we use a simple three-dimensional volumetric painting approach for selection definition. When the user clicks on the image, a ray is cast from the corresponding position on the image plane into the data volume. At the first non-transparent voxel that is intersected by the ray, a volumetric brush (e.g., a three-dimensional gaussian) is ”drawn” into the selection volume for each non-transparent voxel within the bounding box of the brush. Different composition methods can be chosen, for example addition (i.e., actual painting) or subtraction (i.e., erasing). We have found that this approach is intuitive and capable of achieving good results in a short time: the user specifies a transfer function which displays the object of interest and then just paints on it until it is fully selected. However, it is clear that a real-world application should also include more sophisticated algorithms. Just like image editing software normally supports manual and semi-automatic selection mechanisms (e.g., the common ”magic wand tool”), a volume illustration system should include volume painting as well as region growing or watershed segmentation.

6.2 Cutaways and Ghosting

Cutaways (also referred to as cut-away views) are an important tool commonly employed by illustrators to display specific features occluded by other objects. The occluding object is cut out to reveal the structure of interest. Viola et. al. [22] introduced importance-driven volume rendering, a general framework for determining which object is to be cut by using an importance function. Our simplified three-object setup allows static definition of this importance function, which enables us to skip costly importance compositing and thus allows for an efficient implementation. Cutaways are only performed on the background and can be independently defined for ghost and selection.

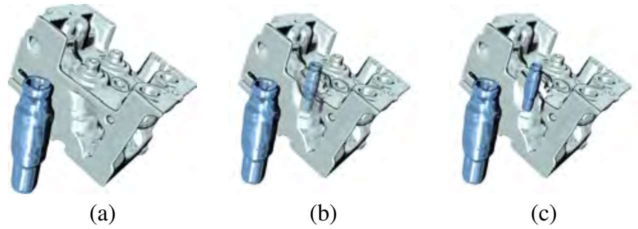


Figure 6: Different degrees of ghosting - from no ghosting (a) to full cutaway (c).

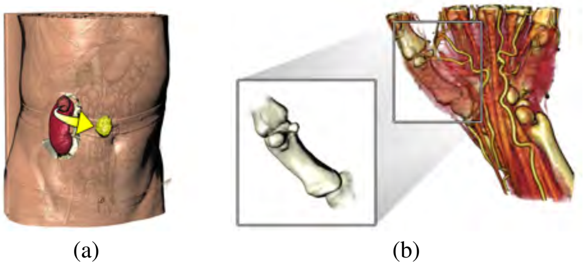


Figure 7: Using different artistic visual conventions. (a) Illustrating a tumor resection procedure using an automatically generated arrow. (b) Detailed depiction of a hand bone using a fan.

Ghosting refers to a technique which is frequently used in conjunction with cutaways. Instead of removing the occluding regions completely, opacity is selectively reduced in a way which attempts to preserve features such as edges. This tends to aid mental reconstruction of these structures and generally gives a better impression of the spacial location of the object in focus. In our approach, the user can smoothly control the degree of ghosting from no ghosting (opacity is not reduced at all) to full cutaway view (occluding structures are completely suppressed) as shown in Figure 6. This is achieved by combining a user-defined ghosting factor with the opacity-enhancement component of the lighting transfer function. Thus, for a lighting transfer function which enhances the opacity of contours, increasing the degree of ghosting will preserve these regions. Again, in the context of importance-driven volume rendering this approach can be seen as a special level-of-sparseness which is designed to closely correspond to traditional illustration techniques.

6.3 Visual Conventions and Interaction

As the selection can undergo a user-defined transformation there are a number of possibilities for combining the effects of transfer functions, cutaways and ghosting, and spacial displacement. In its simplest form, this can be used to illustrate the removal or insertion of an object. Furthermore, ”magic views” on a structure of interest can be generated, where the object is displayed using a different degree of detail, orientation, or rendering style.

Illustrators commonly employ certain visual conventions to indicate the role of an object in their works. In our illustration environment, we provide the user with different kinds of visual enhancements inspired by these conventions:

Boxes: For three-dimensional interaction, bounding boxes provide useful cues on the position and orientation of an object if occlusions are handled correctly. The display of bounding boxes is most useful when the selection is arranged during the design of an illustration. For the presentation of the illustration, however, the bounding boxes can be distracting and potentially occlude important details.

Arrows: Arrows normally suggest that an object actually has been moved during the illustrated process (e.g., in the context of a surgical procedure) or that an object needs to be inserted at a certain location (e.g., in assembly instructions). Analogously, we use arrows to depict the translation between ghost and selection, i.e., the arrow is automatically drawn from the object's original position to its current location. To avoid very short arrows in case the selection and the ghost project to nearby positions in image space, we use the screen-space depth difference to control the curvature of the arrow. This leads to the kind of bent arrows frequently found in illustrations. Figure 7 (a) shows an example for the use of arrows.

Fans: A fan is a connected pair of shapes, such as rectangles or circles, used to indicate a more detailed or alternative depiction of a structure. It can be easily constructed by connecting the screen-space bounding rectangles of ghost and selection. In combination with cutaways and ghosting, this type of enhancement can lead to very expressive visualizations, depicting, for example, two different representations of the same object (see Figure 7 (b)).

Apart from controlling visual appearance, it is useful to provide different interaction types based on the role of an object in the illustration. A selection can be in one of three states which influence the way it behaves in relation to the remaining scene:

Integrated: The selection acts as fully belonging to the three-dimensional scene. This is intuitive, but has certain drawbacks. For example, when the viewpoint is rotated, the selection's movement is dependent on its distance to the origin. It can easily move out of the viewport or can be occluded by other objects.

Decoupled: The opposite to the integrated approach is to fully decouple the selection from the scene. It can be independently manipulated and is not affected by the viewing transformation. This is, for instance, useful when it is required to depict an object at a specific orientation regardless of the viewing transformation.

Pinned: A useful hybrid between the two modes above is to allow the object to be pinned to its current position in image space. Its on-screen location remains static, but it is still affected by rotations. A rotation of the viewpoint causes the same relative rotation of the object. For example, this can be used to generate a special view which always shows the part of an object facing away from the viewer in the background object.

6.4 Annotations

Hand-made illustrations in scientific and technical textbooks commonly use labels or legends to establish a co-referential relation between pictorial elements and textual expressions. As we allow multiple selections to be defined, annotations are important for both recreating the appearance of static illustrations and simplifying orientation in our interactive environment. For placing annotations we need their screen-space bounding rectangles and anchor points. We use the following guidelines to derive a simple layout algorithm for optically pleasing annotation placement (for a more complete description of annotation layout styles and guidelines refer to [1]):

- Annotations must not overlap.

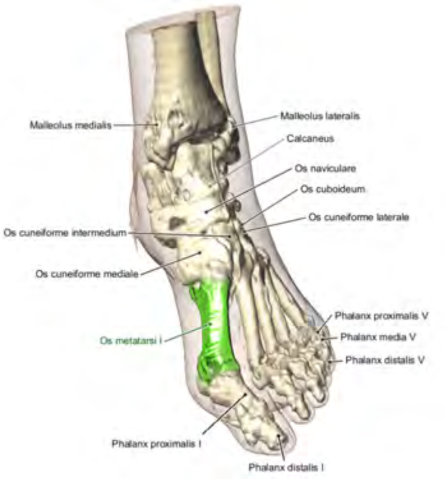


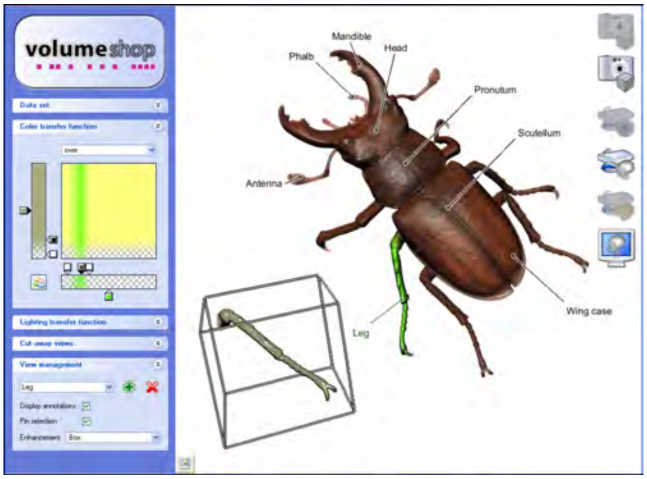
Figure 8: Annotated illustration of a human foot - the current selection is highlighted.

- Connecting lines between annotation and anchor point must not cross.
- Annotations should not occlude any other structures.
- An annotation should be placed as close as possible to its anchor point.

In many textbook illustrations, annotations are placed along the silhouette of an object to prevent occlusions. We can approximate this by extracting the convex hull of the projections of the bounding volumes of all visible objects. The resulting polygon is radially parameterized. Thus, the position of an annotation is defined by one value in the range $[0, 1]$. Based on its location in parametric space, a label is always placed in such a way that it remains outside the convex hull. All annotations are initially placed at the position along the silhouette polygon which is closest to their respective anchor point. We then use a simple iterative algorithm which consists of the following steps:

1. If the connection lines of any two labels intersect, exchange their positions.
2. If exchanging the positions of two labels brings both closer to their anchor points, exchange their positions.
3. If a label overlaps its predecessor, it is moved by a small delta.

These three steps are executed until either all intersections and overlaps are resolved or the maximum number of iterations has been reached. Remaining intersections and overlaps are handled by disabling annotations based on priority. We use the screen-space depth of the anchor point to define these priorities, i.e., annotations whose reference structures are farther away will be disabled first. While this basic algorithm does not result in an optimal placement, it is very fast for a practical number of labels (usually no more than 30 annotations are used in a single illustration) and generally leads to a visually pleasing layout. Due to the initialization of annotation locations at the position on the silhouette closest to the anchor point, the annotations generally move smoothly in animated views. Discontinuities only occur when intersections and overlaps need to be resolved. As some annotated structures might not be visible from every viewpoint, we use the screen-space depth of the anchor point to control the opacity of the connection line between anchor point and label. Figure 8 shows an annotated illustration of a human foot highlighting the current selection.



Leg

Display annotations ☑

Pin selection ☑

Enhancement Box

Figure 9: Screenshot of VolumeShop (`http://www.cg.tuwien.ac.at/volumeshop`) during operation.

7 Implementation

In this section, we briefly describe the implementation of our algorithm for illustrative multi-object volume rendering with support for cutaways and ghosting. It is integrated into VolumeShop, a prototype application for interactive direct volume illustration (see Figure 9). VolumeShop has been implemented in C++ and Cg using OpenGL. While we have clear indications that the current version of NVidia's Cg compiler does not produce optimal code in all circumstances, we have refrained from hand-optimizing assembly language shaders for the sake of portability.

It is possible to implement all presented methods in one single rendering pass. However, this would introduce considerable computational overhead, as, for example, multi-object compositing would have to be performed for every sample point even if it only intersects one object. While current graphics hardware supports dynamic branching, it still introduces severe performance penalties. It is therefore favorable to choose a multi-pass approach. A well-established strategy is to use the early-z culling capability of modern hardware for computational masking. Employing this approach we can identify those regions where less work has to be performed and use simplified vertex and fragment programs in these areas.

We can quickly extract bounding volumes for background, ghost, and selection by traversing our hierarchical data structures and rendering the corresponding geometry. Initially, we set up two depth maps by rendering the bounding volumes of ghost and selection each into a separate depth texture with the depth test set to LESS. These depth maps are used in the subsequent rendering passes to discard fragments, thus emulating a two-sided depth test. For smooth cutaways we additionally filter these depth maps using a large kernel.

In principle, our implementation is comprised of three volume rendering passes using three sets of vertex and fragment programs with increasing complexity:

Background pass: The first volume rendering pass is responsible for the background object. We set the depth test to LESS and render the bounds of the background object into the depth buffer. Depth buffer writes are then disabled to take advantage of early-z culling and the depth test is set to GREATER. Thus, empty space up to the first intersection point of a viewing ray with the background bounding volume is skipped without executing the fragment program. We then render view-aligned slices in back-to-front order and perform shading in a fragment program. Shadow mapping hardware is used to discard fragments whose depth is greater or equal than the corresponding value of the ghost or selection depth texture. Thus, regions which might contain the ghost and/or the selection are effectively cut out from the background object.

Ghost pass: In the second volume rendering pass we start by clearing the depth buffer and rendering the bounding volume of the ghost object with the depth test set to GREATER. Then depth buffer writes are disabled again and the depth test is set to LESS. The fragment program needs to perform shading for background and ghost. Fragments whose depth value is greater or equal than the corresponding value of the selection depth map are discarded. If cutaways are enabled then the opacity of the background is additionally modulated by a user-defined ghosting factor for fragments whose depth value is greater or equal than the corresponding value of the ghost depth map.

Selection pass: For the final pass we render the selection bounds into the cleared depth buffer with the depth test set to GREATER. Depth buffer writes are then disabled again and the depth test is set to LESS. The selection transformation is handled by passing in two sets of texture coordinates: one unmodified set for background and ghost, and an additional set for the selection which is transformed accordingly. In the fragment program we need to perform shading for background, selection, and ghost. We also handle background/selection and ghost/selection intersections by using the colors and opacities defined in the intersection transfer functions. For cutaways, the background's opacity is additionally modulated for fragments whose depth value is greater or equal than the corresponding values in one or both of the depth maps.

For handling of intersections with opaque geometry an additional depth map is generated before the background pass. The color contributions of the geometry are blended into the frame buffer. The depth texture is used in all three rendering passes to discard fragments which are occluded by geometry. Visual enhancements are either displayed as real three-dimensional objects with correct intersection handling (e.g., bounding boxes) or as overlays (e.g., fans). As larger selections will require more fragments to be processed in the more complex rendering passes, the performance of the presented algorithm mainly depends on the size of the selection. Thus, if no selection has been defined we achieve almost the same frame rates as conventional slice-based volume rendering due to the effectiveness of early-z culling. Selections, by definition, typically will be rather small compared to the background. Additionally, if we can determine that the selection does not intersect background or ghost (e.g., by means of a simple bounding box test) we execute a simplified fragment program in the selection pass.

For obtaining performance results we used the following setup: The chosen data set was the standard UNC CT head (256^3) rendered using $\sqrt{3 \cdot 256^2} \approx 444$ slices - a realistic number for high-quality rendering. The selection was set to a cube sized 16^3, 32^3, and 64^3 voxels centered in the middle of the data set. The selection transformation was set to identity. The transfer functions for background, ghost, and selection were set to zero opacity for values up to 1228 and to an opacity of one for all values above. The frame rates given in Table 1 are average figures for three 360° rotations about the x-,y-, and z-axis for a 512^2 viewport. An Intel Pentium 4 3.4 GHz CPU and an NVidia GeForce 6800 GT GPU were used to obtain these measurements.

These results indicate that our approach is well-suited for high-quality rendering in interactive applications. In the future, we expect to further increase the rendering performance by integrating early ray termination as proposed by Krüger and Westermann [11].

selection	frame rate
none	8.28
16^3	8.04
32^3	6.81
64^3	4.86

Table 1: Performance results for rendering 444 slices of the UNC CT head (256^3) using different selection sizes.

8 Conclusion and Future Work

In this work, we introduced the general concept of a direct volume illustration environment. Based on this concept, VolumeShop (`http://www.cg.tuwien.ac.at/volumeshop`), an interactive system for the generation of high-quality illustrations from volumetric data, has been developed. An intuitive three-object setup for the interaction with volumetric data was discussed. We contributed a general technique for multi-object volume rendering which allows for emphasis of intersection regions via two-dimensional transfer functions. Furthermore, we introduced a unified approach to efficiently integrate different non-photorealistic illumination models. Techniques for selective illustration were presented which combine cutaways and ghosting effects with artistic visual conventions for expressive visualization of volume data. In addition, we proposed volume painting as an interactive selection method and presented an algorithm for automated annotation placement. A hardware-accelerated volume renderer was developed which combines the presented techniques for interactive volume illustration.

While we believe that the results achieved with our prototype system are promising, a lot of work remains to be done. In the future we aim to integrate further artistic styles and techniques for the creation of aesthetically pleasing illustrations [20]. We also want to investigate methods for automatically guiding viewpoint specification [21] and light placement [12]. Finally, improved interaction metaphors and techniques could significantly contribute to the usability of a volume illustration system.

Acknowledgements

The work presented in this publication is carried out as part of the ex**vis**ation project (`http://www.cg.tuwien.ac.at/research/vis/exvisation`) supported by the Austrian Science Fund (FWF) grant no. P18322. We would like to thank Sören Grimm and Ivan Viola for several fruitful discussions. Furthermore, we thank the anonymous reviewers for their valuable comments.

The carp data set is courtesy of Michael Scheuring, University of Erlangen, Germany. The stag beetle data set has been provided by Georg Glaeser, Vienna University of Applied Arts, Austria and Johannes Kastner, Wels College of Engineering, Austria. The visible human data set is courtesy of the Visible Human Project, National Library of Medicine, USA. The engine block data set is courtesy of General Electric, USA.

References

[1] K. Ali, K. Hartmann, and T. Strothotte. Label layout for interactive 3D illustrations. *Journal of the WSCG*, 13(1):1–8, 2005.

[2] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Computer Graphics Forum*, 18(3):359–368, 1999.

[3] J. Claes, F. Di Fiore, G. Vansichem, and F. Van Reeth. Fast 3D cartoon rendering with improved quality by exploiting graphics hardware. In *Proceedings of Image and Vision Computing New Zealand 2001*, pages 13–18, 2001.

[4] F. M. Corl, M.R. Garland, and E. K. Fishman. Role of computer technology in medical illustration. *American Journal of Roentgenology*, 175(6):1519–1524, 2000.

[5] B. Csébfalvi, L. Mroz, H. Hauser, A. König, and M. E. Gröller. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum*, 20(3):452–460, 2001.

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

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

[8] 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.

[9] S. Islam, S. Dipankar, D. Silver, and M. Chen. Spatial and temporal splitting of scalar fields in volume graphics. In *Proceedings of the IEEE Symposium on Volume Visualization and Graphics 2004*, pages 87– 94, 2004.

[10] O. Konrad-Verse, B. Preim, and A. Littmann. Virtual resection with a deformable cutting plane. In *Proceedings of Simulation und Visualisierung 2004*, pages 203–214, 2004.

[11] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE Visualization 2003*, pages 287–292, 2003.

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

[13] A. Leu and M. Chen. Modelling and rendering graphics scenes composed of multiple volumetric datasets. *Computer Graphics Forum*, 18(2):159–171, 1999.

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

[15] A. Lu, C. J. Morris, D. S. Ebert, P. Rheingans, and C. Hansen. Non-photorealistic volume rendering using stippling techniques. In *Proceedings of IEEE Visualization 2002*, pages 211–218, 2002.

[16] E. B. Lum and K.-L. Ma. Hardware-accelerated parallel non-photorealistic volume rendering. In *Proceedings of the International Symposium on Non-photorealistic Animation and Rendering 2002*, pages 67–74, 2002.

[17] E. B. Lum and K.-L. Ma. Lighting transfer functions using gradient aligned sampling. In *Proceedings of IEEE Visualization 2004*, pages 289–296, 2004.

[18] T. Porter and T. Duff. Compositing digital images. *Computer Graphics*, 18(3):253–259, 1984.

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

[20] P.-P. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: A model for capturing NPR shading from art. In *Proceedings of Graphics Interface 2001*, pages 143–150, 2001.

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

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

[23] 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.

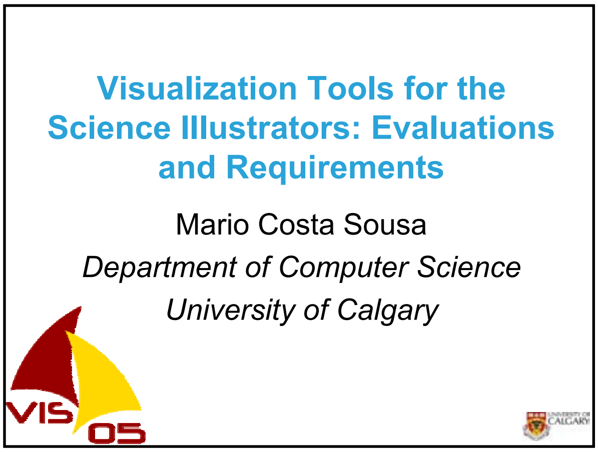
[24] S. W. Wang and A. E. Kaufman. Volume sculpting. In *Proceedings of the Symposium on Interactive 3D Graphics 1995*, pages 151–156, 1995.

[25] B. Wilson, E. B. Lum, and K.-L. Ma. Interactive multi-volume visualization. In *Proceedings of the International Conference on Computational Science 2002*, pages 102–110, 2002.

[26] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

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

Visualization Tools for the Science Illustrators: Evaluations and Requirements

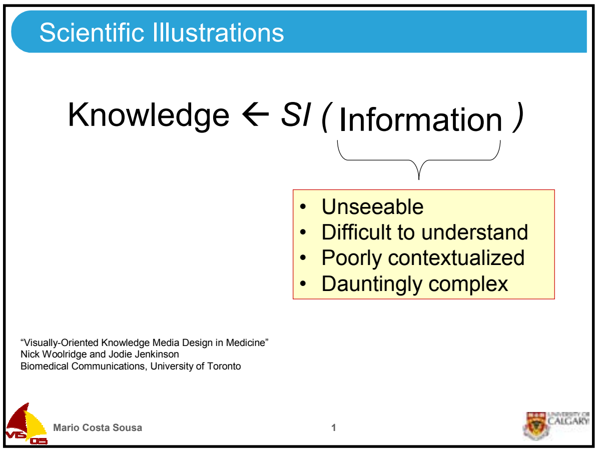


Visualization Tools for the Science Illustrators: Evaluations and Requirements

Mario Costa Sousa

Department of Computer Science

University of Calgary

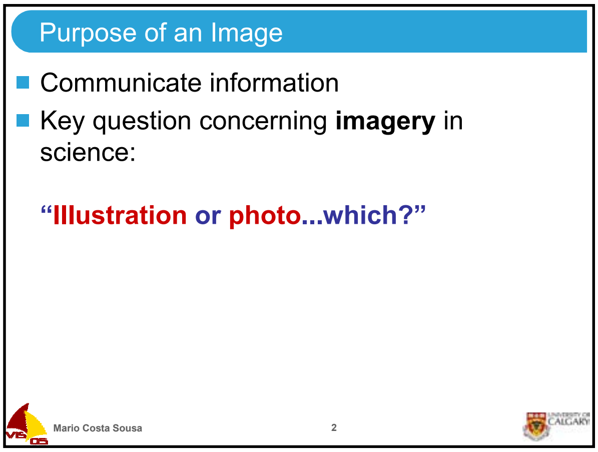


Scientific Illustrations

Knowledge ← *SI* (Information)

- Unseeable
- Difficult to understand
- Poorly contextualized
- Dauntingly complex

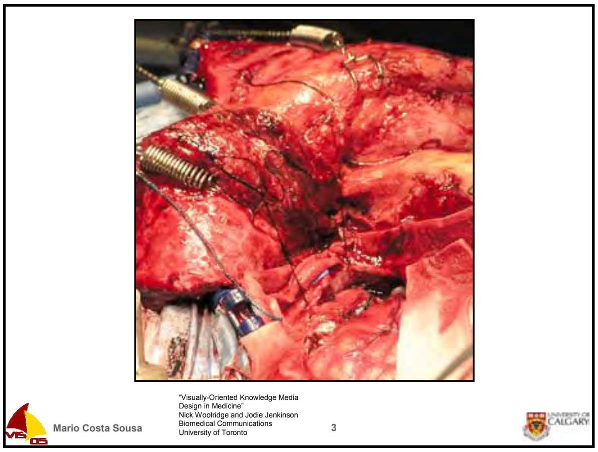
"Visually-Oriented Knowledge Media Design in Medicine"
Nick Woolridge and Jodie Jenkinson
Biomedical Communications, University of Toronto



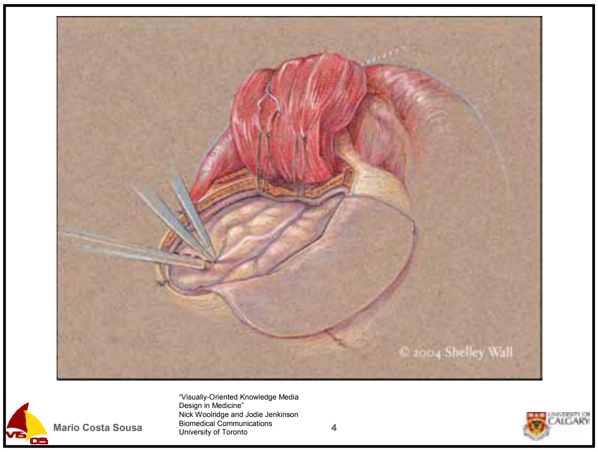
Purpose of an Image

- Communicate information
- Key question concerning **imagery** in science:

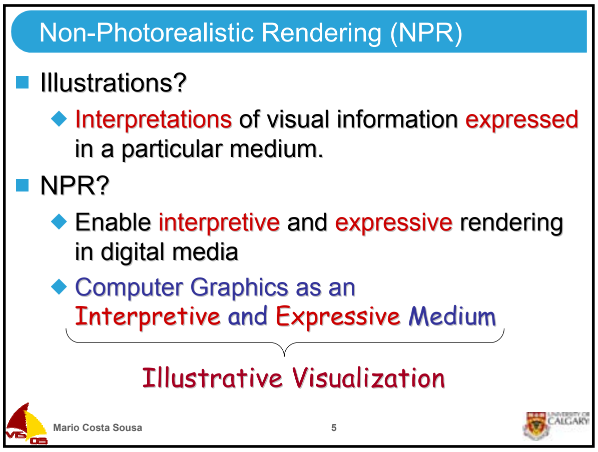
  "Illustration or photo...which?"



"Visually-Oriented Knowledge Media Design in Medicine"
Nick Woolridge and Jodie Jenkinson
Biomedical Communications
University of Toronto



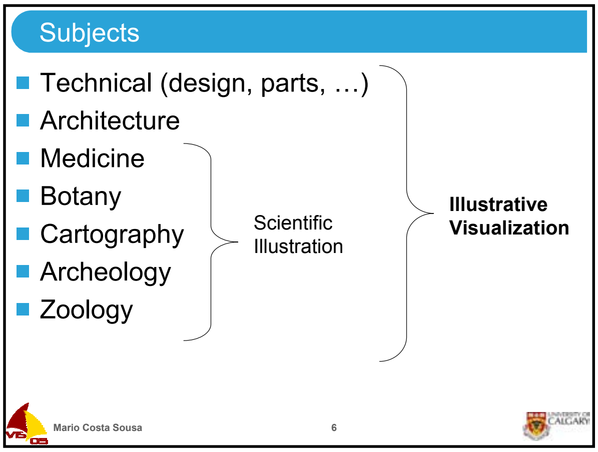
"Visually-Oriented Knowledge Media Design in Medicine"
Nick Woolridge and Jodie Jenkinson
Biomedical Communications
University of Toronto



Non-Photorealistic Rendering (NPR)

- Illustrations?
  - Interpretations of visual information expressed in a particular medium.
- NPR?
  - Enable interpretive and expressive rendering in digital media
  - Computer Graphics as an Interpretive and Expressive Medium

Illustrative Visualization

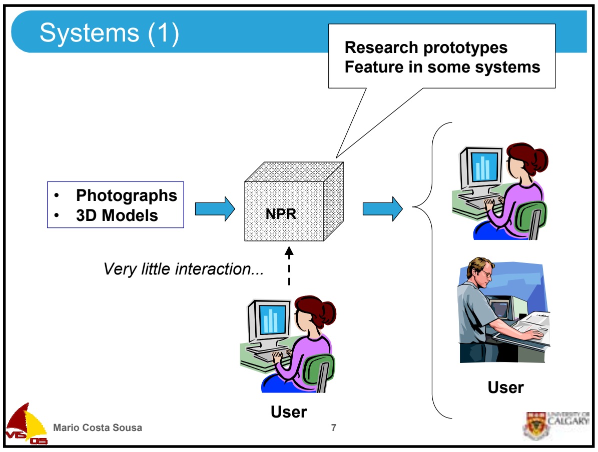


Subjects

- Technical (design, parts, …)
- Architecture
- Medicine
- Botany
- Cartography
- Archeology
- Zoology

Scientific Illustration

Illustrative Visualization



Systems (1)

Research prototypes
Feature in some systems

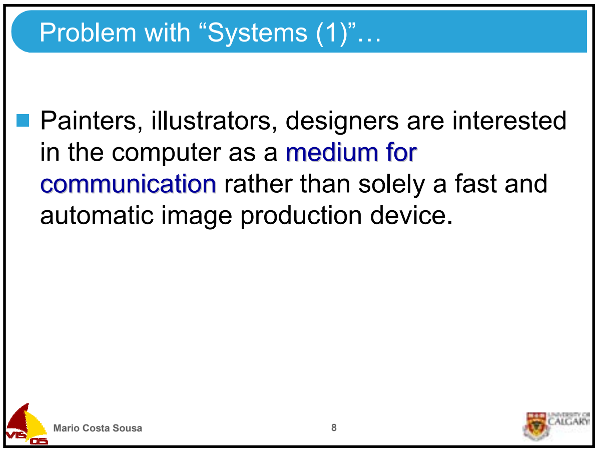
- **Photographs**
- **3D Models**

NPR

Very little interaction...

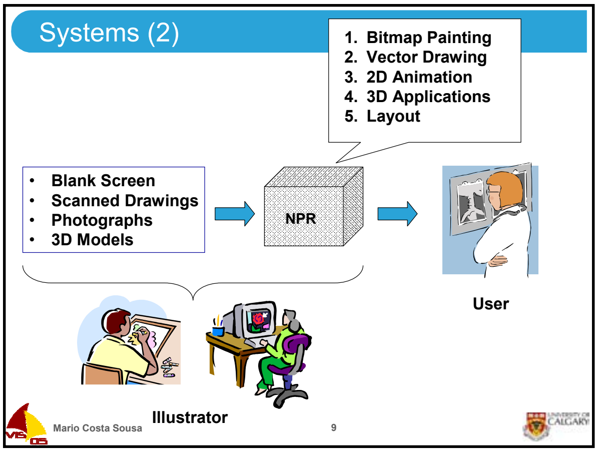
User

User



Problem with "Systems (1)"…

- Painters, illustrators, designers are interested in the computer as a medium for communication rather than solely a fast and automatic image production device.



Systems (2)

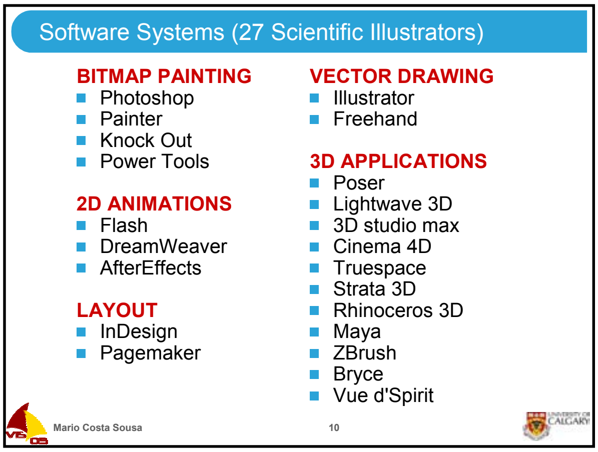
1. **Bitmap Painting**
2. **Vector Drawing**
3. **2D Animation**
4. **3D Applications**
5. **Layout**

- **Blank Screen**
- **Scanned Drawings**
- **Photographs**
- **3D Models**

NPR

User

Illustrator



Software Systems (27 Scientific Illustrators)

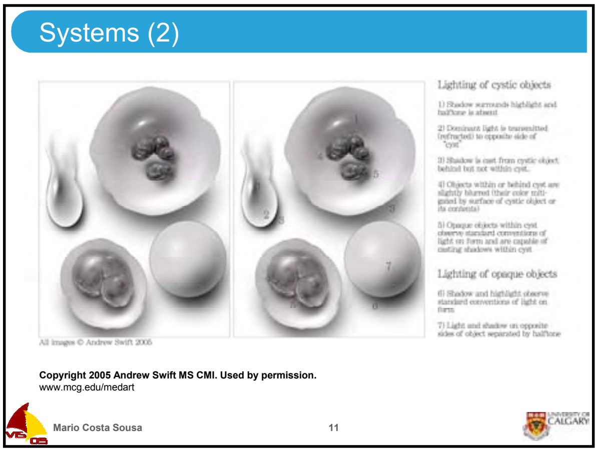
BITMAP PAINTING
- Photoshop
- Painter
- Knock Out
- Power Tools

2D ANIMATIONS
- Flash
- DreamWeaver
- AfterEffects

LAYOUT
- InDesign
- Pagemaker

VECTOR DRAWING
- Illustrator
- Freehand

3D APPLICATIONS
- Poser
- Lightwave 3D
- 3D studio max
- Cinema 4D
- Truespace
- Strata 3D
- Rhinoceros 3D
- Maya
- ZBrush
- Bryce
- Vue d'Spirit



Systems (2)

Lighting of cystic objects

1) Shadow surrounds highlight and halftone is absent

2) Dominant light is transmitted (refracted) to opposite side of "cyst"

3) Shadow is cast from cystic object behind but not within cyst.

4) Objects within or behind cyst are slightly blurred (their color mitigated by surface of cystic object or its contents)

5) Opaque objects within cyst observe standard conventions of light on form and are capable of casting shadows within cyst

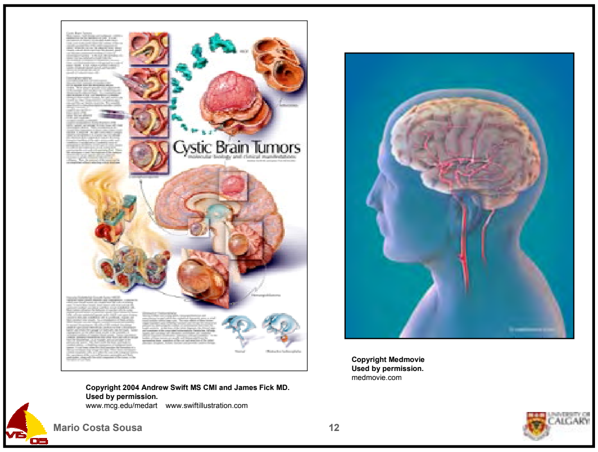
Lighting of opaque objects

6) Shadow and highlight observe standard conventions of light on form

7) Light and shadow on opposite sides of object separated by halftone

All images © Andrew Swift 2005

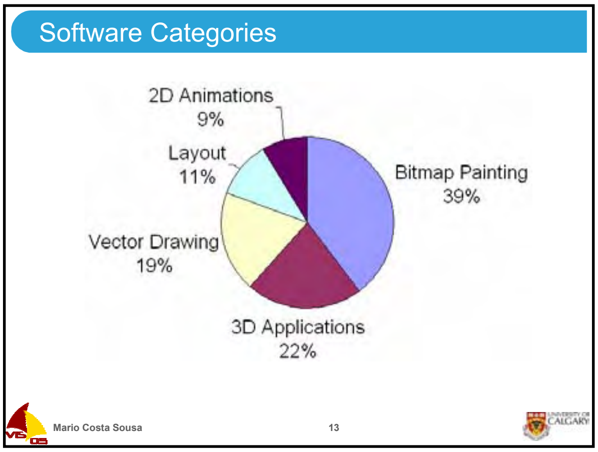
Copyright 2005 Andrew Swift MS CMI. Used by permission.
www.mcg.edu/medart



Copyright 2004 Andrew Swift MS CMI and James Fick MD.
Used by permission.
www.mcg.edu/medart www.swiftillustration.com

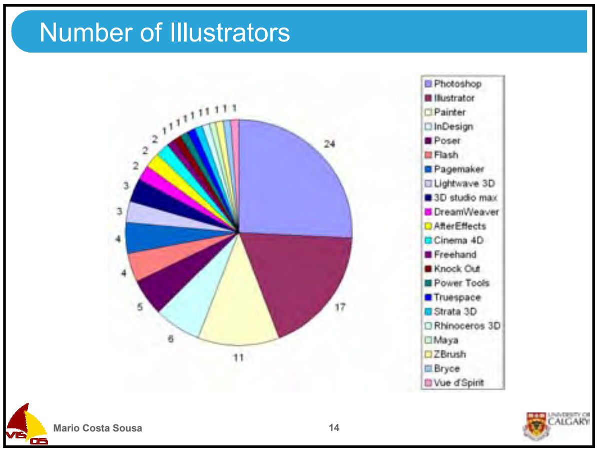
Copyright Medmovie
Used by permission.
medmovie.com

Mario Costa Sousa 12



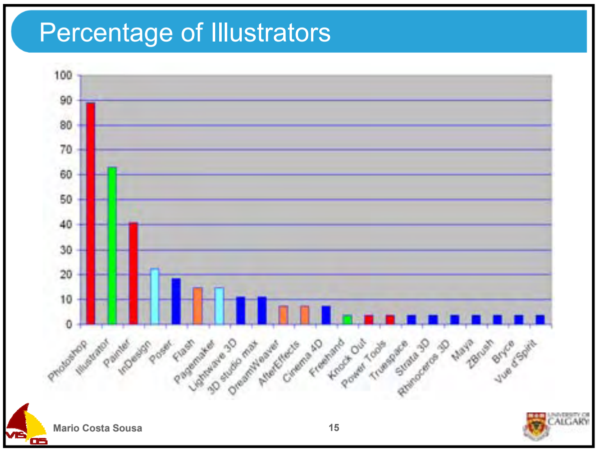
Software Categories

Mario Costa Sousa 13



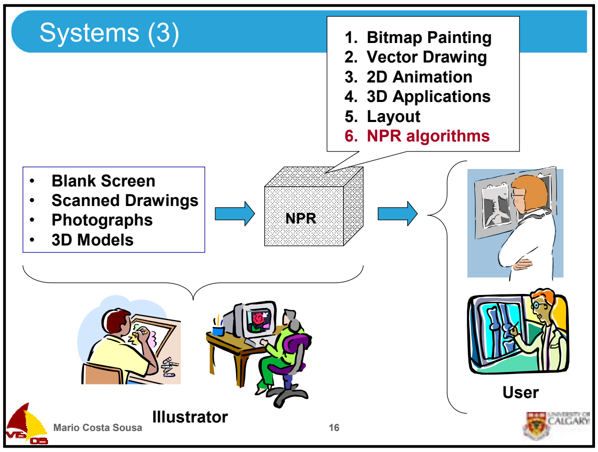
Number of Illustrators

Mario Costa Sousa 14



Percentage of Illustrators

Mario Costa Sousa 15

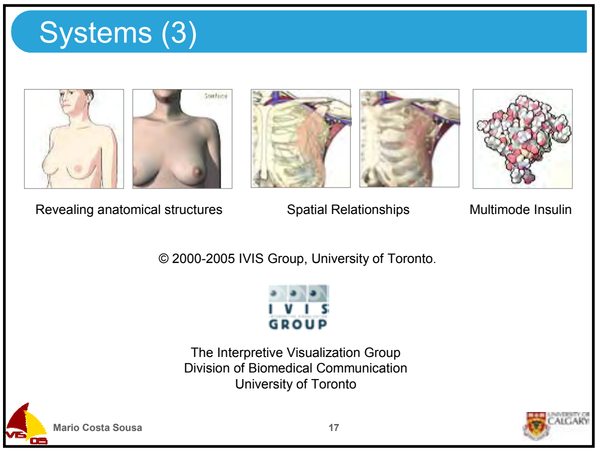


Systems (3)

1. **Bitmap Painting**
2. **Vector Drawing**
3. **2D Animation**
4. **3D Applications**
5. **Layout**
6. **NPR algorithms**

- **Blank Screen**
- **Scanned Drawings**
- **Photographs**
- **3D Models**

Mario Costa Sousa 16



Systems (3)

Revealing anatomical structures

Spatial Relationships

Multimode Insulin

© 2000-2005 IVIS Group, University of Toronto.

I V I S
GROUP

The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto

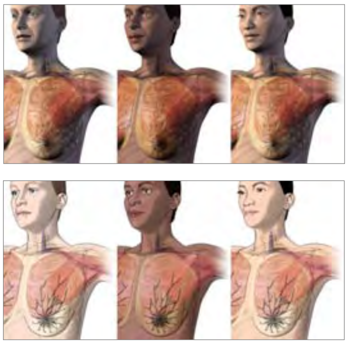
Mario Costa Sousa 17

Systems (3)

http://www.medvisuals.com/medvisuals/headstart/

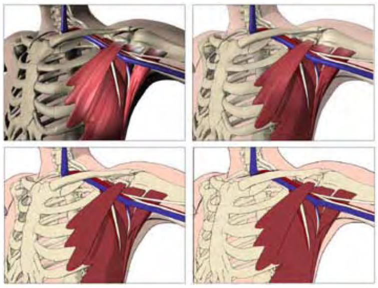
© Jennifer Polk, H.B.Sc., M.Sc.BMC

The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto



© 2000-2005 IVIS Group, University of Toronto.

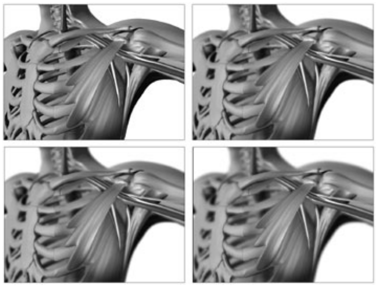
The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto



Visual complexity in anatomical relationships

© 2000-2005 IVIS Group, University of Toronto.

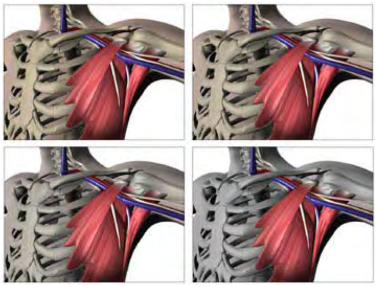
The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto



Depth of field in anatomical relationships

© 2000-2005 IVIS Group, University of Toronto.

The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto

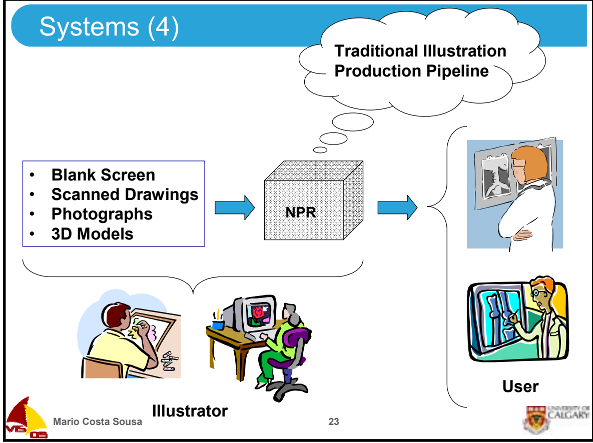


Selective color saturation in anatomical relationships

© 2000-2005 IVIS Group, University of Toronto.

The Interpretive Visualization Group
Division of Biomedical Communication
University of Toronto

Systems (4)



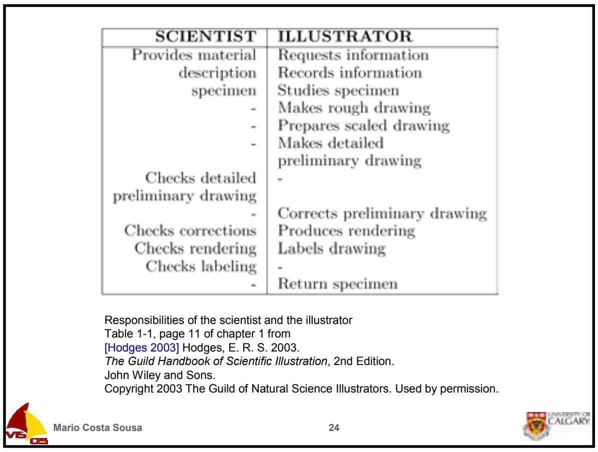
Traditional Illustration Production Pipeline

- Blank Screen
- Scanned Drawings
- Photographs
- 3D Models

NPR

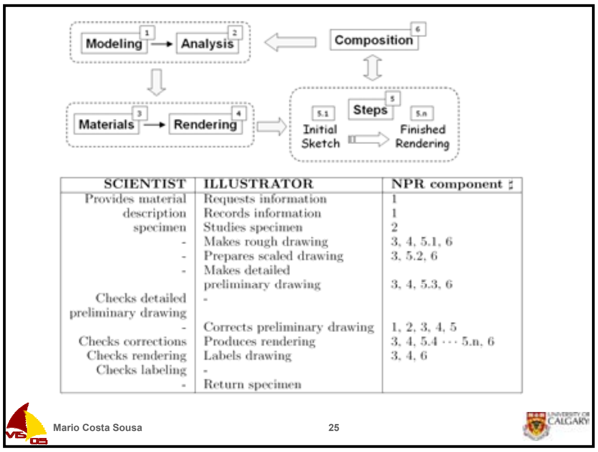
User

Illustrator

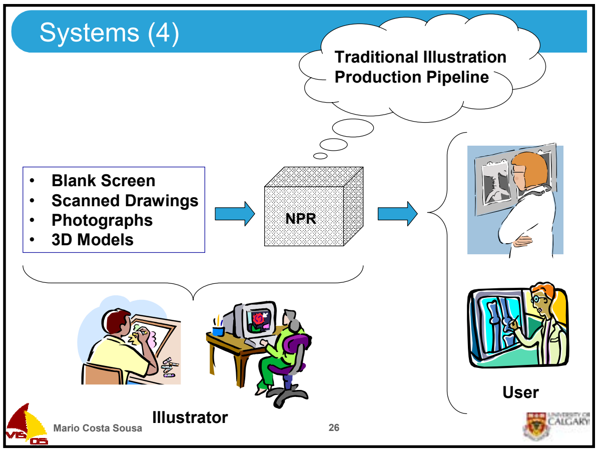


SCIENTIST	ILLUSTRATOR
Provides material	Requests information
description	Records information
specimen	Studies specimen
-	Makes rough drawing
-	Prepares scaled drawing
-	Makes detailed
	preliminary drawing
Checks detailed	-
preliminary drawing	
-	Corrects preliminary drawing
Checks corrections	Produces rendering
Checks rendering	Labels drawing
Checks labeling	-
-	Return specimen

Responsibilities of the scientist and the illustrator
Table 1-1, page 11 of chapter 1 from
[Hodges 2003] Hodges, E. R. S. 2003.
The Guild Handbook of Scientific Illustration, 2nd Edition.
John Wiley and Sons.

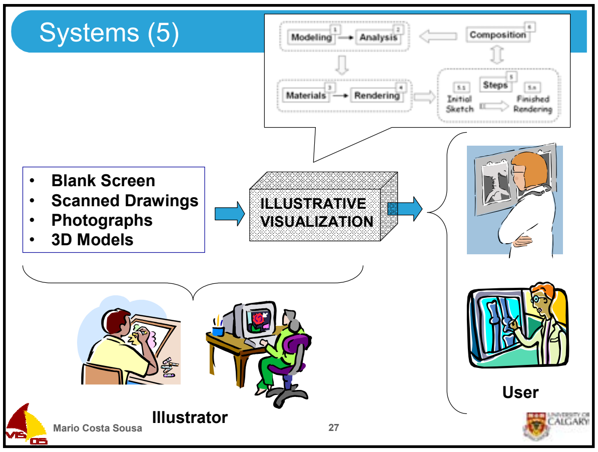


SCIENTIST	ILLUSTRATOR	NPR component ♯
Provides material	Requests information	1
description	Records information	1
specimen	Studies specimen	2
-	Makes rough drawing	3, 4, 5.1, 6
-	Prepares scaled drawing	3, 5.2, 6
-	Makes detailed	
	preliminary drawing	3, 4, 5.3, 6
Checks detailed	-	
preliminary drawing		
-	Corrects preliminary drawing	1, 2, 3, 4, 5
Checks corrections	Produces rendering	3, 4, 5.4 · · · 5.n, 6
Checks rendering	Labels drawing	3, 4, 6
Checks labeling	-	
-	Return specimen	



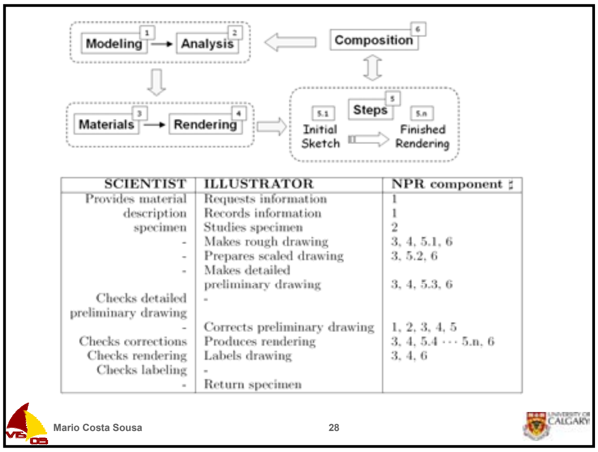
Systems (4)

- Blank Screen
- Scanned Drawings
- Photographs
- 3D Models

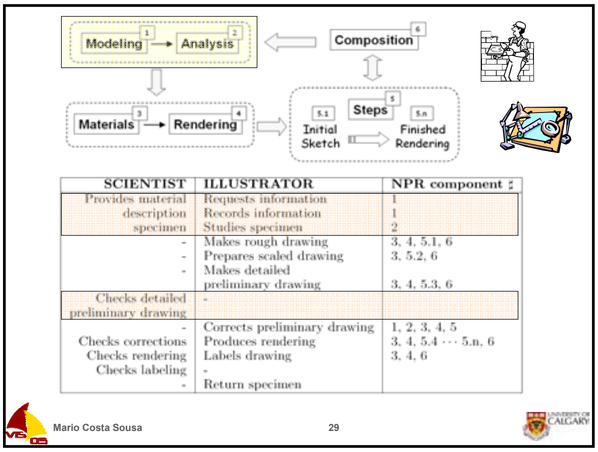


Systems (5)

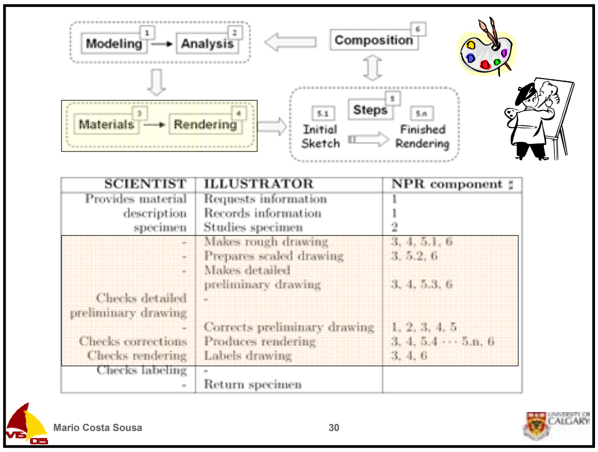
- Blank Screen
- Scanned Drawings
- Photographs
- 3D Models



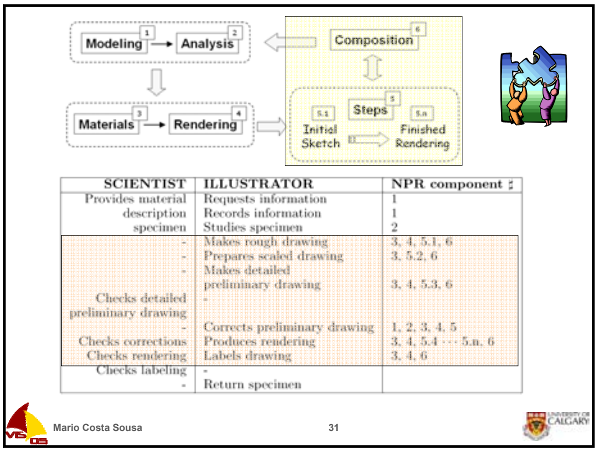
SCIENTIST	ILLUSTRATOR	NPR component ♯
Provides material	Requests information	1
description	Records information	1
specimen	Studies specimen	2
-	Makes rough drawing	3, 4, 5.1, 6
-	Prepares scaled drawing	3, 5.2, 6
-	Makes detailed	
	preliminary drawing	3, 4, 5.3, 6
Checks detailed	-	
preliminary drawing		
-	Corrects preliminary drawing	1, 2, 3, 4, 5
Checks corrections	Produces rendering	3, 4, 5.4 · · · 5.n, 6
Checks rendering	Labels drawing	3, 4, 6
Checks labeling	-	
-	Return specimen	



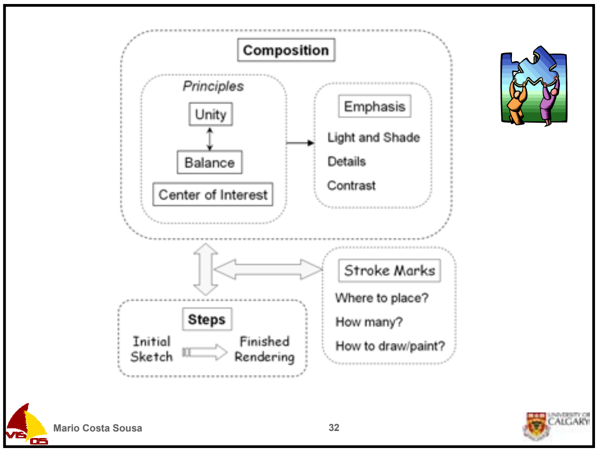
SCIENTIST	ILLUSTRATOR	NPR component ♯
Provides material	Requests information	1
description	Records information	1
specimen	Studies specimen	2
-	Makes rough drawing	3, 4, 5.1, 6
-	Prepares scaled drawing	3, 5.2, 6
-	Makes detailed	
	preliminary drawing	3, 4, 5.3, 6
Checks detailed	-	
preliminary drawing		
-	Corrects preliminary drawing	1, 2, 3, 4, 5
Checks corrections	Produces rendering	3, 4, 5.4 · · · 5.n, 6
Checks rendering	Labels drawing	3, 4, 6
Checks labeling	-	
-	Return specimen	

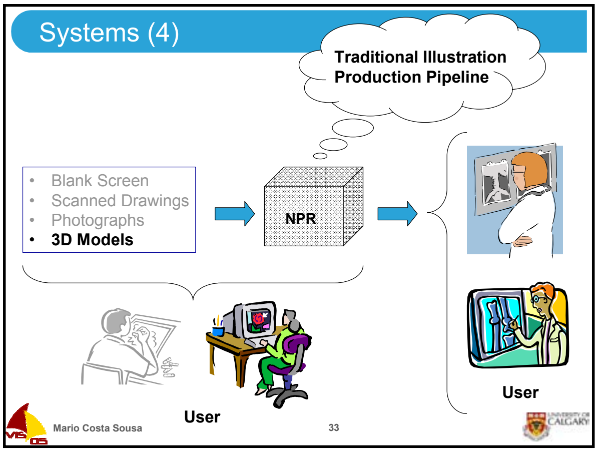


SCIENTIST	ILLUSTRATOR	NPR component ♯
Provides material	Requests information	1
description	Records information	1
specimen	Studies specimen	2
-	Makes rough drawing	3, 4, 5.1, 6
-	Prepares scaled drawing	3, 5.2, 6
-	Makes detailed preliminary drawing	3, 4, 5.3, 6
Checks detailed preliminary drawing	-	
-	Corrects preliminary drawing	1, 2, 3, 4, 5
Checks corrections	Produces rendering	3, 4, 5.4 ··· 5.n, 6
Checks rendering	Labels drawing	3, 4, 6
Checks labeling	-	
-	Return specimen	



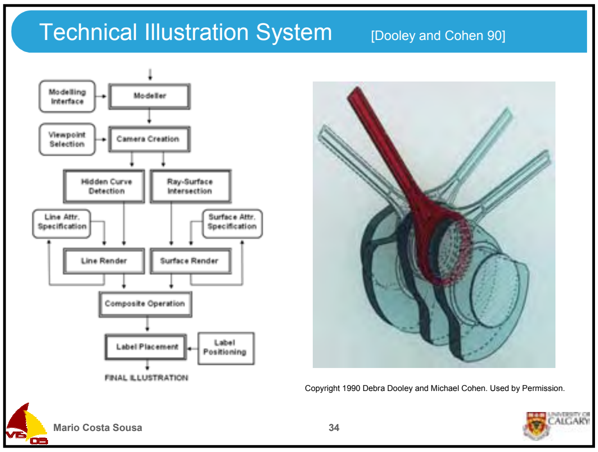
SCIENTIST	ILLUSTRATOR	NPR component ♯
Provides material	Requests information	1
description	Records information	1
specimen	Studies specimen	2
-	Makes rough drawing	3, 4, 5.1, 6
-	Prepares scaled drawing	3, 5.2, 6
-	Makes detailed preliminary drawing	3, 4, 5.3, 6
Checks detailed preliminary drawing	-	
-	Corrects preliminary drawing	1, 2, 3, 4, 5
Checks corrections	Produces rendering	3, 4, 5.4 ··· 5.n, 6
Checks rendering	Labels drawing	3, 4, 6
Checks labeling	-	
-	Return specimen	





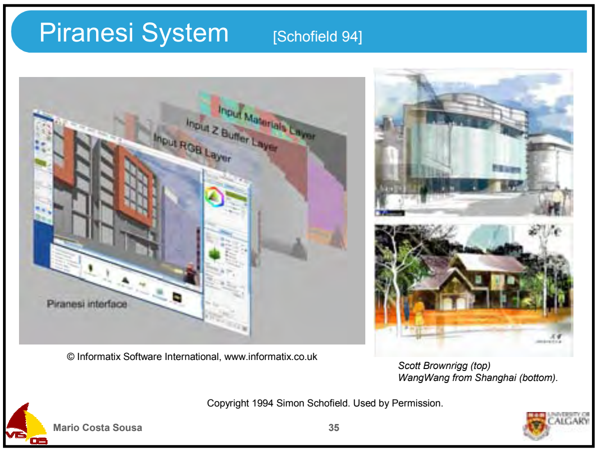
Systems (4)

- Blank Screen
- Scanned Drawings
- Photographs
- **3D Models**



Technical Illustration System [Dooley and Cohen 90]

Copyright 1990 Debra Dooley and Michael Cohen. Used by Permission.

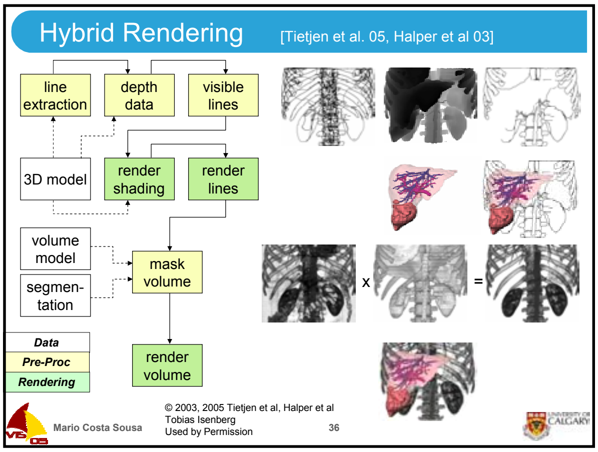


Piranesi System [Schofield 94]

© Informatix Software International, www.informatix.co.uk

Scott Brownrigg (top)
WangWang from Shanghai (bottom).

Copyright 1994 Simon Schofield. Used by Permission.



Hybrid Rendering [Tietjen et al. 05, Halper et al 03]

line extraction

depth data

visible lines

3D model

render shading

render lines

volume model

segmen-tation

mask volume

render volume

Data

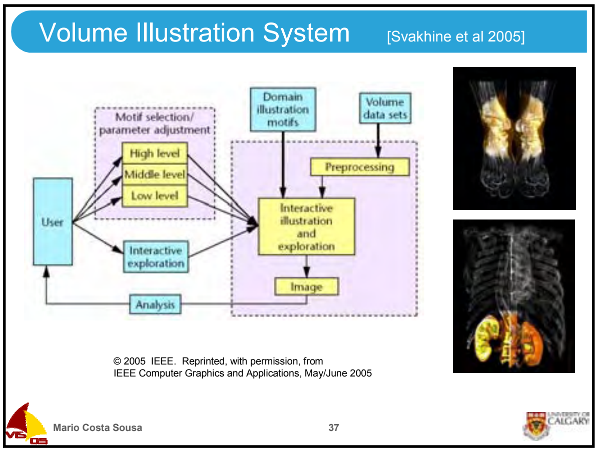
Pre-Proc

Rendering

x

=

© 2003, 2005 Tietjen et al, Halper et al
Tobias Isenberg
Used by Permission



Volume Illustration System [Svakhine et al 2005]

Motif selection/ parameter adjustment

High level

Middle level

Low level

User

Interactive exploration

Domain illustration motifs

Volume data sets

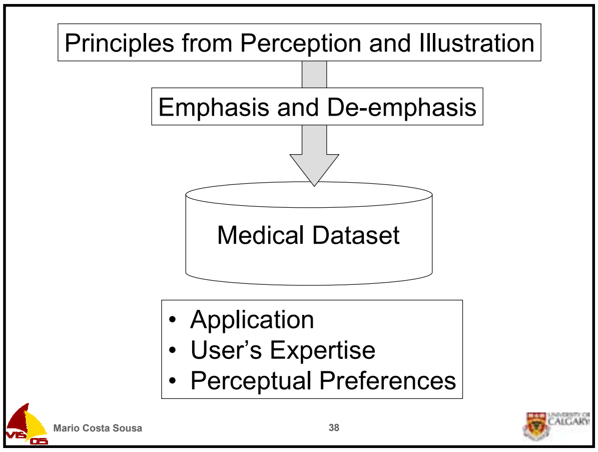
Preprocessing

Interactive illustration and exploration

Image

Analysis

© 2005 IEEE. Reprinted, with permission, from
IEEE Computer Graphics and Applications, May/June 2005

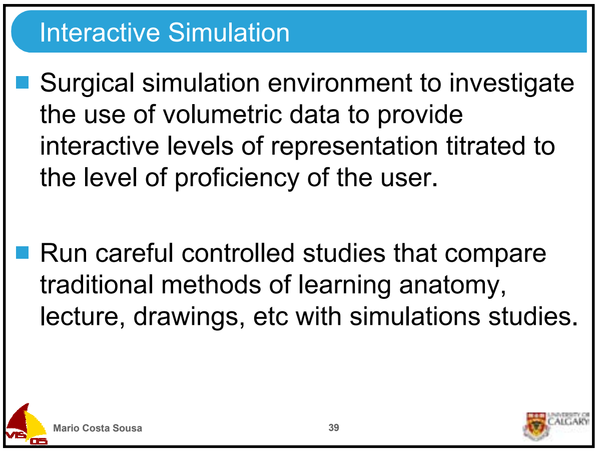


Principles from Perception and Illustration

Emphasis and De-emphasis

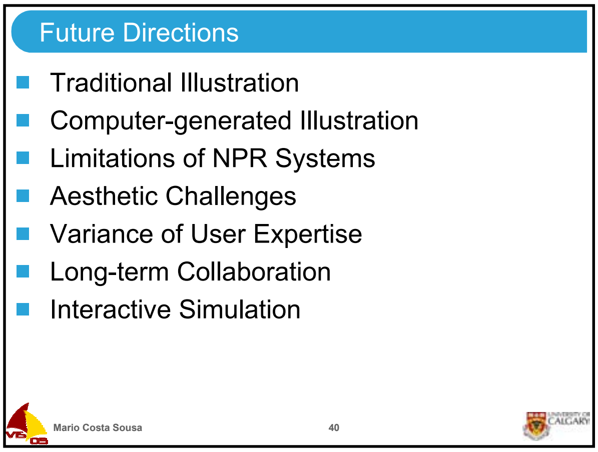
Medical Dataset

- Application
- User's Expertise
- Perceptual Preferences



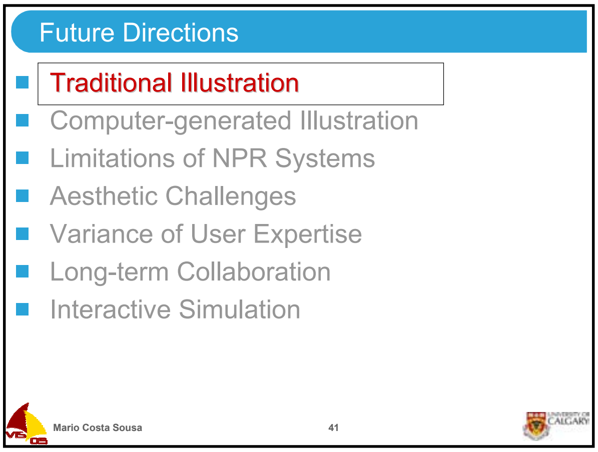
Interactive Simulation

- Surgical simulation environment to investigate the use of volumetric data to provide interactive levels of representation titrated to the level of proficiency of the user.
- Run careful controlled studies that compare traditional methods of learning anatomy, lecture, drawings, etc with simulations studies.



Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation

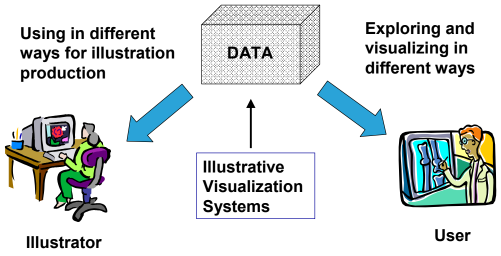
Traditional Illustrator

- Subtle understanding of how to:
  - Effectively manipulate the media to create subtle cues to represent (abstract or realistic)
  - Include ways to emphasize or subjugate information to effectively communicate various messages to their viewer.

Future Directions

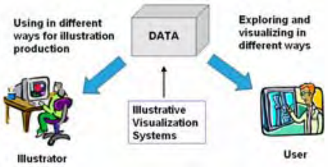
- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation

Computer-generated Illustration


Using in different ways for illustration production
DATA
Exploring and visualizing in different ways
Illustrative Visualization Systems
Illustrator
User

Computer-generated Illustration

- Use of patient-specific data
- Integration of NPR techniques
- The combination of direct volume rendering and NPR techniques

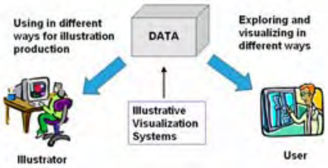

Using in different ways for illustration production
DATA
Exploring and visualizing in different ways
Illustrative Visualization Systems
Illustrator
User

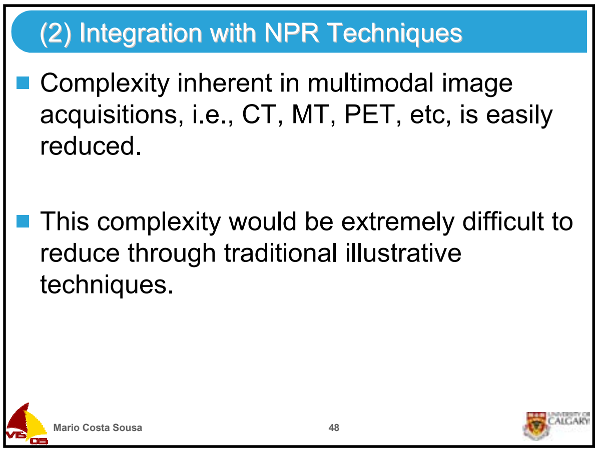
(1) Use of patient-specific data

- Direct volume-volume rendering
- Actual case examples of phenotypical variation in anatomy as well as pathological variations can be presented.
- More salient and relevant to clinical research and medical education.
- The value in patient education when employed to convey diagnostic and treatment plans to patients and/or their families is self-evident.

Computer-generated Illustration

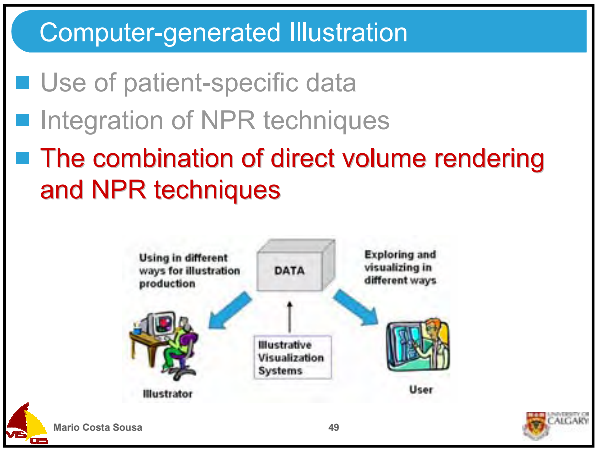
- Use of patient-specific data
- Integration of NPR techniques
- The combination of direct volume rendering and NPR techniques


Using in different ways for illustration production
DATA
Exploring and visualizing in different ways
Illustrative Visualization Systems
Illustrator
User



(2) Integration with NPR Techniques

- Complexity inherent in multimodal image acquisitions, i.e., CT, MT, PET, etc, is easily reduced.
- This complexity would be extremely difficult to reduce through traditional illustrative techniques.



Computer-generated Illustration

- Use of patient-specific data
- Integration of NPR techniques
- The combination of direct volume rendering and NPR techniques

Using in different ways for illustration production

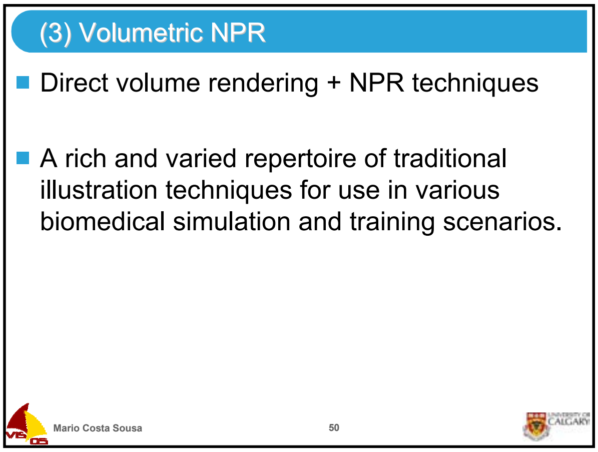
DATA

Exploring and visualizing in different ways

Illustrative Visualization Systems

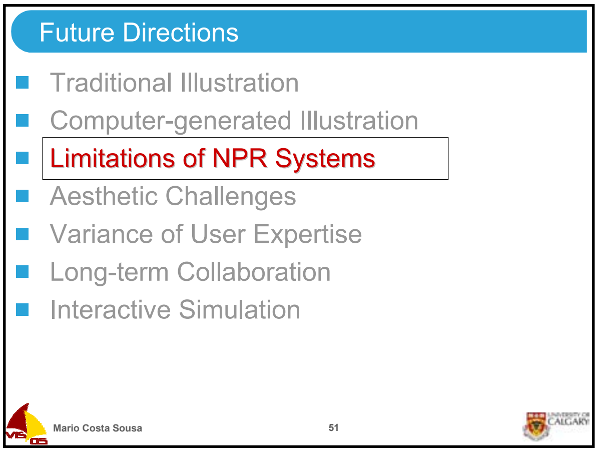
Illustrator

User



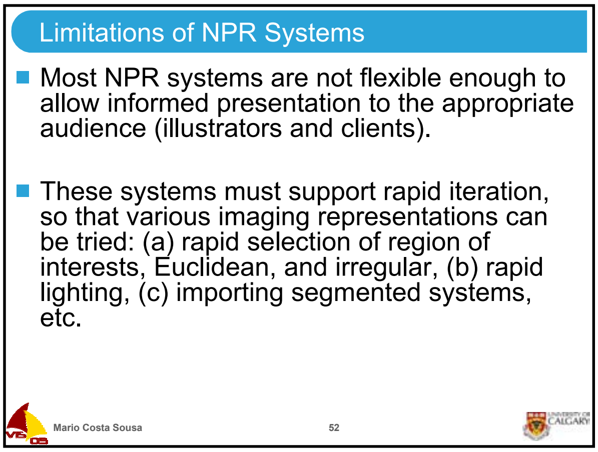
(3) Volumetric NPR

- Direct volume rendering + NPR techniques
- A rich and varied repertoire of traditional illustration techniques for use in various biomedical simulation and training scenarios.



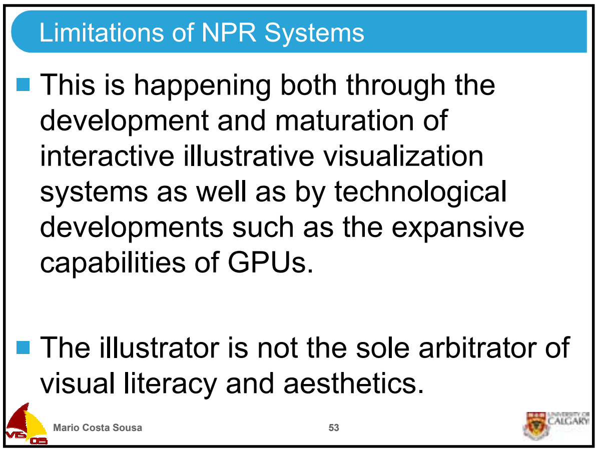
Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



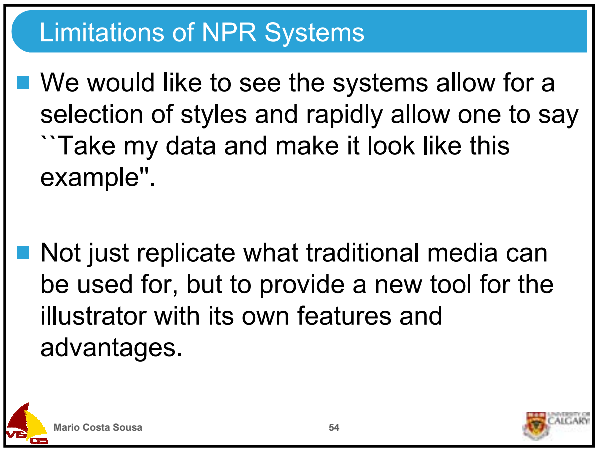
Limitations of NPR Systems

- Most NPR systems are not flexible enough to allow informed presentation to the appropriate audience (illustrators and clients).
- These systems must support rapid iteration, so that various imaging representations can be tried: (a) rapid selection of region of interests, Euclidean, and irregular, (b) rapid lighting, (c) importing segmented systems, etc.



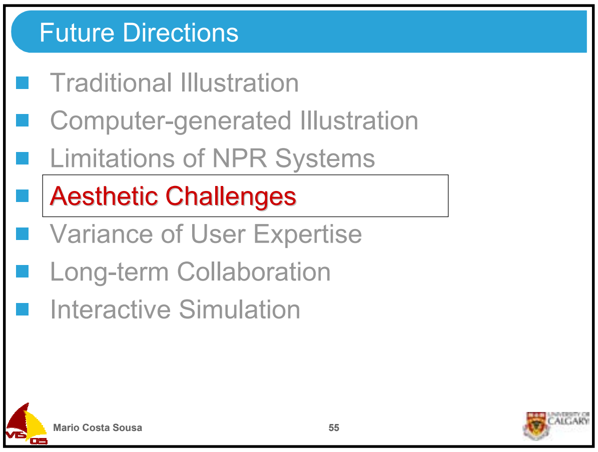
Limitations of NPR Systems

- This is happening both through the development and maturation of interactive illustrative visualization systems as well as by technological developments such as the expansive capabilities of GPUs.
- The illustrator is not the sole arbitrator of visual literacy and aesthetics.



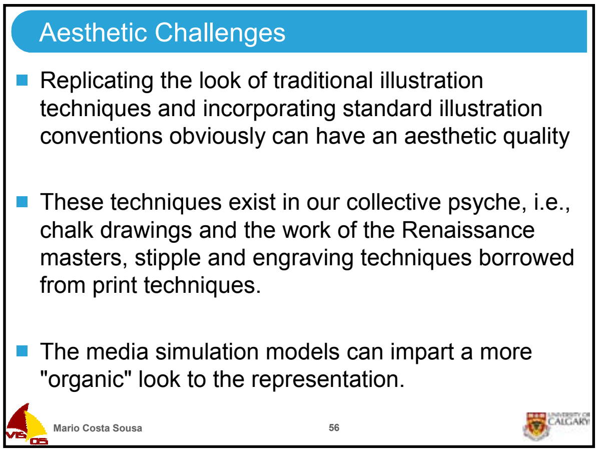
Limitations of NPR Systems

- We would like to see the systems allow for a selection of styles and rapidly allow one to say ``Take my data and make it look like this example".
- Not just replicate what traditional media can be used for, but to provide a new tool for the illustrator with its own features and advantages.



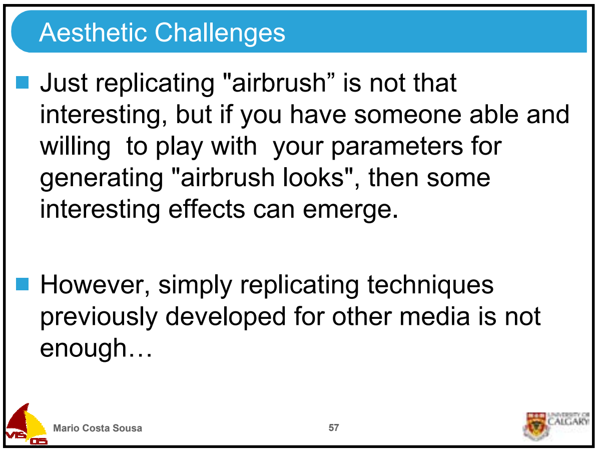
Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



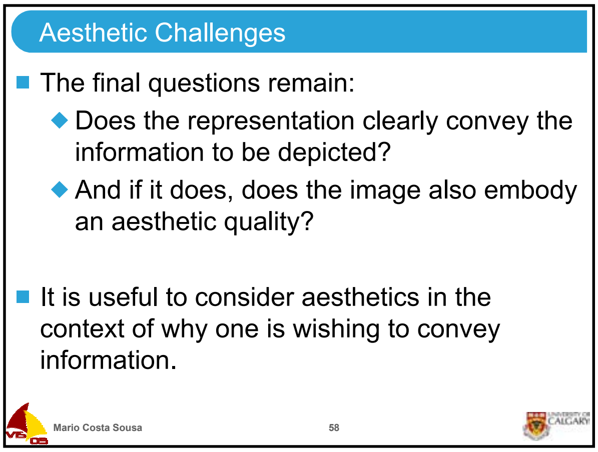
Aesthetic Challenges

- Replicating the look of traditional illustration techniques and incorporating standard illustration conventions obviously can have an aesthetic quality
- These techniques exist in our collective psyche, i.e., chalk drawings and the work of the Renaissance masters, stipple and engraving techniques borrowed from print techniques.
- The media simulation models can impart a more "organic" look to the representation.



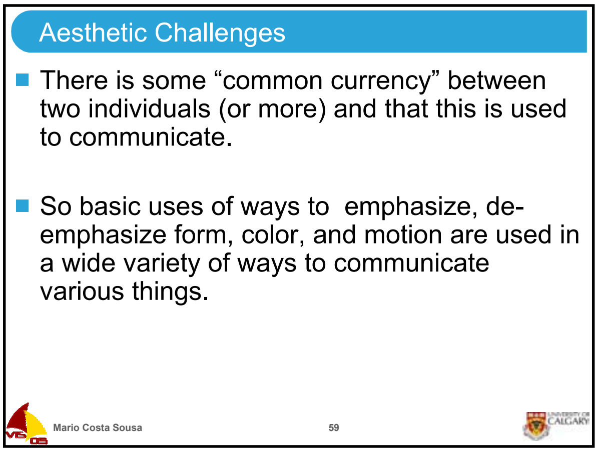
Aesthetic Challenges

- Just replicating "airbrush” is not that interesting, but if you have someone able and willing to play with your parameters for generating "airbrush looks", then some interesting effects can emerge.
- However, simply replicating techniques previously developed for other media is not enough…



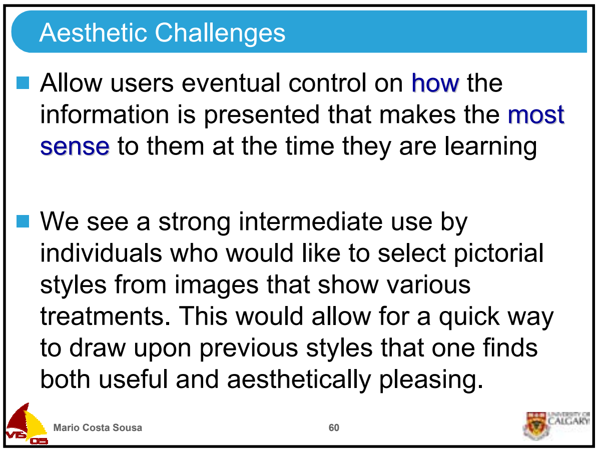
Aesthetic Challenges

- The final questions remain:
  - Does the representation clearly convey the information to be depicted?
  - And if it does, does the image also embody an aesthetic quality?
- It is useful to consider aesthetics in the context of why one is wishing to convey information.



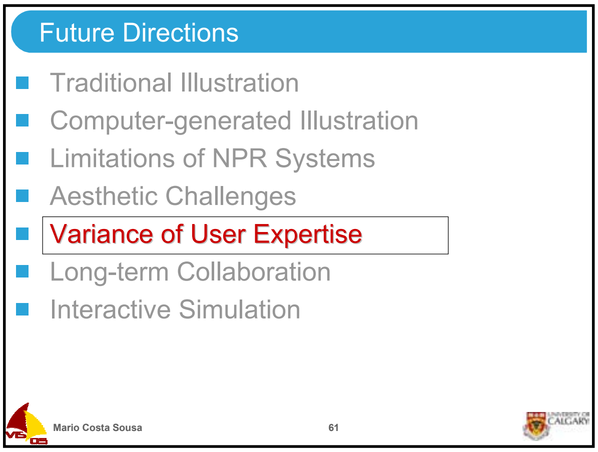
Aesthetic Challenges

- There is some “common currency” between two individuals (or more) and that this is used to communicate.
- So basic uses of ways to emphasize, de-emphasize form, color, and motion are used in a wide variety of ways to communicate various things.



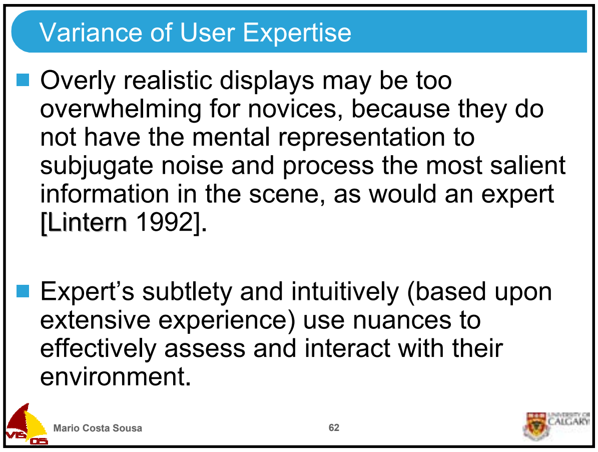
Aesthetic Challenges

- Allow users eventual control on how the information is presented that makes the most sense to them at the time they are learning
- We see a strong intermediate use by individuals who would like to select pictorial styles from images that show various treatments. This would allow for a quick way to draw upon previous styles that one finds both useful and aesthetically pleasing.



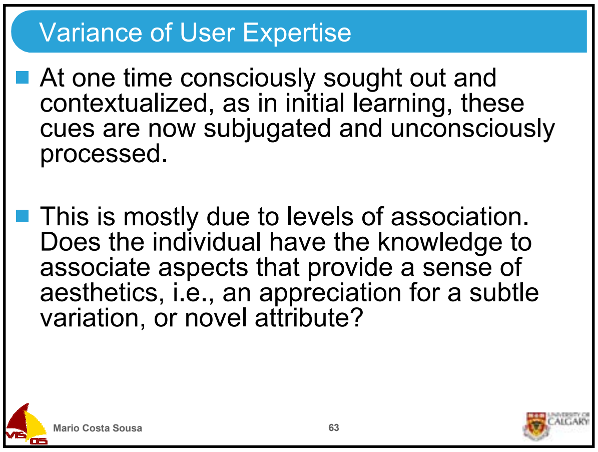
Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



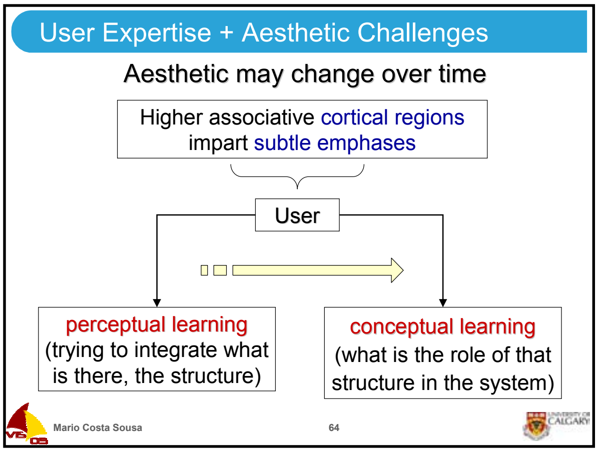
Variance of User Expertise

- Overly realistic displays may be too overwhelming for novices, because they do not have the mental representation to subjugate noise and process the most salient information in the scene, as would an expert [Lintern 1992].
- Expert's subtlety and intuitively (based upon extensive experience) use nuances to effectively assess and interact with their environment.



Variance of User Expertise

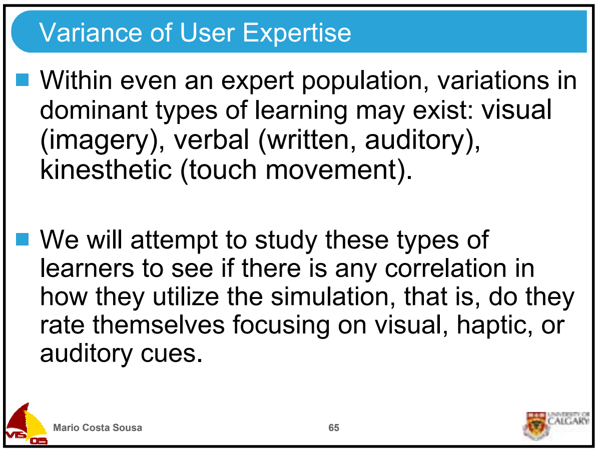
- At one time consciously sought out and contextualized, as in initial learning, these cues are now subjugated and unconsciously processed.
- This is mostly due to levels of association. Does the individual have the knowledge to associate aspects that provide a sense of aesthetics, i.e., an appreciation for a subtle variation, or novel attribute?



User Expertise + Aesthetic Challenges

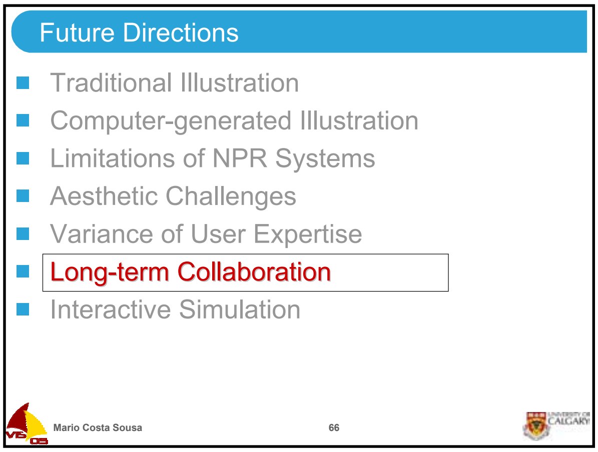
Aesthetic may change over time

Higher associative cortical regions impart subtle emphases
User
perceptual learning (trying to integrate what is there, the structure)
conceptual learning (what is the role of that structure in the system)



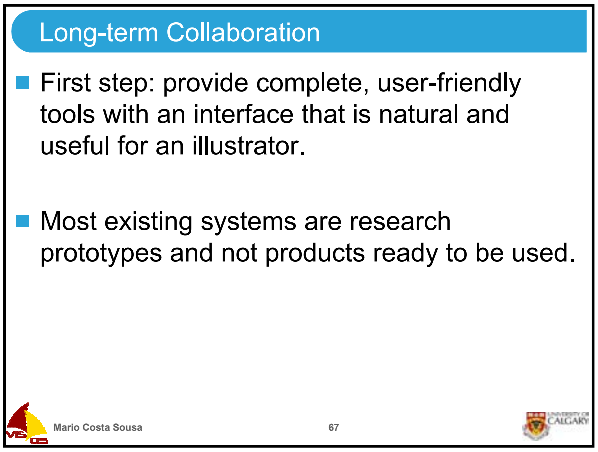
Variance of User Expertise

- Within even an expert population, variations in dominant types of learning may exist: visual (imagery), verbal (written, auditory), kinesthetic (touch movement).
- We will attempt to study these types of learners to see if there is any correlation in how they utilize the simulation, that is, do they rate themselves focusing on visual, haptic, or auditory cues.



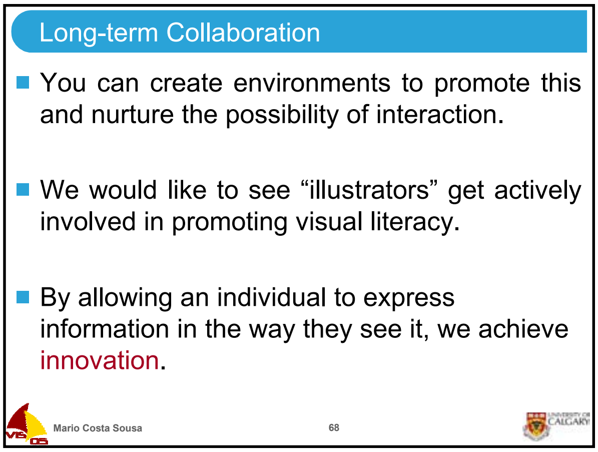
Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



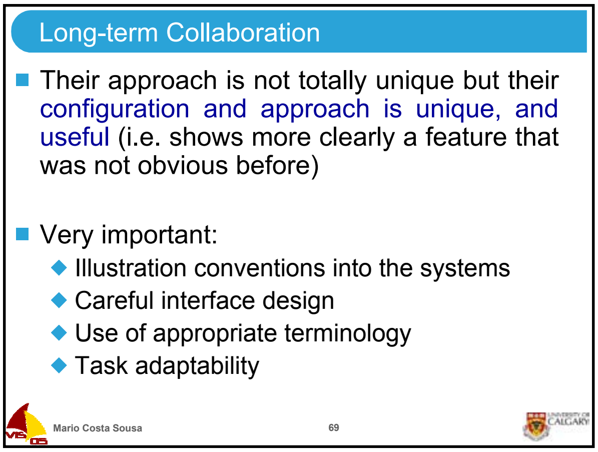
Long-term Collaboration

- First step: provide complete, user-friendly tools with an interface that is natural and useful for an illustrator.
- Most existing systems are research prototypes and not products ready to be used.



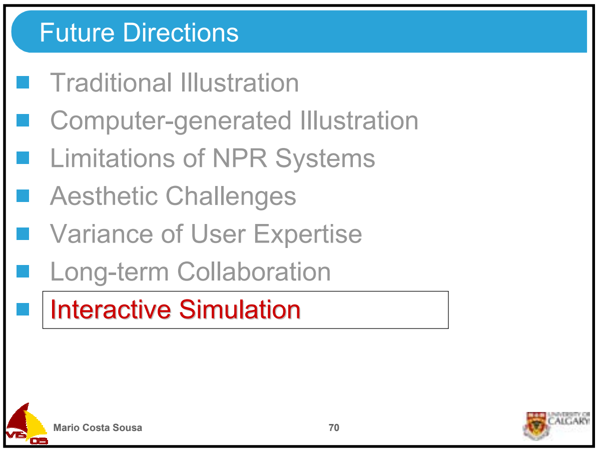
Long-term Collaboration

- You can create environments to promote this and nurture the possibility of interaction.
- We would like to see “illustrators” get actively involved in promoting visual literacy.
- By allowing an individual to express information in the way they see it, we achieve innovation.



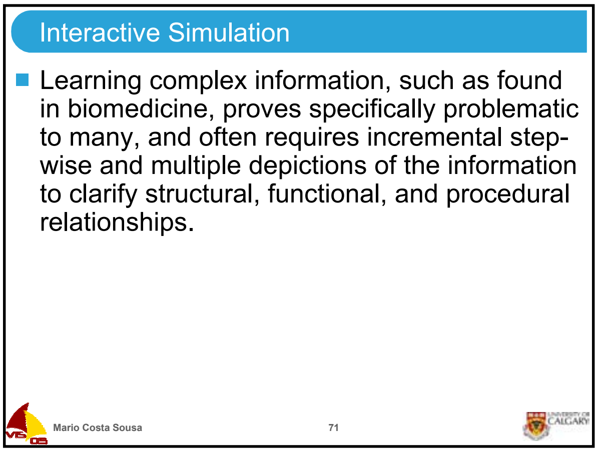
Long-term Collaboration

- Their approach is not totally unique but their configuration and approach is unique, and useful (i.e. shows more clearly a feature that was not obvious before)
- Very important:
  - Illustration conventions into the systems
  - Careful interface design
  - Use of appropriate terminology
  - Task adaptability



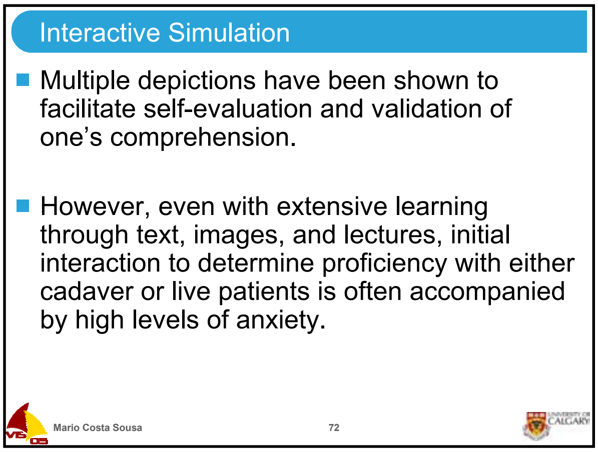
Future Directions

- Traditional Illustration
- Computer-generated Illustration
- Limitations of NPR Systems
- Aesthetic Challenges
- Variance of User Expertise
- Long-term Collaboration
- Interactive Simulation



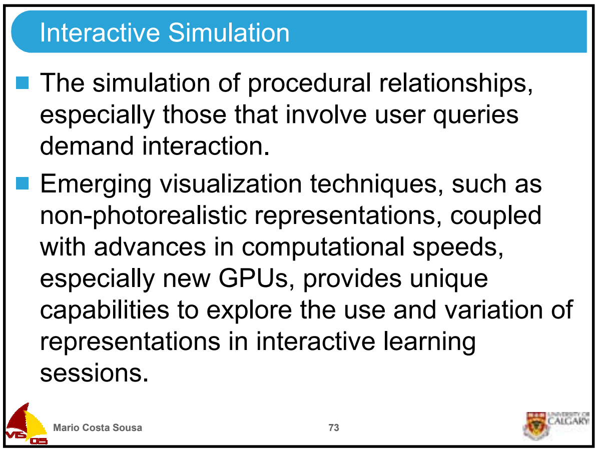
Interactive Simulation

- Learning complex information, such as found in biomedicine, proves specifically problematic to many, and often requires incremental step-wise and multiple depictions of the information to clarify structural, functional, and procedural relationships.



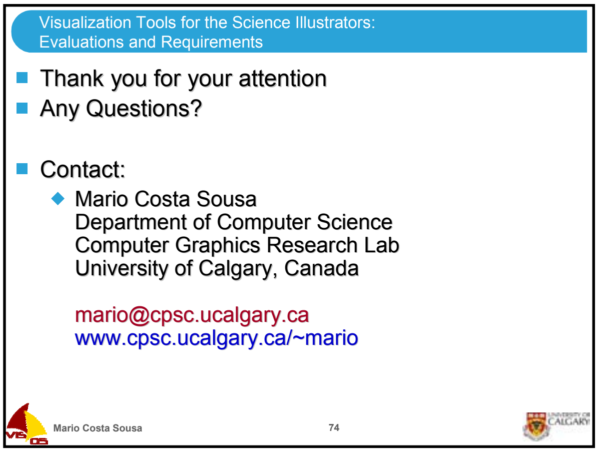
Interactive Simulation

- Multiple depictions have been shown to facilitate self-evaluation and validation of one's comprehension.
- However, even with extensive learning through text, images, and lectures, initial interaction to determine proficiency with either cadaver or live patients is often accompanied by high levels of anxiety.



Interactive Simulation

- The simulation of procedural relationships, especially those that involve user queries demand interaction.
- Emerging visualization techniques, such as non-photorealistic representations, coupled with advances in computational speeds, especially new GPUs, provides unique capabilities to explore the use and variation of representations in interactive learning sessions.

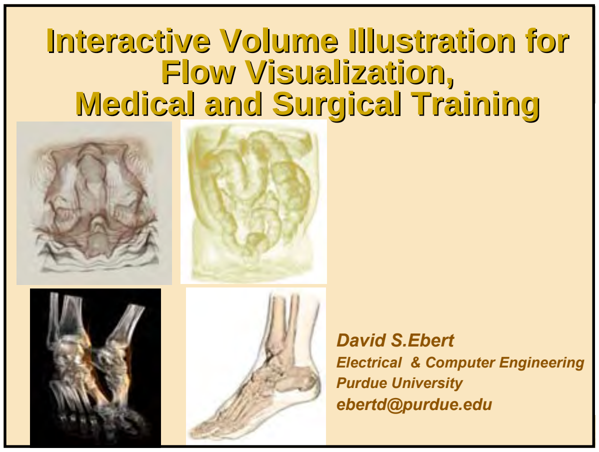


Visualization Tools for the Science Illustrators: Evaluations and Requirements

- Thank you for your attention
- Any Questions?
- Contact:
  - Mario Costa Sousa
    Department of Computer Science
    Computer Graphics Research Lab
    University of Calgary, Canada

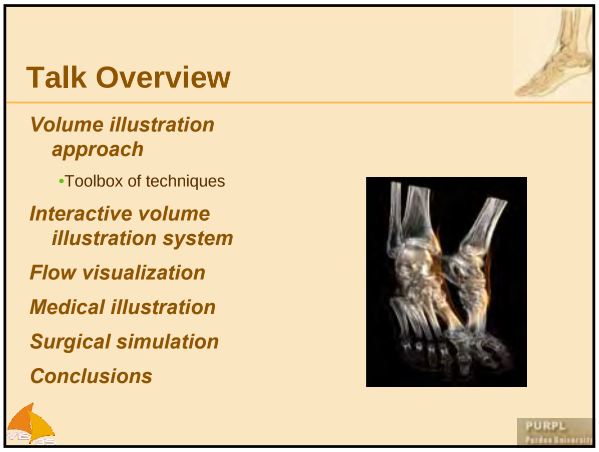
    mario@cpsc.ucalgary.ca
    www.cpsc.ucalgary.ca/~mario

Interactive Volume Illustration for Flow Visualization Medical and Surgical Training



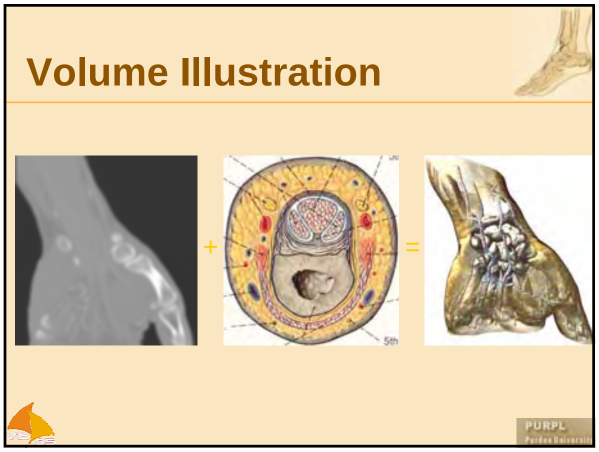
Interactive Volume Illustration for Flow Visualization, Medical and Surgical Training

David S.Ebert
Electrical & Computer Engineering
Purdue University
ebertd@purdue.edu

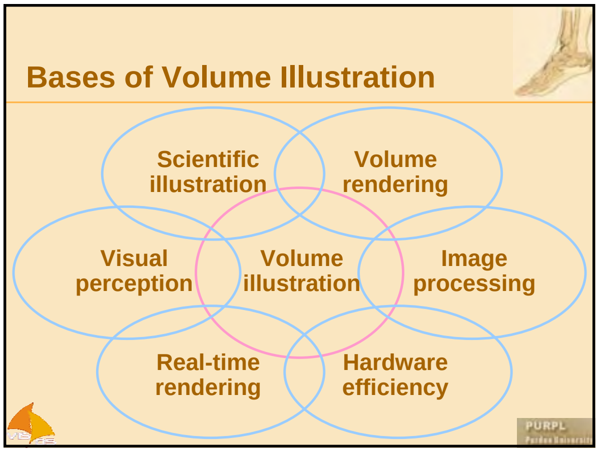


Talk Overview

- *Volume illustration approach*
  - Toolbox of techniques
- *Interactive volume illustration system*
- *Flow visualization*
- *Medical illustration*
- *Surgical simulation*
- *Conclusions*



Volume Illustration



Bases of Volume Illustration

Scientific illustration

Volume rendering

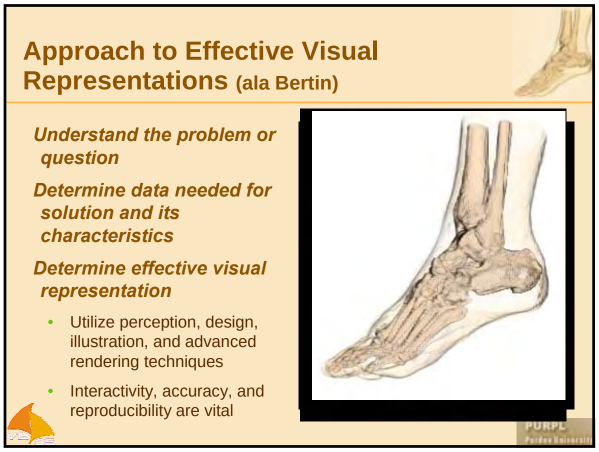
Visual perception

Volume illustration

Image processing

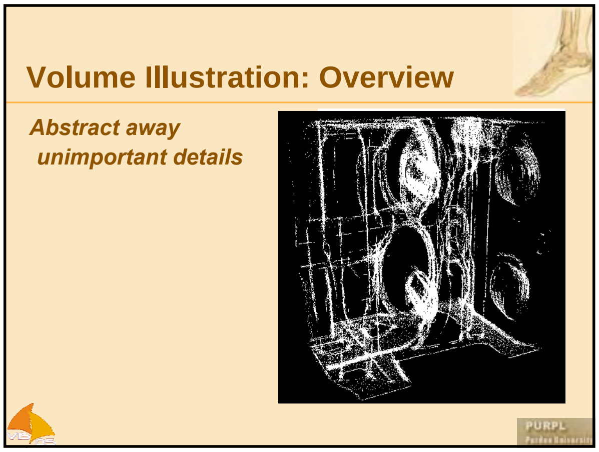
Real-time rendering

Hardware efficiency



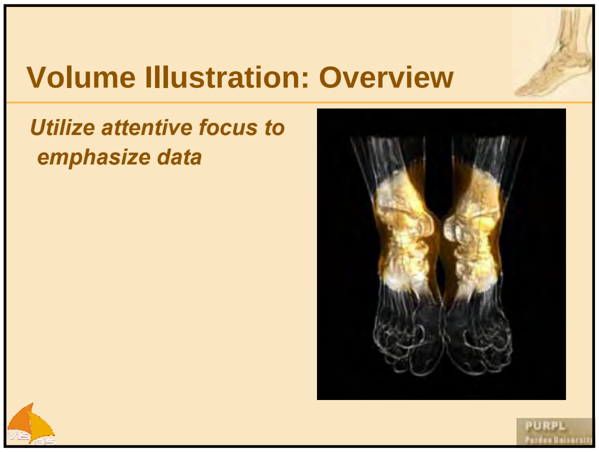
Approach to Effective Visual Representations (ala Bertin)

- *Understand the problem or question*
- *Determine data needed for solution and its characteristics*
- *Determine effective visual representation*
  - Utilize perception, design, illustration, and advanced rendering techniques
  - Interactivity, accuracy, and reproducibility are vital



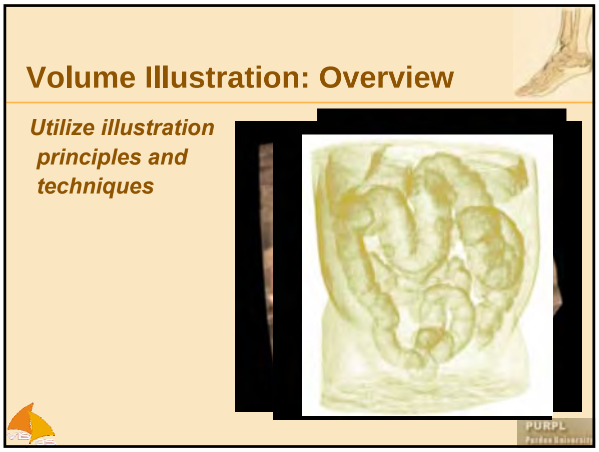
Volume Illustration: Overview

- *Abstract away unimportant details*



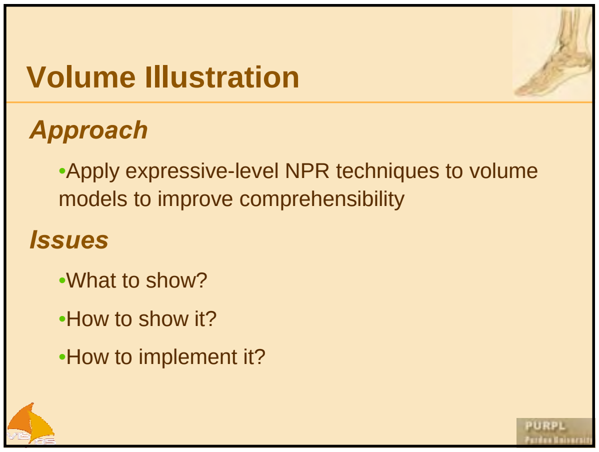
Volume Illustration: Overview

Utilize attentive focus to emphasize data



Volume Illustration: Overview

Utilize illustration principles and techniques



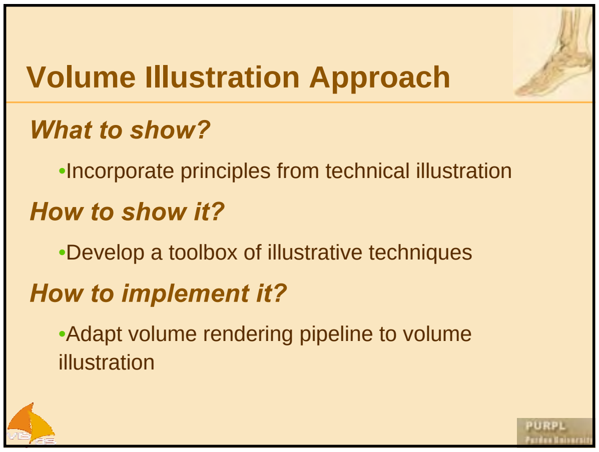
Volume Illustration

Approach

- Apply expressive-level NPR techniques to volume models to improve comprehensibility

Issues

- What to show?
- How to show it?
- How to implement it?



Volume Illustration Approach

What to show?

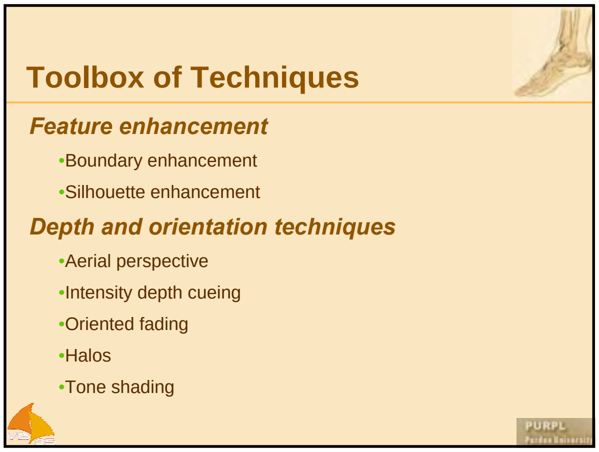
- Incorporate principles from technical illustration

How to show it?

- Develop a toolbox of illustrative techniques

How to implement it?

- Adapt volume rendering pipeline to volume illustration



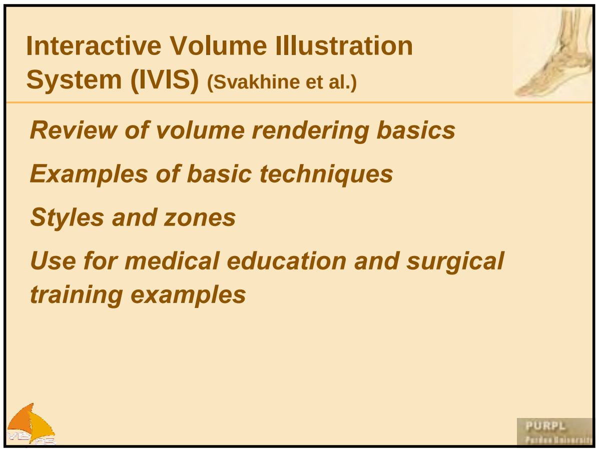
Toolbox of Techniques

Feature enhancement

- Boundary enhancement
- Silhouette enhancement

Depth and orientation techniques

- Aerial perspective
- Intensity depth cueing
- Oriented fading
- Halos
- Tone shading



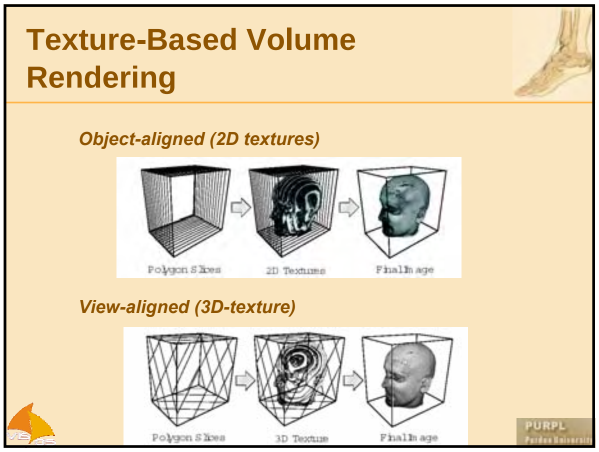
Interactive Volume Illustration System (IVIS) (Svakhine et al.)

Review of volume rendering basics

Examples of basic techniques

Styles and zones

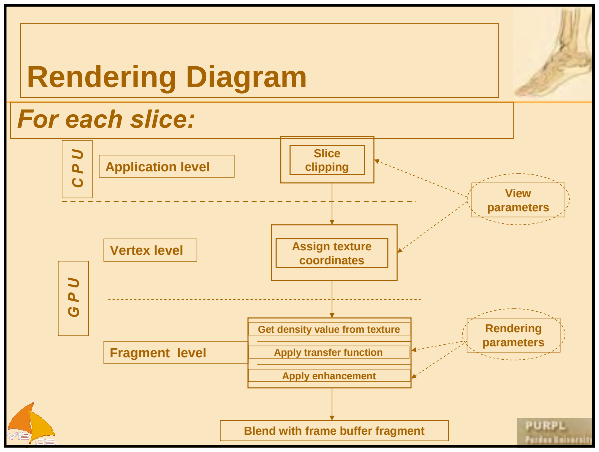
Use for medical education and surgical training examples



Texture-Based Volume Rendering

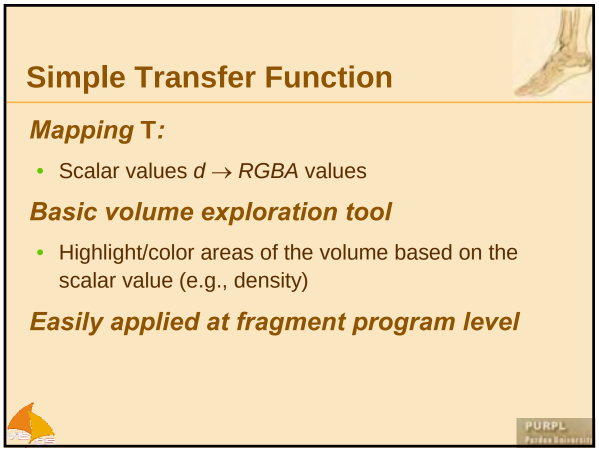
Object-aligned (2D textures)

View-aligned (3D-texture)



Rendering Diagram

For each slice:



Simple Transfer Function

Mapping* T*:

- Scalar values *d* → *RGBA* values

Basic volume exploration tool

- Highlight/color areas of the volume based on the scalar value (e.g., density)

Easily applied at fragment program level

Illustrative Enhancements

Boundary enhancement

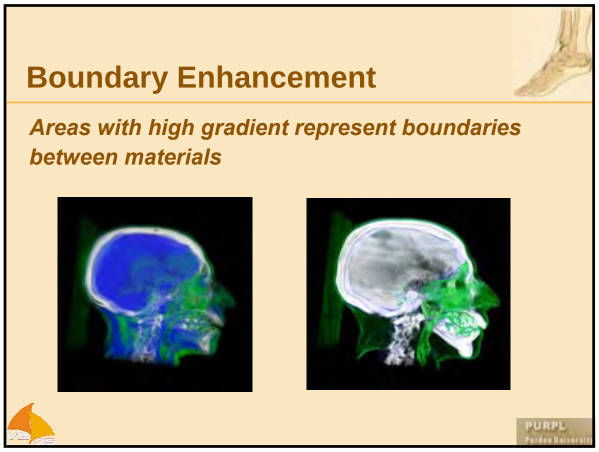
Sketching

Edge coloring

Tone shading / illumination

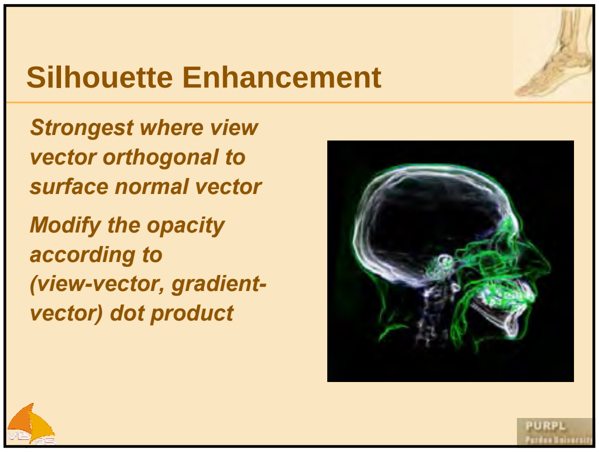
Distance color blending

Feature halos



Boundary Enhancement

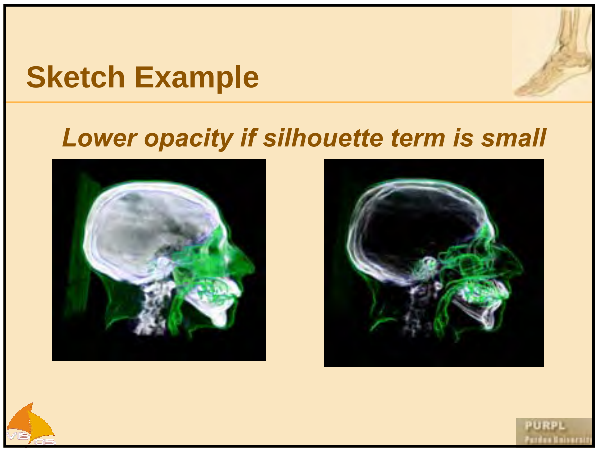
Areas with high gradient represent boundaries between materials



Silhouette Enhancement

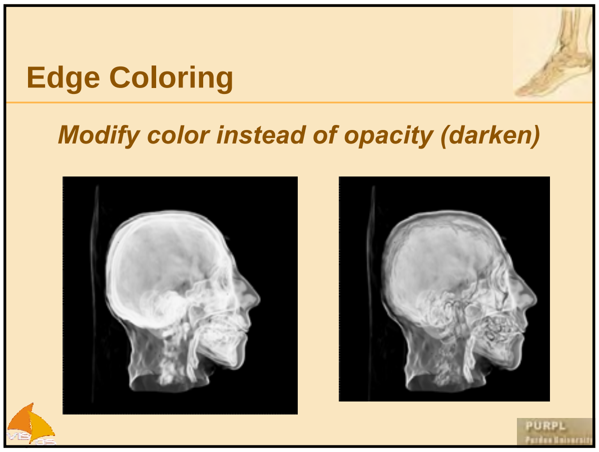
Strongest where view vector orthogonal to surface normal vector

Modify the opacity according to (view-vector, gradient-vector) dot product



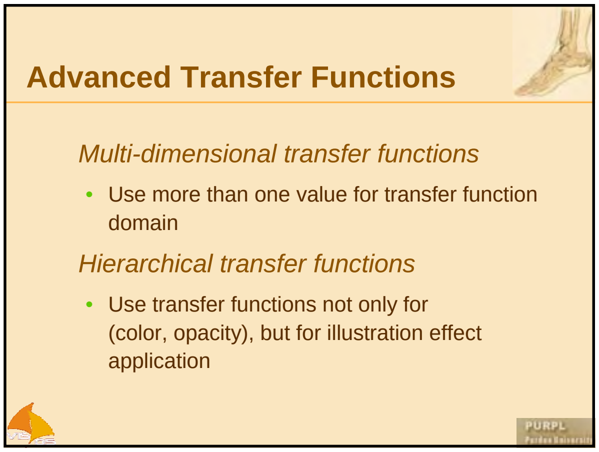
Sketch Example

Lower opacity if silhouette term is small



Edge Coloring

Modify color instead of opacity (darken)



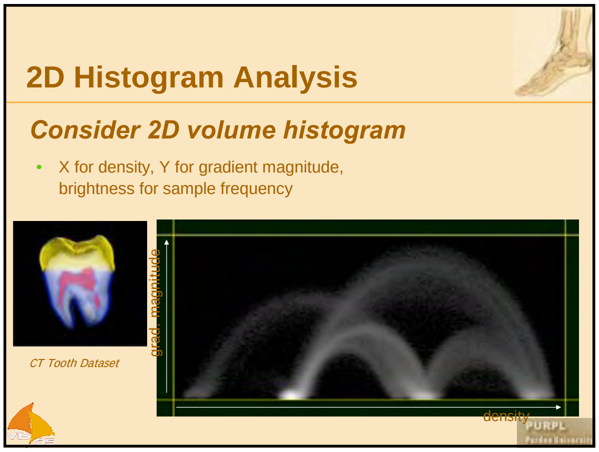
Advanced Transfer Functions

Multi-dimensional transfer functions

- Use more than one value for transfer function domain

Hierarchical transfer functions

- Use transfer functions not only for (color, opacity), but for illustration effect application

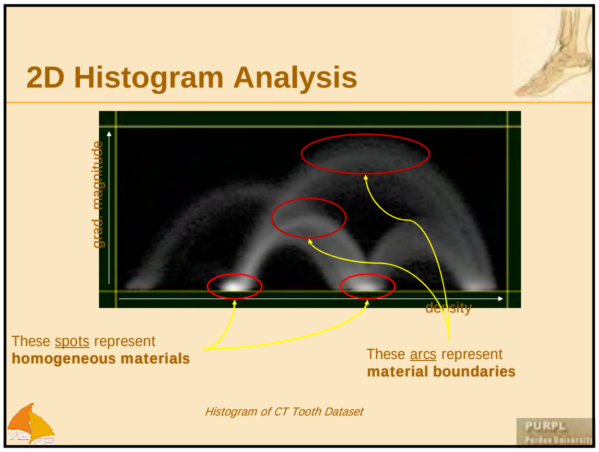


2D Histogram Analysis

Consider 2D volume histogram

- X for density, Y for gradient magnitude, brightness for sample frequency

CT Tooth Dataset

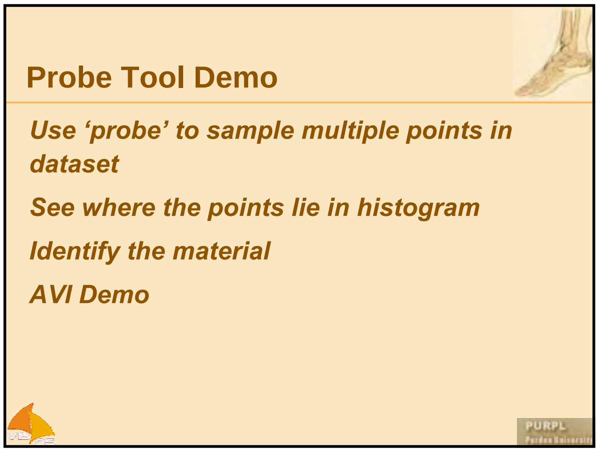


2D Histogram Analysis

These spots represent **homogeneous materials**

These arcs represent **material boundaries**

Histogram of CT Tooth Dataset



Probe Tool Demo

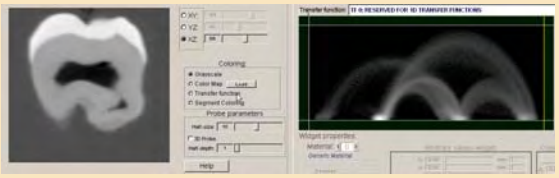
Use 'probe' to sample multiple points in dataset

See where the points lie in histogram

Identify the material

AVI Demo

Probing Demo



Why Is It Better?

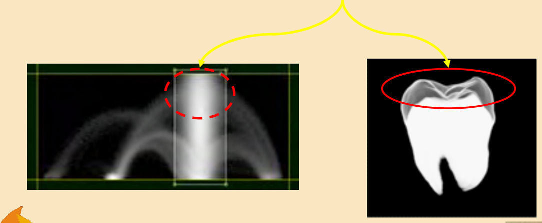
Easier to control levels of gradient magnitudes we want to see

Resolves ambiguities

- Parts of the dataset may be inseparable in 1D domain, but separable in 2D domain

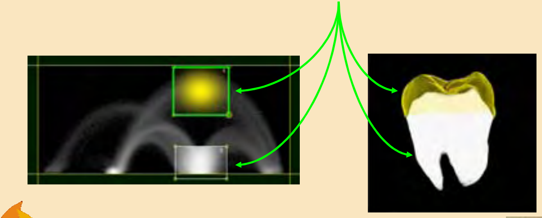
1-D vs 2-D Transfer Functions

Unrelated to what we want to see



1-D vs 2-D Transfer Functions

Different materials are separated



Hierarchical Transfer Functions

Usually transfer functions control (color, opacity)

They can also be used to control parameters/contribution of the effects

Design multiple transfer functions to control:

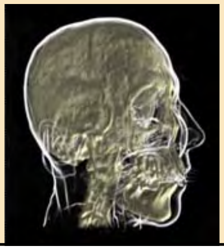
- Sketch
- Illumination
- etc.

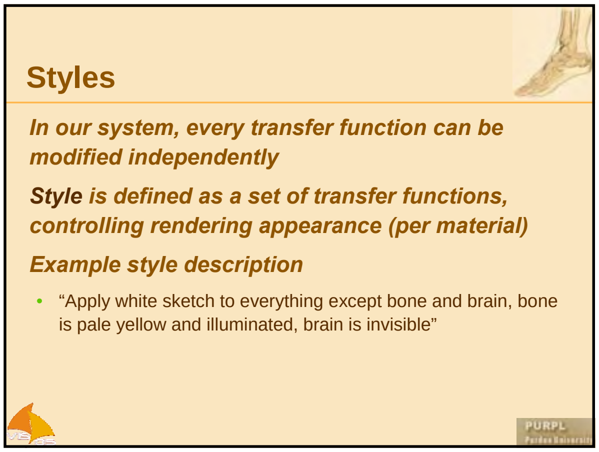
This allows applying different combinations of effects to different materials

Example

Sketch applied to 'skin' material

Illumination applied to 'bone' material





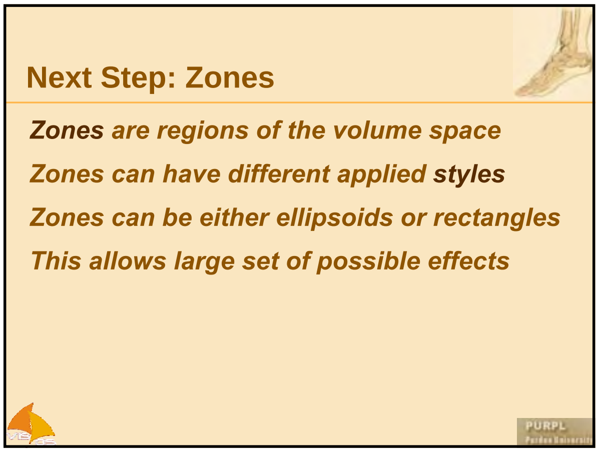
Styles

In our system, every transfer function can be modified independently

***Style** is defined as a set of transfer functions, controlling rendering appearance (per material)*

Example style description

- “Apply white sketch to everything except bone and brain, bone is pale yellow and illuminated, brain is invisible”



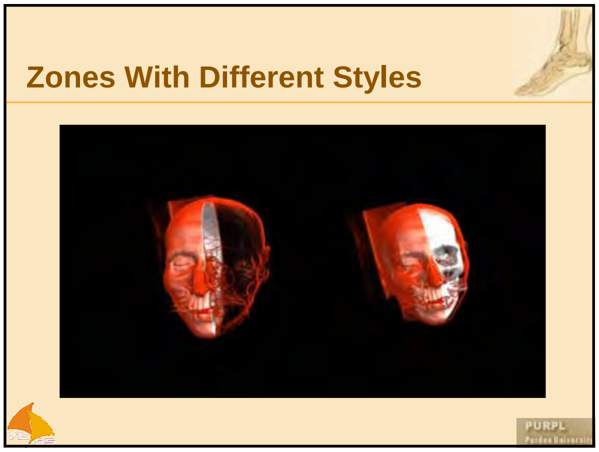
Next Step: Zones

***Zones** are regions of the volume space*

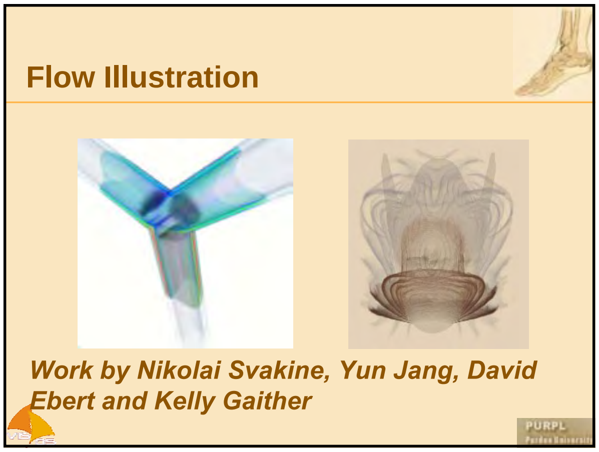
*Zones can have different applied **styles***

Zones can be either ellipsoids or rectangles

This allows large set of possible effects

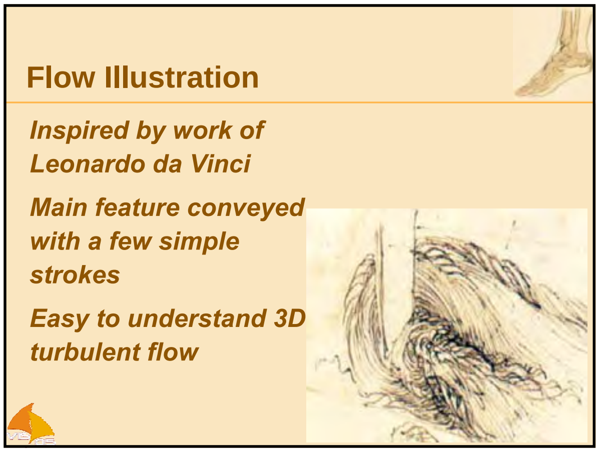


Zones With Different Styles



Flow Illustration

Work by Nikolai Svakine, Yun Jang, David Ebert and Kelly Gaither

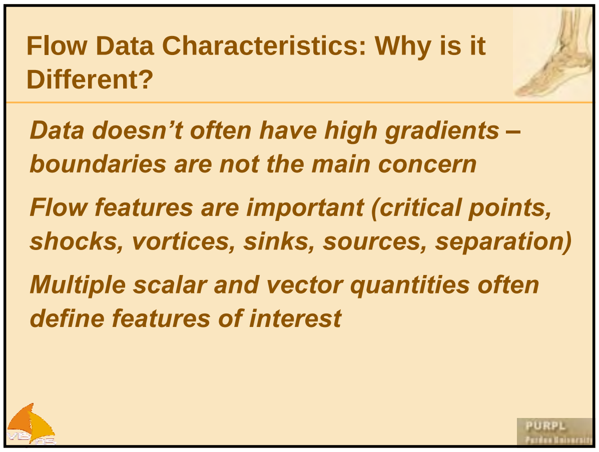


Flow Illustration

Inspired by work of Leonardo da Vinci

Main feature conveyed with a few simple strokes

Easy to understand 3D turbulent flow



Flow Data Characteristics: Why is it Different?

Data doesn’t often have high gradients – boundaries are not the main concern

Flow features are important (critical points, shocks, vortices, sinks, sources, separation)

Multiple scalar and vector quantities often define features of interest

Flow Illustration: What Works?

Silhouettes – projected 2D gradients remove visual clutter and detail and show major structures

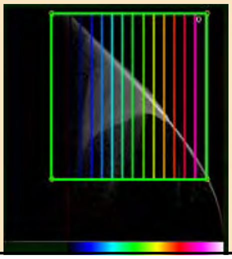
Flow Illustration: What Works?

Contouring transfer function – shows flow contours and structures

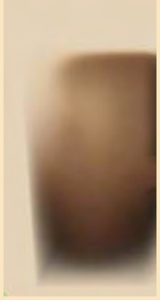
Flow Contours

Transfer function widget using sine function

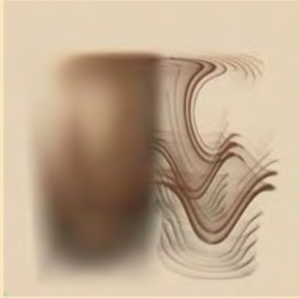
- Control frequency and amplitude



Flow Contours



No enhancement

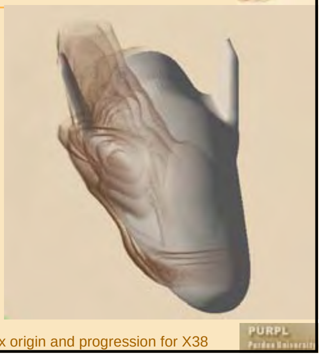


Left – boundary & silhouette
Right – add contours

Flow Illustration: Examples

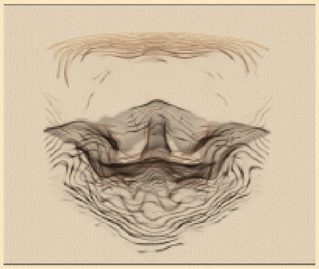
Found cause of vortices from illustration

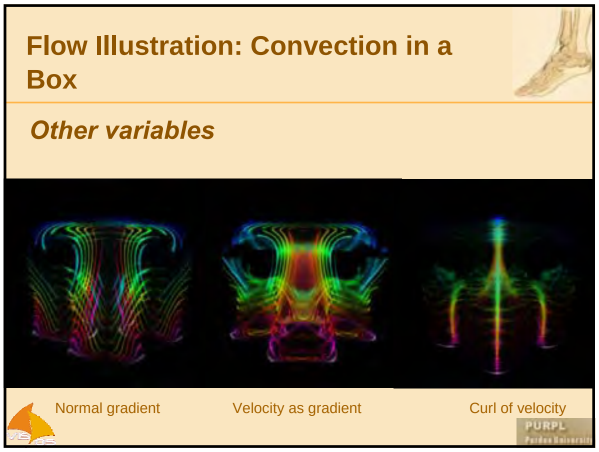
- Had always assumed wrong source!



Vortex origin and progression for X38

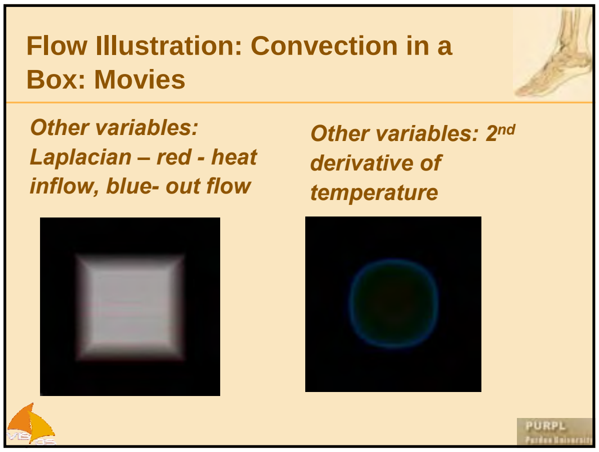
Flow Illustration: Convection in a Box





Flow Illustration: Convection in a Box

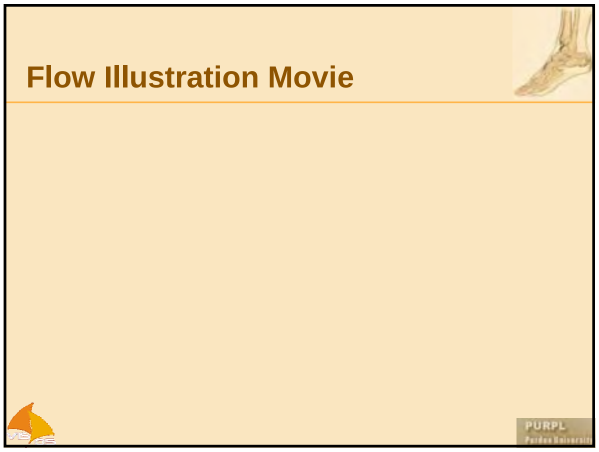
Other variables



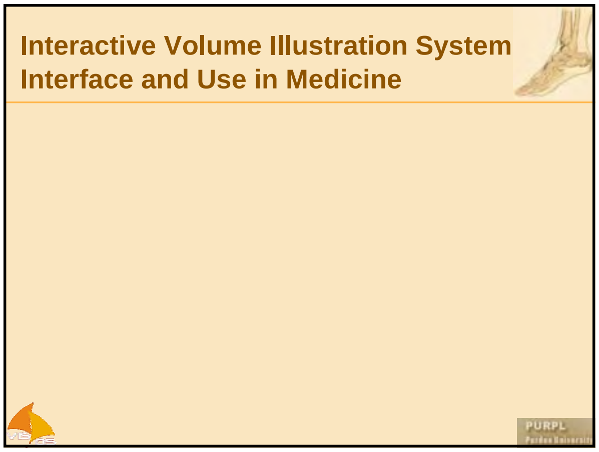
Flow Illustration: Convection in a Box: Movies

Other variables: Laplacian – red - heat inflow, blue- out flow

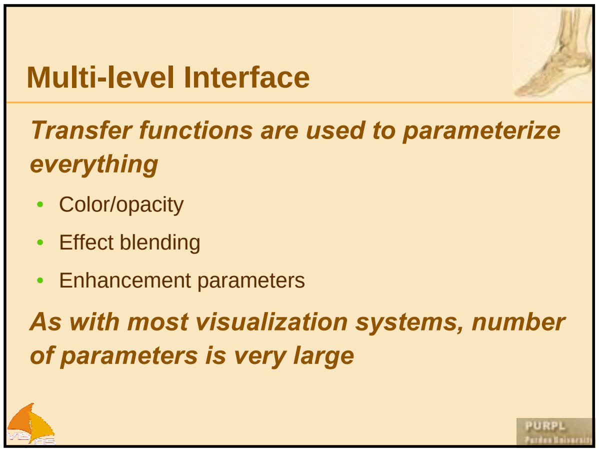
Other variables: 2nd derivative of temperature



Flow Illustration Movie



Interactive Volume Illustration System Interface and Use in Medicine

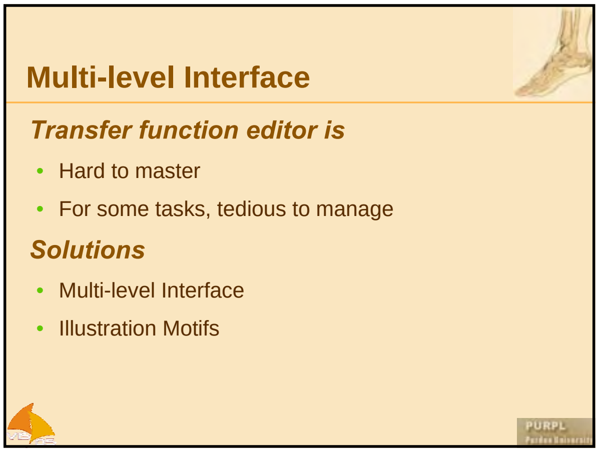


Multi-level Interface

Transfer functions are used to parameterize everything

- Color/opacity
- Effect blending
- Enhancement parameters

As with most visualization systems, number of parameters is very large



Multi-level Interface

Transfer function editor is

- Hard to master
- For some tasks, tedious to manage

Solutions

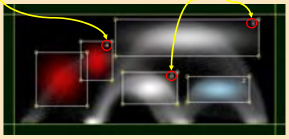
- Multi-level Interface
- Illustration Motifs

Defining Materials

Materials are regions of the transfer domain

Each widget has material ID associated with it

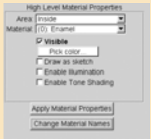
There can me more than one widget with same material ID



Higher Levels

Materials are assigned meaningful names

Settings formulated in common language terms



High Level Material Properties

Area: Inside

Material: (0): Enamel

☑ **Visible**

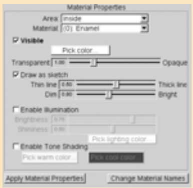
Pick color...

☐ Draw as sketch

☐ Enable Illumination

☐ Enable Tone Shading

High level



Material Properties

Area: inside

Material: (0): Enamel

☑ **Visible**

Pick color...

Transparent 1.00 Opaque

☑ Draw as sketch

Thin line 0.80 Thick line

Dim 0.80 Bright

☐ Enable Illumination

Brightness: 0.75

Shininess: 0.80

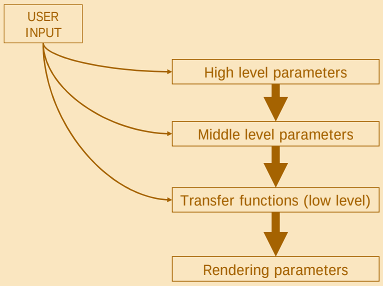
Pick lighting color...

☐ Enable Tone Shading

Pick warm color... Pick cool color...

Middle level

Parameter Pipeline



USER INPUT

High level parameters

Middle level parameters

Transfer functions (low level)

Rendering parameters

Medical Motifs

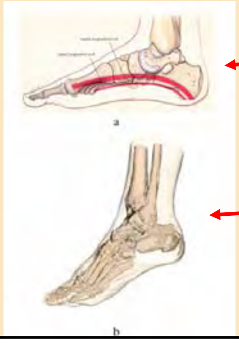
***Motifs** are settings upon which illustrator can quickly build styles specification*

Example:

- Anatomical illustration
- Surgical simulation
- Different levels of expertise for intended users

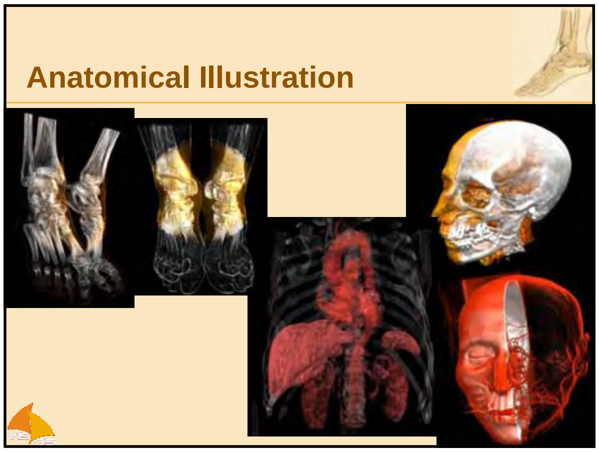
Comparison of Motifs

Medical illustrations



Foot bone structure from anatomy textbook

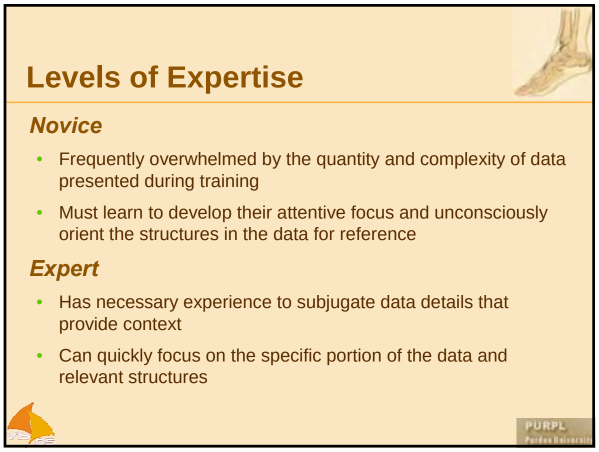
Same structure shown by IVIS with Visible Human foot dataset



Anatomical Illustration

Surgical Training

Images by Nik Svakhine



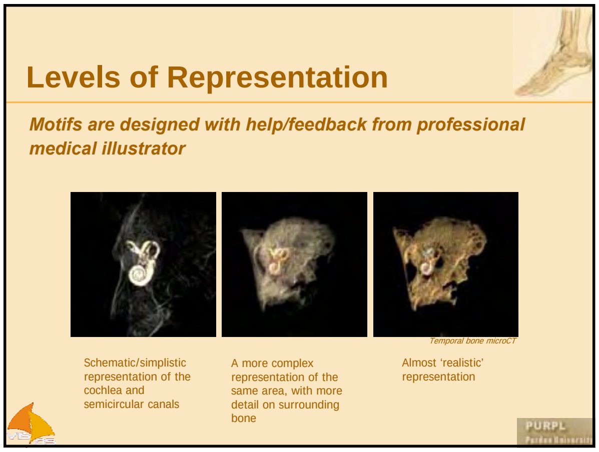
Levels of Expertise

Novice

- Frequently overwhelmed by the quantity and complexity of data presented during training
- Must learn to develop their attentive focus and unconsciously orient the structures in the data for reference

Expert

- Has necessary experience to subjugate data details that provide context
- Can quickly focus on the specific portion of the data and relevant structures



Levels of Representation

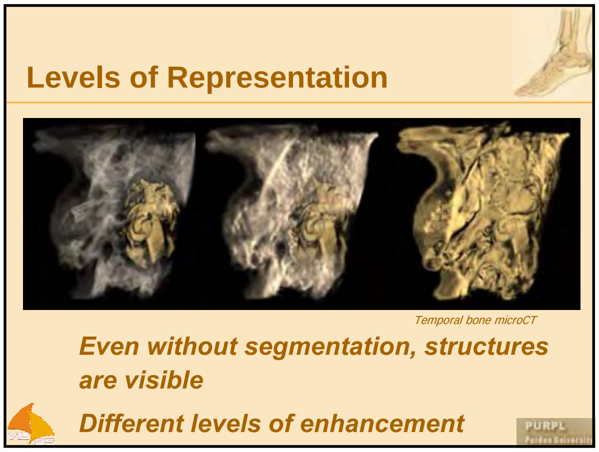
Motifs are designed with help/feedback from professional medical illustrator

Temporal bone microCT

Schematic/simplistic representation of the cochlea and semicircular canals

A more complex representation of the same area, with more detail on surrounding bone

Almost 'realistic' representation

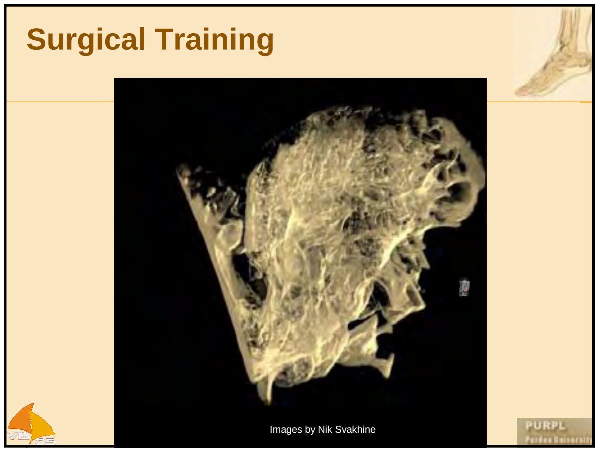


Levels of Representation

Temporal bone microCT

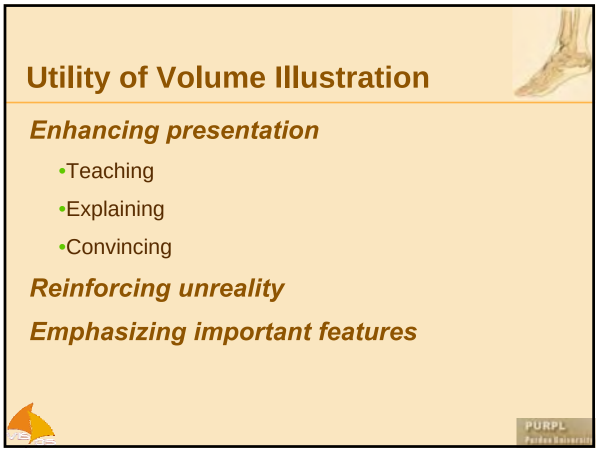
Even without segmentation, structures are visible

Different levels of enhancement



Surgical Training

Images by Nik Svakhine



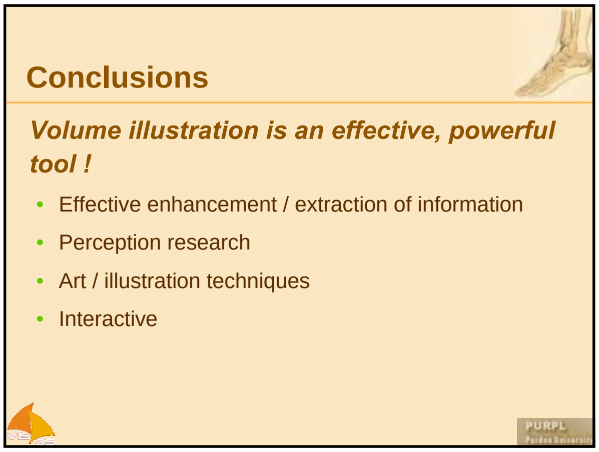
Utility of Volume Illustration

Enhancing presentation

- Teaching
- Explaining
- Convincing

Reinforcing unreality

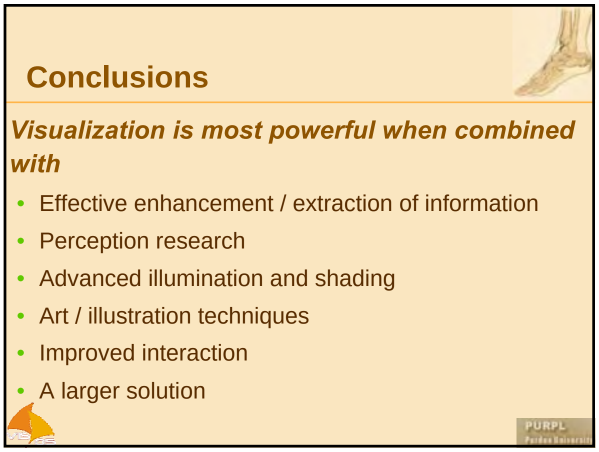
Emphasizing important features



Conclusions

Volume illustration is an effective, powerful tool !

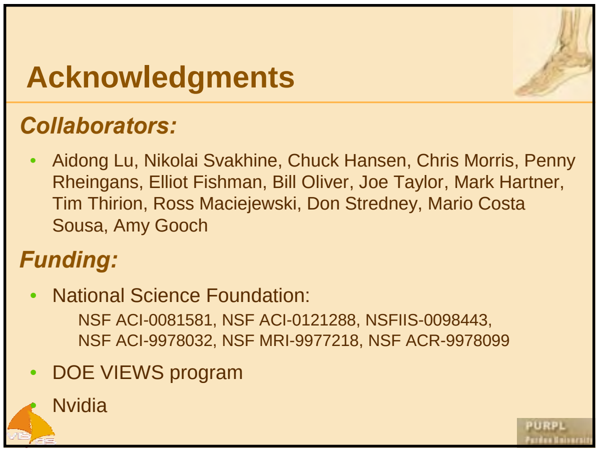
- Effective enhancement / extraction of information
- Perception research
- Art / illustration techniques
- Interactive



Conclusions

Visualization is most powerful when combined with

- Effective enhancement / extraction of information
- Perception research
- Advanced illumination and shading
- Art / illustration techniques
- Improved interaction
- A larger solution



Acknowledgments

Collaborators:

- Aidong Lu, Nikolai Svakhine, Chuck Hansen, Chris Morris, Penny Rheingans, Elliot Fishman, Bill Oliver, Joe Taylor, Mark Hartner, Tim Thirion, Ross Maciejewski, Don Stredney, Mario Costa Sousa, Amy Gooch

Funding:

- National Science Foundation:
  NSF ACI-0081581, NSF ACI-0121288, NSFIIS-0098443,
  NSF ACI-9978032, NSF MRI-9977218, NSF ACR-9978099
- DOE VIEWS program
- Nvidia

Illustration Motifs for Effective Medical Volume Illustration

Nikolai Svakhine[*]
Purdue University

David S. Ebert [†]
Purdue University

Don Stredney [‡]
Ohio Supercomputer Center

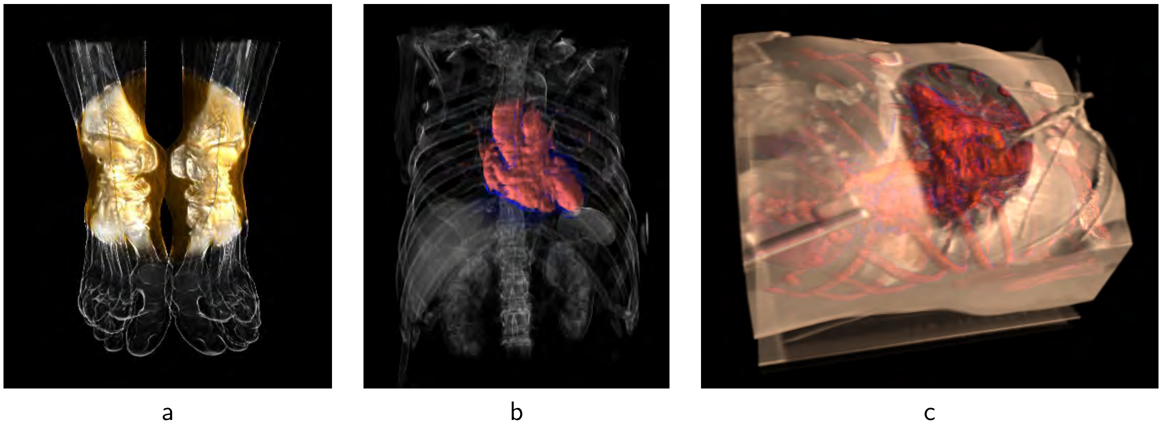


Figure 1: Medical illustrations of the (a) tarsal joints of the foot, (b) anterior view of the heart, and (c) surgical view of the heart.

Abstract

The main goal of visualization and rendering is to effectively convey information to the viewer. For scientific visualization, this goal translates into determining which portions of data to highlight and how to render these portions to concisely convey the appropriate information to the specific audience. These are the tasks that medical and technical illustrators have undertaken for centuries. We have developed a system that builds upon and extends recent work in volume illustration to produce images that simulate pictorial representations for scientific and biomedical visualizations. Our system is designed in collaboration with a trained biomedical illustrator whose work focuses on visualization for clinical research and resident surgical training. Our system combines traditional and novel volume illustration techniques. A high-level interface enables the user to specify the type of illustration and visualization goals to produce effective, illustrative rendering on commodity graphics hardware at nearly interactive rates.

CR Categories: I.3.8 [Computer Graphics]: Applications—; I.3.7 [Three-Dimensional Graphics and Realism]: shading—;

Keywords: volume rendering, volume illustration, illustrative styles, transfer functions, non-photorealistic rendering

1 Introduction

The enormous amount of three-dimensional data generated by modern scientific experiments, simulations, and scanners exacerbates the tasks of effectively exploring, analyzing, and communicating the essential information from these datasets. The expanding field of biomedicine creates datasets that challenge current techniques to effectively communicate information for use in diagnosis, staging, simulation, and training. In contrast, medical illustration succinctly represents essential anatomical structures in a clear way and is used extensively in the medical field for communicative and illustrative purposes [1]

Thus, the idea of rendering real medical datasets using traditional medical illustrative styles inspired work in volume illustration [6]. The main goal of the volume illustration approach is to enhance the expressiveness of volume rendering by highlighting important features within a volume while subjugating insignificant details, and rendering the result in a way that resembles an illustration. Recently, this approach has been extended to interactive volume illustration [11, 12] by using a commodity PC graphics hardware volume rendering implementation to accelerate the enhanced rendering, resulting in nearly interactive rates.

Our work goes beyond previous work in developing illustration techniques to create a flexible illustration system that incorporates domain knowledge of illustration styles. Our new volume rendering techniques enable emphasis and subjugation of information to create images with an appearance similar to medical illustrations, as shown in Figures 1 and 2. By selective application of enhancements, incorporation of rendering techniques to direct visual focus, and incorporation of several illustrative style profiles, our system can semi-automatically create visualizations of volume data in a variety of illustration motifs. In collaboration with a trained medical illustrator and medical visualization developer, we have designed a framework for use in various medical illustration tasks.

2 Previous Work

Our approach for volume illustration is based on standard volume rendering algorithms enhanced with non-photorealistic (NPR) techniques, as proposed by Ebert and Rheingans [6]. There are many other developed volumetric techniques that greatly aid the process of volume exploration and illustration. In particular, Höhne et al. [8] have developed a framework for generating volume based 3D interactive atlases from cross-sectional images, Weiskopf et al. [13]

[*] e-mail: svakhine@purdue.edu
[†] e-mail: ebertd@purdue.edu
[‡] e-mail: don@osc.edu

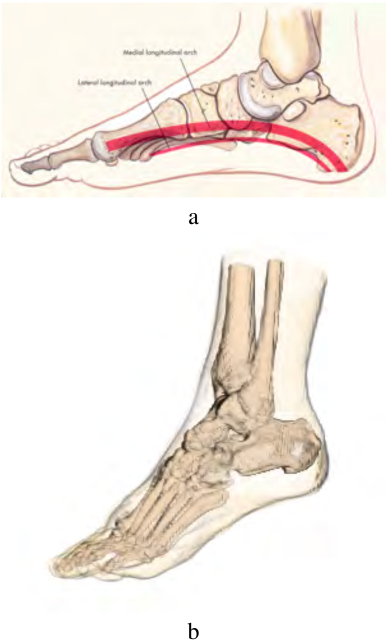


Figure 2: a) Foot bone structure from anatomy textbook [1], b) Same structure shown by our volume illustration system with visible human foot dataset.

present interactive clipping for texture-based volume rendering using arbitrary geometry, and Hadwiger et al. [7] describe two-level volume rendering with various enhancements for pre-segmented datasets.

The need to focus attention of the viewer on a particular part of the volume has been previously implemented through a user-controlled focal region used as an enhancement tool [2], where the enhanced iso-surface rendering algorithms are applied to the data contained in the sphere-bound volume of interest using high-resolution rendering inside, and coarse rendering outside.

We are employing an analogous interface to these focal area rendering techniques, achieving similar effects for interactive volume rendering. However, our system allows more flexible rendering motifs to be selected in the focal region and the outside volume.

Virtually all of this previous work describes parameterized rendering systems which focus mainly on the quality of the image, requiring the adjustment of many parameters. This usually makes these systems difficult for the end user to manipulate. Therefore, we have created a compact high-level interaction mechanism which provides different sets of basic settings and illustration motifs to create a more natural, controllable system for illustration generation. There are many areas in medicine where such illustrations are applicable, such as clinical research, resident surgical training, and general medical education.

3 Basis and Goals of Traditional Illustration

Most volume illustration work is directed at replicating traditional scientific illustration techniques. However, we want to look at the problem on a different level, considering the entire illustration process.

The fundamental goals of visualization are insight and communication; therefore, visualizations must allow a rich and varied repertoire for clear and concise depictions of complex phenomenon to aid in the task of sensemaking. The visualization challenge then becomes providing a diverse set of tools that allows for a wide variance of approaches to the information, one that not only allows one to “see” the information, but also facilitates “thinking” about the information.

All of these techniques include the variation, emphasis, and subordination of spatial information (pattern and form), light frequency (color), and temporal sequence (motion). Artists and illustrators manipulate line, shape, color, space and pattern in pictorial representation to depict and convey meaning and aesthetics. These cues help create the percept made by our neural mechanisms that process form, color, and movement [10]. Therefore, through visual cueing, not complete replication, the artist/illustrator manages the image so another brain can recreate a reality similar to the one conceived by the originator. The brain relies heavily on symbolic information to impart a deeper and more extensive meaning. Through emulation of previously adopted schema in our non-realistic volume rendering, we have generated preliminary data to study if these motifs can be effective in facilitating comprehension of large, complex three-dimensional datasets, and to explore how levels of representation can be employed to facilitate transitions from novice to expert in complex, interactive surgical training environments [4].

Studies in flight simulation have demonstrated that novices are overwhelmed by complexity, where experts will not attend to schematic or simplified representations. Similarly, surgical residents are often initially overwhelmed with information when they try to orient themselves during an actual surgery. Therefore, a medical illustration guide [3] gives direction for surgical illustration as follows: “A drawing of surgery removes inessentials. It eliminates distracting background and simplifies and emphasizes information.” The main value of our illustrative approach is the ability to initially select the appropriate level of complexity based on the target user’s expertise level for use in both medical education and surgical training. These selections will be employed in the studies to evaluate the efficacy of levels of representation for facilitating complex structural anatomy and procedural techniques for surgical residents.

4 Illustration System Overview

Our volume illustration system is a combination of existing volume visualization algorithms, new extensions to volume rendering techniques, a high-level, more natural user interface, and a domain-specific illustration specification framework. One of the main issues in the design of such a system is providing high-level control of the visualization parameters. As volume visualization systems incorporate more enhancement, rendering, and shading techniques, the number of user adjustable parameters tends to increase dramatically. In such a system, to create the most appropriate rendering of the dataset for a given application, the user must search a very high-dimensional parameter space, which makes the systems non-intuitive and extremely difficult to use. Therefore, through our interdisciplinary efforts, we have created a new, more natural specification interface with intuitive and understandable parameters.

Our system objective is to provide an intuitive interface for users with various backgrounds to be able to quickly generate the required results and to keep the performance acceptable for the application. The results are images and animations used for educational, training, and simulation purposes. The interface incorporates three levels of interaction: a low-level, a mid-level, and a high-level. The low-level is for software developers, experienced illustrators, and system builders. The mid-level interface is for illustrators and experienced end-users that want to make adjustments once they understand the controls. The high-level interface is designed for the end user, and in our current application focus, it would be for medical students, surgeons, and surgical residents.

As mentioned previously, we have designed our illustration system on schema found in perception and subsequently exploited

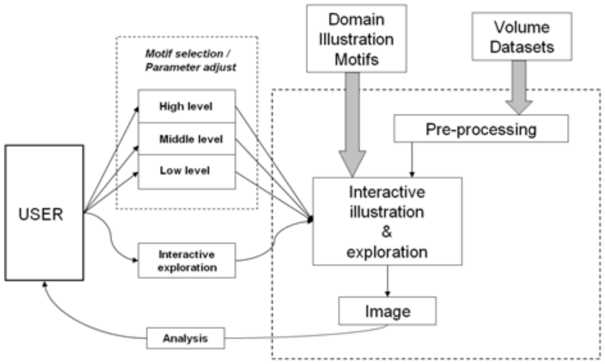


Figure 3: Interactive volume illustration system

through illustrative techniques. One of the primary perceptual principles used by illustrators is visual focus: the most important information in the illustration will be rendered to facilitate orientation and to hold the viewer's attention. The surrounding details are deemphasized (subjugated) to provide context, while not overloading the viewer's attention. Therefore, a simple focal volumetric region is utilized to determine the spatial distribution of illustrative techniques to be applied.

Our spatial focusing method is designed to reflect the attentive focus that results from foveal viewing, similar to the Magic-Sphere [5]. Our method attenuates on the edge similar to the acuity of foveal vision and is meant to direct and focus the attention of the viewer. This technique is often used in traditional medical illustration. Other approaches include using anatomical objects as the basis of focus/context, but they also require segmented datasets. To make our system most generally applicable, we provide spatial focusing. However, segmentation information can be readily used in our system if it is available.

We also classify illustrative techniques in terms of their use for color perception and form perception. We utilize variances to emphasize or de-emphasize form, using a repertoire of volume rendering enhancements, such as stipples, lines, continuous shading, as well as illumination and contrast. Finally, interactivity is crucial to utilize our shape-from-motion perception and to improve our understanding of these complex three-dimensional datasets. Subsequently, we ensure the interactive performance of our system by adjusting the rendering quality to meet the interactivity requirements.

High-level control of the visualization process is achieved through the use of *illustration motifs*, the sets of rendering parameters that define an illustrative approach. Since we are incorporating traditional medical illustration motifs, we can capitalize on the familiarity of these motifs and presentation methods with medical professionals. The motif specification contains a compact set of core parameters and specification of the target application, providing the user with a much more manageable set of understandable controls.

An overview of the components of, and user interaction with, the system is provided in Figure 3. While our initial system includes motifs specific to medical illustration and several medical visualization applications, the motif specification interface allows additional domain motifs and target applications to be easily incorporated. The user interaction with the system through our new interface is described below in Section 4.2.

4.1 Medical Motifs

Artists employ various techniques to manipulate or guide the viewer's attentive focus. These techniques, based on physiological principles, include brightness and complexity, with complexity being composed of increased variations of feature detail (pattern/texture, color, and position). In order for the end user to perceive the detailed picture of a specific portion of the volume, an artist will apply high-quality, structure-emphasizing techniques to this area, while providing a sketch of the outer area. The high-quality artistic rendering allows the user to perceive all the necessary details (conscious attention), while the outer sketch provides context. This context is perceived unconsciously, but still compellingly represents the outer environment for reference.

We, therefore, introduce different styles for the region of interest renderings and the outer context renderings. In order to begin the process, an illustrator first must select the regions of the volume to be emphasized or de-emphasized. The general volumetric selection process is a complicated user-interface problem, so we do not use volume marking directly. Instead, we employ known methods traditionally used in volume rendering: spatial focusing, transfer function selection, and segmentation.

Our spatial focusing method is implemented with a simple ellipsoidal/rectangular focal area for general datasets as shown in Figure 7, and allows segment-based focusing for segmented datasets. This focusing is useful for specification of the region of interest and the outer context for applying different rendering styles.

Transfer function selection is based on the idea that separate materials in the volume correspond to different value ranges in volume data value, and in its first and second derivatives [9]. While this may or may not be true for all problems, transfer function selection has proven useful for material-based classification of the volume. To define the material, one can simply select the appropriate ranges in the transfer function domain.

The segmentation, when available, essentially provides the process with an extra volumetric map, which identifies a material/part to which each voxel belongs. Creating a meaningful segmentation for the volume has to be done as a preprocessing stage using a variety of techniques.

Once the selection stage is complete, the user needs a simple and understandable way to communicate illustration parameters to the rendering system, while only needing to know the general requirements of the target application. In our system, the motifs can be used to describe a wide variety of effects, including the ability to highlight a selection to emphasize and focus the viewer's attention, remove layers of occluding material while exposing the important objects, and provide a chalk-sketch of the outer shape of the body part in order to visualize only the contours, limiting visual confusion. These capabilities are exposed through the multi-level user interface described in Section 4.2.

As has been mentioned previously, a motif can be viewed as the settings template that is used as a basis for a particular illustration style. In our current system, motifs are characterized by (1) the meaning of the zone of interest and (2) styles, defined inside and outside of the zone. The power of these motifs is to exploit a single data base for a wide variety of rich and complex representations.

A simple example is the *anatomical illustration* motif. The zone of this motif has a meaning of "attentive focus." Inside this zone, the settings are set to render the full detail of the illustrated feature (using Phong illumination, boundary enhancement, etc.), while outside features that define general shape are drawn as a sketch (using a black or white color and boundary and silhouette enhancement). An example of using this motif is shown in Figure 4a.

For another example set of motifs, we describe two motifs used for surgery simulation. Our approach uses the trainee's level of expertise as a basis for motif classification. As previously mentioned, novices are frequently overwhelmed by the quantity and complexity of data presented during training and must learn to develop their attentive focus for the surgical target and unconsciously orient the contextual structures in the data for reference. Therefore, in

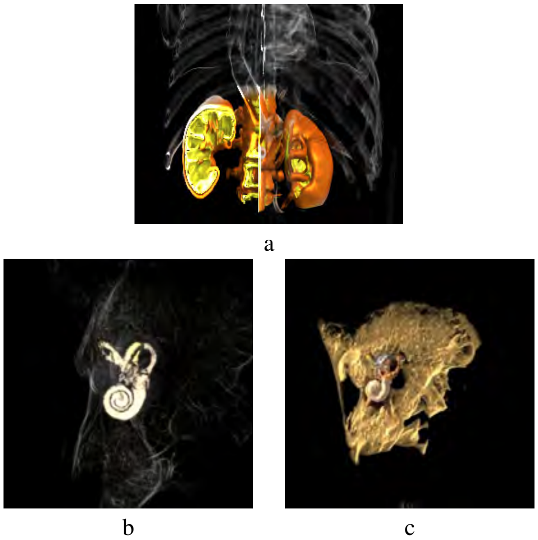
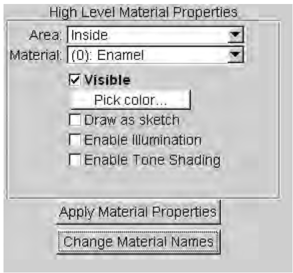


Figure 4: Motif examples: a) anatomical illustration motif for kidney illustration, b) and c) novice and expert motifs for surgery simulation in the otic capsule of the inner ear


High Level Material Properties

Area: Inside

Material: (0): Enamel

☑ **Visible**

Pick color...

☐ Draw as sketch

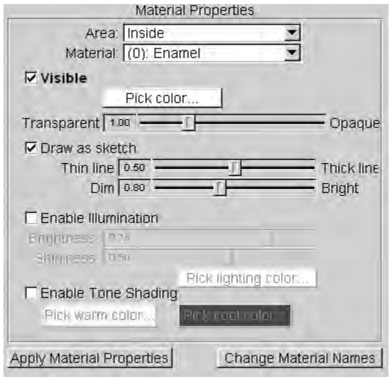
☐ Enable Illumination

☐ Enable Tone Shading

Apply Material Properties

Change Material Names

Figure 5: High-level control panel


Material Properties

Area: Inside

Material: (0): Enamel

☑ **Visible**

Pick color...

Transparent 1.00 Opaque

☑ Draw as sketch

Thin line 0.50 Thick line

Dim 0.80 Bright

☐ Enable Illumination

Brightness 0.75

Shininess 0.50

Pick lighting color...

☐ Enable Tone Shading

Pick warm color...

Pick cool color...

Apply Material Properties

Change Material Names

Figure 6: Mid-level control panel

a *novice surgery simulation* motif, a zone of interest is placed in the surgery target area. To define styles, we note that there needs to be a silhouetted context for subjugation of details outside of this zone, while an illustrative rendering with color cues is used inside the zone to enhance the important structures (Figure 4b). In contrast, an expert has the necessary experience to subjugate the data details that provide context and can quickly focus on the specific portion of the data and relevant structures. Thus, in our *expert surgery simulation* motif, we provide more detailed rendering with the selective choice of color and illumination enhancements for structure identification (Figure 4c). However, even if an expert is introduced to new information (outside of their own expertise), they may need to start at either the schematic or intermediate level to structure their mental representation. Therefore, we also need the ability to switch from novice to intermediate to expert levels of representation.

4.2 Application Interface

Our volume illustration system provides three levels of user interaction for different classes of users. The three levels balance the level of control with the flexibility and ease-of-use of the system.

The high-level interface is for novice users or end-application users who wish to interact quickly with the system and produce images from the included high-level motifs, (e.g., a surgical resident or attending physician to create images for patients or general surgical education). The user is able to specify the type of illustration motif (e.g., anatomical illustration) and select different materials within the data to be emphasized. The high-level interface is shown in Figure 5.

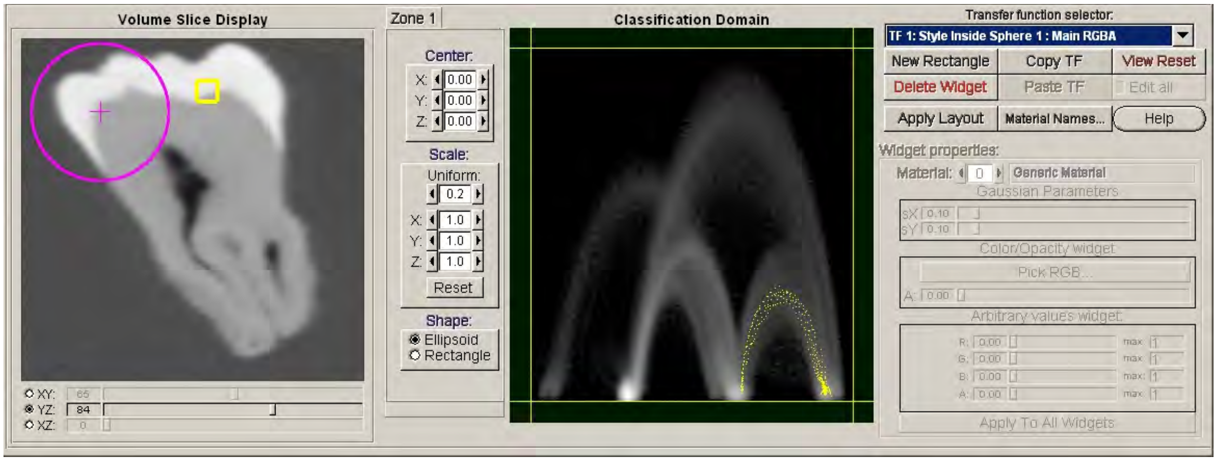
The mid-level interface, shown in Figure 6, allows more experienced users (e.g., medical illustrators) to create and fine-tune the high-level motifs, and make general adjustments to the application of illustration techniques. This level of interface is designed to be the most frequently used, as it allows fine adjustment of all rendering parameters at a manageable level. Using simple spinner controls with intuitive parameter names, the user is able to specify the approximate magnitude of various effects based on the position of the voxel (in different areas or for different volume segments) and on the voxel data values.

Finally, the low-level interface allows expert users to navigate the data thoroughly, identify new materials, name them for easy identification in the high- and mid- levels, adjust existing material properties and specification, and modify all the parameters of the transfer functions. The main part of this level is the 2D transfer function editor, similar to the one described in [9]. It contains the map of the classification domain, with 2D histogram displayed as the background for reference, and allows user to select regions of the domain by creating widgets and assigning parameters to them.

Another part of the interface is the cross-section display window, which allows the user to see grayscale image of an arbitrary cross-section of the volume, and is accessible from all levels of the interface. This window can be used for two purposes. First, the user can use this view (on any level) to specify the position of the region of interest by placing its center point on the cross-section. Second, when using the low-level interface, the user can "probe" the volume to identify materials in the volume: after clicking on the window, the voxels from the small neighborhood of the corresponding point are projected on the 2D classification domain and are displayed as points in the transfer function editor window. Figure 7 shows the cross-section display window along with the low-level control panel, containing the transfer function editor.

After the low-level is used to identify the materials in the classification domain, the user can utilize the mid-level interface or the high-level interface. When the user uses higher levels and the settings are changed, the transfer functions are re-generated. However, they still can be edited in the transfer function editor and, thus, the low-level interface can also be used to fine-tune the result and individual parameters of each illustration technique.

To facilitate the parameterization of the rendering, "rendering profiles" can be created based on illustration motifs, target application of the final image, and specific datasets. These profiles contain user settings on all levels, including high-level flags, mid-level parameters and low-level specifications of transfer functions. Profiles can be created based on previous illustration work with a particular dataset, or datasets produced using the same acquisition technique



Transfer function selector: TF 1: Style Inside Sphere 1 : Main RGBA

Center: X: 0.00 Y: 0.00 Z: 0.00

Scale: Uniform: 0.2 X: 1.0 Y: 1.0 Z: 1.0

Shape: ● Ellipsoid ○ Rectangle

○ XY: 65
● YZ: 84
○ XZ: 0

Material: 0 Generic Material

sX 0.10
sY 0.10

A: 0.00

R: 0.00 max 1
G: 0.00 max 1
B: 0.00 max: 1
A: 0.00 max 1

Figure 7: Left side: cross-section window with the zone of interest (purple circle) and a probe (yellow square). Right side: low-level interface with transfer function editor with map of the classification domain and widget controls. Classification domain map shows probing result (yellow dots).

(e.g., normal CT scan, contrast enhanced CT scan, T1-weighted MRI scan). Of course, knowledge of patient-to-patient variance and domain expertise is needed to adjust the profiles to provide the most optimized data representation.

5 Rendering Overview

5.1 Renderer

We use texture-based volume rendering in combination with programmable fragment programs for our volume illustration system. Before rendering, the volume data is converted into two 3D textures, which are used in fragment shaders as sources for voxel value and voxel gradient (the gradient volume is pre-computed). The textures are standard OpenGL 1.4 RGBA textures with 8-bits per component. All the computations inside the fragment program use 32-bit floating point precision, and for higher blending precision, we use 32-bit floating point buffers (e.g., the NVIDIA OpenGL extension, available on FX boards).

For volume rendering, the volume is sliced by view-aligned quadrilaterals that are rendered in back-to-front order and blended together to form the final image. As each slice is being rendered, the application level program clips the slice against the volume boundary, while shading occurs at the fragment level. Thus, to change the motif parameters, we only need to change the input of the fragment program, without re-processing the data. The illustration techniques are easily adapted to modern graphics hardware, where enhancement calculations occur on the fragment level.

Our basic set of illustrative enhancements include the following feature and orientation enhancements: boundary enhancement, sketch enhancement, color distance attenuation, aerial perspective, null halos, Phong illumination, and tone shading. We have developed an interactive implementation of these techniques based on recent work described in [12].

All of the transfer functions are stored as 2D-lookup tables consecutively packed into a 3D texture. In the fragment shader program, when the value needs to be calculated, a 3D lookup is used, with the transfer function arguments as the first and second coordinates and transfer function ID as the third coordinate.

5.2 Multi-level Transfer Functions

To create a system that has the power of scientific illustration, we need to be able to render different objects, materials, and regions within the dataset using different illustration techniques. Traditional transfer functions do not have the flexibility and capabilities needed. Therefore, we have developed a collection of hierarchical transfer functions that we will refer to as multi-level transfer functions.

One of the basic components of the illustration process is to first identify the different materials within the dataset. In volume rendering, we can use multiple scalar values per voxel for material classification. The basic scalar value is the voxel value (X-ray absorption for CT datasets, hydrogen density for MRI datasets). Other commonly used scalar values are the first and second derivative magnitudes [9]. The Cartesian product of the domains of these chosen local voxel values forms the *classification domain* and the number of considered values determines the dimensionality of the classification domain. Currently we use a 2D-domain of (Voxel Density, Gradient Magnitude). With segmented datasets, we use the segmentation data to separate the materials, instead of the transfer function on the classification domain.

Traditionally, transfer functions are used only to map classification domain values to the color and opacity of samples. Thus, changing the transfer function specification highlights some materials, hides others, and assigns different colors to different materials. While these capabilities provide good cues for initial volume exploration, they are not sufficient for effective illustration. We call the traditional transfer function the *basic transfer function*. Other transfer functions in our multi-level set are classified below based on their purpose (illustration goal) and classification domain (voxel value).

We need the ability to render different materials in different styles. Since style is defined as a set of volumetric enhancement effects applied to a sample, we use a set of transfer functions that map classification domain values to the magnitude of a particular effect and, thus, makes it possible to distinguish the materials not just by opacity and color, but also by rendering technique. These functions form a group of *selective enhancement transfer functions* (e.g., Figure 8a).

In order to direct the visual focus of the viewer, the rendering algorithm must apply different styles based on the voxel position (e.g., whether or not the voxel is inside the region of interest). In order to do this, we use a *style selection function*. Here, a style selection function is a condition that determines which style to apply to a voxel, based on its position. If the current voxel is in the boundary zone (e.g., close to the boundary of the region of interest),

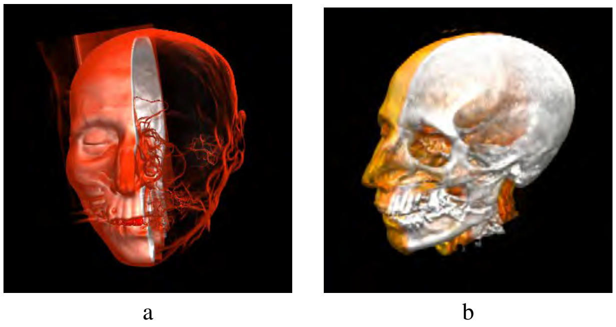


Figure 8: a) CT head with silhouette effect on tissue, b) CT head with clipped tissue

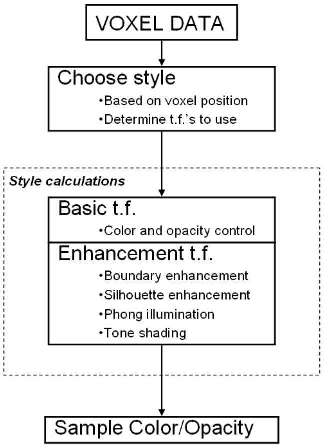


Figure 9: Illustrative calculations on fragment level

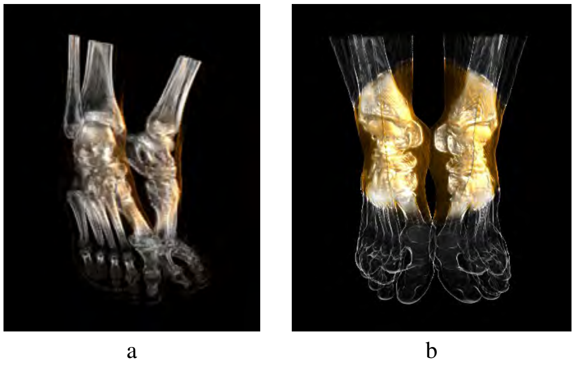


Figure 10: Visible Woman feet dataset illustrations, highlighting tarsal (ankle) joints

appropriate interpolation of the two styles is calculated and used as the result.

The data flow inside fragment program is illustrated in Figure 9. The style selection chooses the set of selective enhancement transfer functions, and those functions are used to determine sample color and opacity. By using this approach that can apply different styles to the different zones, we can render the image with a combination of various effects and illustration techniques and, thus, generate images similar to the various styles of medical and technical illustration.

6 Initial Results

In order to demonstrate the illustration capabilities of our system, we have used four representative datasets for illustration: the Visible Woman feet CT[1] dataset, a contrast enhanced abdominal CT[2] dataset, a temporal bone microCT[3] dataset, and a virtual colonoscopy[4] dataset. The selection of example regions of interest has been done in collaboration with medical experts, who provided insight into the significance of various anatomical structures. All the images were generated by loading the appropriate motif and adjusting rendering parameters on all of the three levels of the interface.

The anatomical illustration motif was designed to mimic the traditional medical illustration found in medical anatomy texts [1]. As previously mentioned in the description of the motif, the main idea is to draw the focal region in full color, while subjugating the outer tissue through sketching for reference. Surgical simulation, on the contrary, has the goal of providing realism by showing an image similar to the appearance of the patient on the operating table, with different levels of subjugation of the unneeded details. The novice surgical illustration motif level shows only the relevant anatomical structures, while the expert level shows all of the patient data with minimal, selective enhancement to increase the realism of the simulation.

Two anatomical illustrations of the Visible Woman feet CT dataset are presented in Figure 10. These images are based on the anatomical illustration motif, and employ an illuminated realistic volume rendering inside the zone of interest and varying levels of silhouette/boundary enhancement to “sketch” the surrounding structures. The region of interest is placed to emphasize and focus attention to the subject of the tarsal (ankle) bones of the foot. The silhouette technique (stronger in Figure 9b) is employed to subjugate the outlines of the surrounding bones and surface of the feet.

Figures 11a-d are also generated with our anatomical illustration motif. Here, we use it to focus attention and illustrate the chambers of the heart (a), kidneys with a cutaway view for detailed structure (b) and major components of the circulatory system plus the liver, spleen and kidneys (c). The surrounding structures are deemphasized, but serve to orient the user and provide location. Figures 11a-c once again use sketching of the outer context and more realistic illuminated volume rendering for organs and systems of interest. Figure 11d shows a false-colored, simplified illustration of the gross anatomy of the chest and abdominal cavity created from this same dataset, with the digestive organs removed. The relationship of the lungs, spine, and aorta are clearly shown in the middle top of the image, while the liver (green), spleen (blue) and kidney (blue/red) are highlighted in the lower half of the image. In contrast to these anatomical illustrations, the illustration in Figure 11e shows the same dataset rendered using an intermediate surgical illustration motif, where the structure of the heart chambers are drawn in reference to the body and occluding tissue is removed. A portion of the left ventricle has been removed and the heart is more transparent than in reality to create an illustration for an annular ring or valve replacement surgery.

Figure 11f shows an anterior view of the colon, where a part of the transverse colon is highlighted in red and a portion of it is cut away with three extra clipping planes to show internal struc-

[1] Visible Human Project

[2] Johns Hopkins University

[3] Kim Powell, Cleveland Clinic Foundation

[4] Dirk Bartz, University of Tubingen

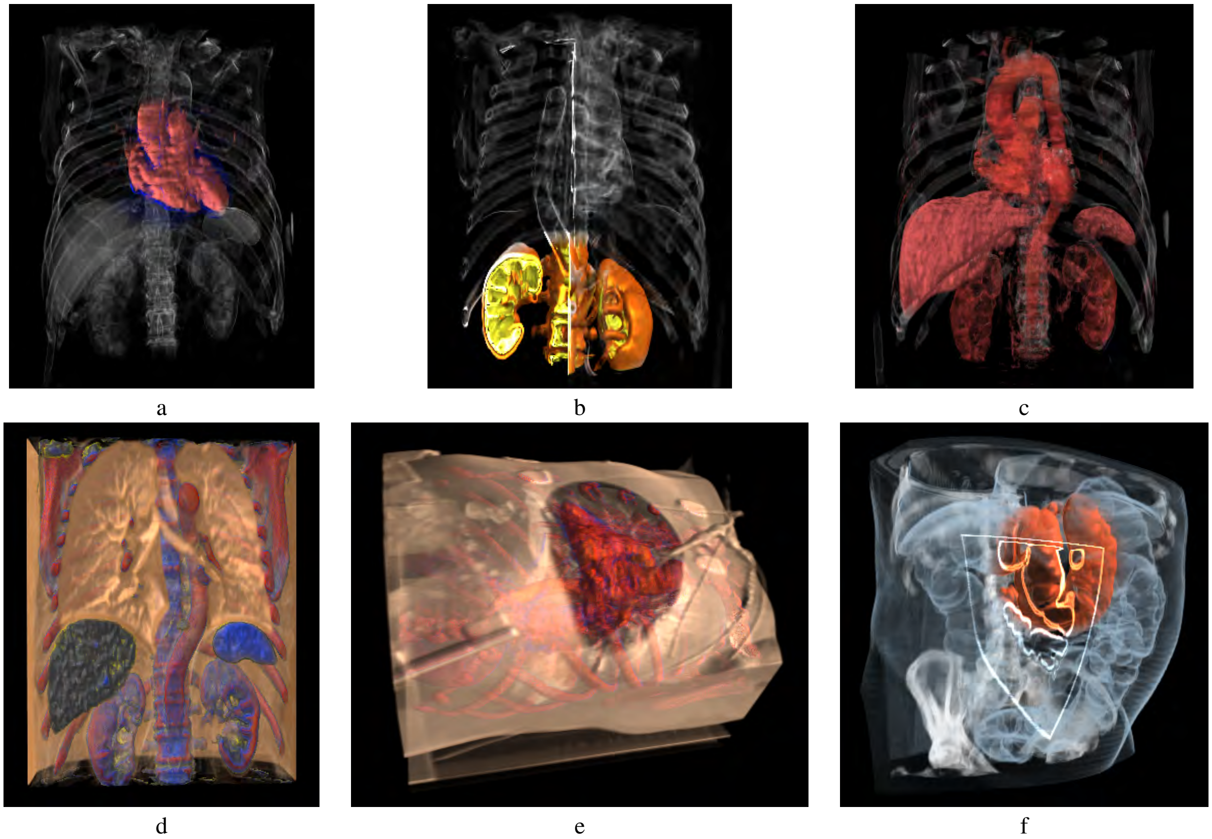


Figure 11: Abdomen CT dataset illustrations a) chambers of the heart b) kidney structure c) circulatory system, liver, spleen and kidneys d) false-colored chest and abdomen cavity illustration e) chambers of the heart, surgical illustration f) colon illustration with bent region highlighted and cut with clipping planes

ture. The motif used there has two zones. The first zone simply highlights the volume region of interest by using a red color, and is placed on the bent region of the colon. The second zone is a rectangular region and is used to cut away the front portion of the body (inside this region, everything has zero opacity), and enhanced contour lines along the cut to show the intestine detail. This image was generated using an augmented shader for contour enhancement on the edge of the zone.

Figures 11a through d use an objective view, common in anatomical illustration, while Figure 11e uses the surgeon's subjective view, as used in surgical illustrations.

Figure 12 contains several renderings of a micro CT-scan of a temporal bone. These images emphasize important parts of the bone and the cochlea structure, which is occluded by the surrounding structure and is, therefore, difficult to visualize with traditional methods. Figure 12a shows a novice surgical simulation view where the cochlea is shown in detail while the rest of the bone is simplified to aid navigation and orientation. Figure 12b shows more detail of the entire structure, the cochlea part is enhanced but does not stand out as much, while Figure 12c shows a non-enhanced rendering of the dataset, much closer to a real surgical view.

All of the images were generated on a Pentium IV 1.5 Ghz PC with 1.5 Gbytes of RAM and a GeForce FX 6800 Ultra card (128 Mbytes of VRAM), using 500 slices, and a screen area of approximately 400x400. For these settings, current performance is about 4 frames per second. However, in the "preview mode" we use one simplified transfer function to color the volume, which speeds the rendering to 20 fps for the same settings.

The majority of the enhancement calculations occur at the fragment level, so the more effects we include, the more texture lookups are performed in a single pass, and the longer the fragment program becomes. The performance also directly depends on the screen size and the number of slices.

7 Conclusions

We have presented a new, effective, volume illustration system based on illustration motifs. Our system incorporates many feature enhancement techniques and new multi-level transfer functions, provides interactive exploration and analysis of volume datasets, allows adjustment of illustrative parameters based on classification and spatial data, and creates images structurally similar to medical illustrations. One significant contribution is the enabling of different illustration techniques applied to different regions or materials in a manner that incorporates medical illustration expertise and application goals.

This new system is a powerful exploration, previewing, analysis, and illustration tool. In order to facilitate the usage of the system, we employ a multi-level user interface, which allows both flexibility and ease of final illustration generation. Collaboration with medical illustrators and surgical simulation training developers allowed us to provide appropriate interfaces at multiple levels.

We have received encouraging feedback that our system will be useful for providing training modules for medical education and also be very useful in surgery simulation for training surgeons, such as one of the applications that our biomedical illustrator co-author

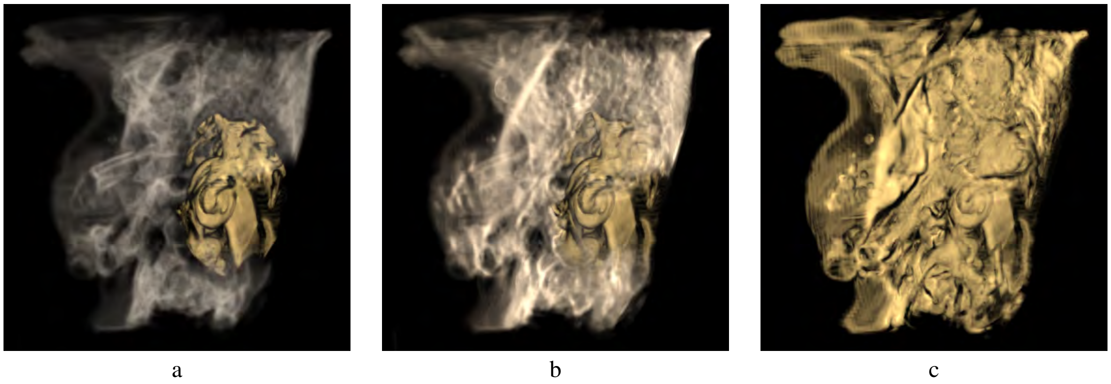


Figure 12: Temporal bone illustrations. a) cochlea structure enhanced b) cochlea structure slightly enhanced, with more realistic appearance c) non-enhanced realistic rendering

has been developing.

8 Future Work

In continued collaboration with field experts in medical visualization, we will include more volume rendering enhancements in the toolkit, which will extend the current illustration approaches to provide more flexibility for various applications. We also plan to extend this work to real-time interactive training systems. For instance, one of the authors (Stredney) is involved in a study that is developing and validating an interactive temporal bone dissection simulator.This system emulates temporal bone dissection, which is used to gain proficiency in temporal bone surgical technique. We believe it could prove useful to dynamically adjust the region of interest and level of detail based on the level of expertise of the user. This would allow for increased usage of the system for both novice training, by employing schema to control emphasis and subjugated areas, while also providing the expected complexity and sophistication that would be required by experts for use in pre-operative assessment and treatment planning.

9 Acknowledgements

The authors would like to thank nVidia for their support in providing the prototype GeForce FX 6800. This paper is based upon work supported by the National Science Foundation (Grant Nos. 0222675, 0081581, 0121288, 0196351, and 0328984) and by a grant from the National Institute on Deafness and Other Communication Disorders, of the National Institutes of Health, 1 R01 DC06458-01A1.

References

[1] Catherine Parker Anthony and Gary A. Thibodeau. *Textbook of Anatomy and Physiology*. Times Mirror/Mosby College Publishing, St. Louis, 1987.

[2] Eric A. Bier, Maureen C. Stone, Ken Pier, Ken Fishkin, Thomas Baudel, Matt Conway, William Buxton, and Tony DeRose. Toolglass and magic lenses: the see-through interface. In *Conference companion on Human factors in computing systems*, pages 445–446. ACM Press, 1994.

[3] Mary Helen Briscoe. *Preparing scientific illustrations: a guide to better posters, presentations, and publications*. Springer, New York, 1996.

[4] J. Bryan, D. Sessanna, D. Stredney, and G.J. Wiet. Virtual temporal bone dissection: A case study. In *Proceedings of the conference on Visualization '01*, pages 497–500. IEEE Computer Society Press, 2001.

[5] Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Magicsphere: an insight tool for 3d data visualization. *Computer Graphics Forum*, 13(3):317–328, 1994.

[6] David Ebert and Penny Rheingans. Volume illustration: nonphotorealistic rendering of volume models. In *Proceedings of the conference on Visualization '00*, pages 195–202. IEEE Computer Society Press, 2000.

[7] Markus Hadwiger, Christoph Berger, and Helwig Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proceedings of the conference on Visualization '03*, pages 301–308. IEEE Computer Society Press, 2003.

[8] K. H. Höhne, M. Bomans, M. Riemer, R. Schubert, U. Tiede, and W. Lierse. A 3d anatomical atlas based on a volume model. *IEEE Computer Graphics and Applications*, 12(4):72–78, 1992.

[9] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings Visualization 2001*, pages 255–262, October 2001.

[10] M.S. Livingston. Art, illusion, and the visual system. *Scientific American*, 258:78–85, 1988.

[11] Eric B. Lum and Kwan-Liu Ma. Hardware-accelerated parallel nonphotorealistic volume rendering. In *Proceedings of the second international symposium on Non-photorealistic animation and rendering*, pages 67–ff. ACM Press, 2002.

[12] N. Svakhine and D. Ebert. Interactive volume illustration and feature halos. *Pacific Graphics '03 Proceedings*, 15(3):67–76, 2003.

[13] Daniel Weiskopf, Klaus Engel, and Thomas Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, jul-sep 2003.

Illustration and Photography Inspired Visualization of Flows and Volumes

Nikolai A. Svakhine[*]
Purdue University

Yun Jang[*]
Purdue University

David Ebert[*]
Purdue University

Kelly Gaither[†]
University of Texas at Austin

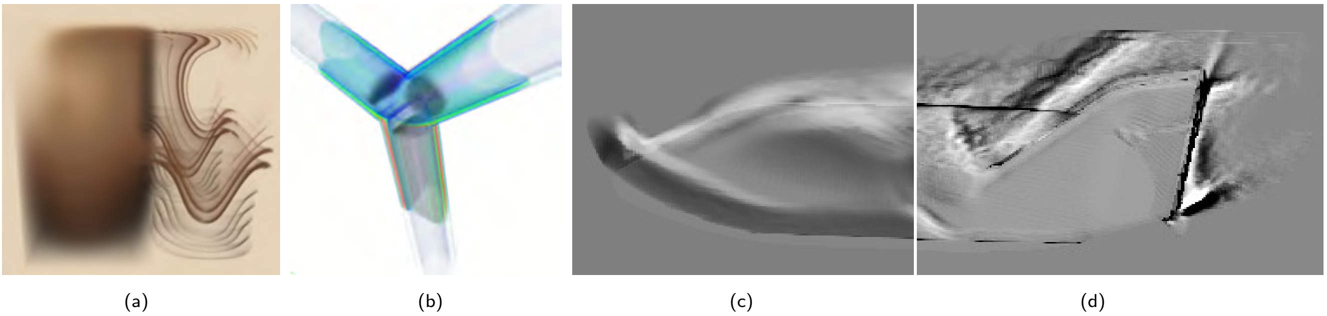


Figure 1: Examples of oriented structural flow visualizations harnessing the effectiveness of illustrative and photographic flow techniques. (a) is an illustrative drawing of temperature advection in a convection dataset. The left half shows a normal volume rendering while the right half shows a contour volume rendering using a da Vinci-inspired color palette. (b) emulates a color-pencil sketch to visualize shear stress in a Y-pipe. (c) shows the bow shock of flow past the X38 spacecraft created from a volumetric Schlieren visualization, while (d) shows the flow and tail fin vortices visualized using a volumetric shadowgraph.

Abstract

Understanding and analyzing complex volumetrically varying data is a difficult problem. Many computational visualization techniques have had only limited success in succinctly portraying the structure of three-dimensional turbulent flow. Motivated by both the extensive history and success of illustration and photographic flow visualization techniques, we have developed a new interactive volume rendering and visualization system for flows and volumes that simulates and enhances traditional illustration, experimental advection, and photographic flow visualization techniques. Our system uses a combination of varying focal and contextual illustrative styles, new advanced two-dimensional transfer functions, enhanced Schlieren and shadowgraphy shaders, and novel oriented structure enhancement techniques to allow interactive visualization, exploration, and comparative analysis of scalar, vector, and time-varying volume datasets. Both traditional illustration techniques and photographic flow visualization techniques effectively reduce visual clutter by using compact oriented structure information to convey three-dimensional structures. Therefore, a key to the effectiveness of our system is using one-dimensional (Schlieren and shadowgraphy) and two-dimensional (silhouette) oriented structural information to reduce visual clutter, while still providing enough three-dimensional structural information for the user's visual system to understand complex three-dimensional flow data. By combining these oriented feature visualization techniques with flexible transfer function controls, we can visualize scalar and vector data, allow comparative visualization of flow properties in a succinct, informative manner, and provide continuity for visualizing time-varying datasets.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading;

Keywords: interactive volume illustration, flow visualization, non-photorealistic rendering, photographic techniques

1 Introduction

Representing and understanding the complexities of three-dimensional flow structures has been a problem of interest for centuries [35]. The ability to understand and predict flow behavior affects our daily lives and safety through applications ranging from cardiology to aircraft design to global weather and climate predictions. Over the past century, many experimental visualization techniques have been developed to capture and depict flow properties. These methods range from photography (e.g., Schlieren photography, shadowgraphy) to dye injection, effectively capturing oriented flow gradients and communicating flow properties. Many of these techniques reduce detail and use lower-dimensional (e.g., 1D and 2D) structural information to depict flow features. Although these experimental techniques have produced some of the most stunning and scientifically meaningful flow imagery, they are expensive, time consuming, and not applicable to large scale problems. Unfortunately, computerized flow visualization techniques are still not as effective as traditional experimental visualizations.

Over the past several centuries, artists and illustrators have also developed very informative representations of complex flow structures, allowing the clear depiction of fundamental flow properties and characteristics. One of the key features of these illustrations is the effective use of lower dimensional visual cues – silhouettes (oriented two-dimensional gradient information) – to succinctly show structural information.

Interestingly, both flow illustration techniques and experimental photographic flow visualization techniques use a common principle to effectively convey flow information: *the display of selective lower-dimensional oriented feature information reduces the visual clutter of turbulent three-dimensional structures, while providing enough oriented structural information for the human visual system to reconstruct the three-dimensional flow*.

Inspired by the commonalities and effectiveness of both traditional illustration and experimental photographic flow visualiza-

[*] e-mail: svakhine, jangy, ebertd@purdue.edu
[†] e-mail:kelly@tacc.utexas.edu

tion methods, we have developed new effective, interactive, flow illustration techniques for the representation, exploration, and analysis of complex, time-varying three-dimensional flow structures. Our visualization system combines new advanced volume visualization, photographic, and illustration techniques to coherently and concisely depict flow features and directionality. We have developed new photographic volume rendering techniques, new illustrative style transfer functions, enhanced multivariate scalar and vector two-dimensional flow transfer function widgets, enhanced banding transfer function widgets, and expanded silhouette and boundary enhancement techniques that enable improved representation of structure orientation and reduction of confusing visual clutter. Our system can create flow sketches, traditional and enhanced Schlieren and shadowgraph effects, and improved experimental advection effects. By expanding the domain of two-dimensional transfer functions and feature enhancement to include multiple flow scalar and vector quantities, our system can interactively compute and depict important flow features and allows interactive comparative visualization for advanced flow analysis. Our system is also effective at highlighting important flow structures through the variation of rendering and shading techniques within the flow volume and can be applied to any three-dimensional scalar or vector dataset, creating new methods for oriented structure visualization.

2 Background

As computers have matured, numerical simulations have emerged as a viable means of modeling and predicting the behavior of complex flow phenomena. Computational methods for visualizing these flows have followed suit to provide a means for understanding and analyzing the results of these simulations. Early flow visualization methods provided powerful methods for querying the solutions, but only allowed very primitive graphical techniques, leaving the analyst to bridge the gap between the visual representation and the physics of the governing equations. Techniques for displaying contours, cutting planes, and isosurfaces gave analysts the first means for visually depicting three-dimensional volumetric flow. These methods, however, are often insufficient to display the complexity of important three-dimensional flow properties such as directionality, divergence, convergence, separation, boundary layers, vortices, wakes, and shocks.

Attempts to visualize these flow properties have led to the development of many additional visualization techniques: vector glyphs [22], line integral convolution (LIC) [3], texture advection (e.g., [12, 16, 30, 32]) and stream tracing (particle tracing, streamlines, and streaklines) [17]. For stream tracing, illumination, transparency [34] and halos [31] have been added to help disambiguate the relationship of the complex thin stream lines. While many of these techniques are effective for two-dimensional flows, the density of their representation often severely limits their effectiveness for three-dimensional flows, particularly in viscous flows and those containing boundary layers. Even recent advancements to improve the clarity of these representations are not sufficient to clearly show the important flow structures (e.g., [11]).

Methods focused on visualizing three-dimensional flow (e.g., volume visualization [5, 6, 19]) have been unable to provide images that sharply focus on the features of interest, while still representing the three-dimensional structure. Recent work in volume flow visualization has also included multiple one-dimensional transfer function techniques for comparative visualization of multiple flow variables [21].

Our approach for flow illustration is based upon previous work in volume rendering algorithms with non-photorealistic (NPR) techniques [9, 14, 29]. Stompel et al. [25] have applied NPR techniques (e.g., stroke based rendering, silhouettes) to improve the understanding of flow volumes, showing the potential of applying volume illustration techniques [9] to scalar flow volumes and streamlines. These volume illustration techniques have been recently extended to interactive rendering by utilizing the extended programmability of modern graphics hardware [18, 13, 20, 25, 26] and rendering different zones of the volume in different styles [27].

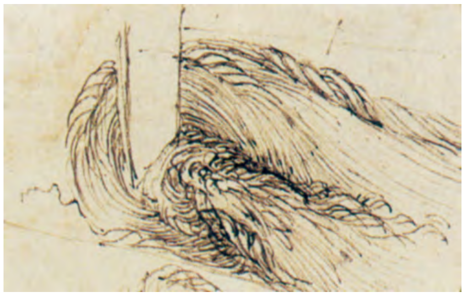


Figure 2: Hand-drawn illustration of water by Leonardo da Vinci from his studies to determine the processes underlying water flow [7].

Our illustration techniques have been applied to a wide variety of datasets to verify their broad applicability. The X38 dataset shows a single time step in the reentry process of the crew return vehicle into the atmosphere at a 30 degree angle of attack. The natural convection dataset simulates a non-Newtonian fluid in a box, heated from below, cooled from above, with a fixed linear temperature profile imposed on the side walls. The jet flow dataset contains 128 time steps from a simulation of unsteady flow emanating from a 2D rectangular slot jet, with the jet source located at the left. The Y-pipe dataset is a simulation of a pressure-driven flow of a non-Newtonian Powell-Eyring fluid in a branched pipe at a nominal Reynolds number of 20 and a viscosity ratio of 0.01. The tornado dataset is a computational simulation of a synthetic tornado. The cloud dataset is a microphysical cumulus cloud large-eddy simulation used to study drop formation processes and entrainment in cumulus cloud formations.

Our system utilizes new illustrative rendering, extended two-dimensional transfer function widgets, illumination, boundary enhancement, and silhouette enhancement to create a variety of effective flow illustrations. The effect of these enhancements and transfer function widgets, the details of our new illustration flow styles, their physical basis and significance, and their relation to classical experimental flow techniques are described in the following sections.

3 Illustrations

Throughout history, illustrators and artists have developed a powerful set of techniques and visual metaphors to succinctly convey information. These techniques emphasize, modify, and subordinate information to the human visual system, including pattern and form (spatial information), color (light frequency), and motion (temporal information). The goal of the illustrator is to create a representation to convey information to the viewer, not recreate reality. Therefore, illustrators use lines, colors, patterns, and selective removal of detail to more effectively convey the essential information. Classic illustrations, such as the flow illustration in Figure 2 by Leonardo da Vinci [35], demonstrate the success of illustrations to convey the structure and motion of complex three-dimensional phenomena. This figure shows the reduction of detail in the flow and the use of silhouette sketching to provide enough oriented structural information to understand the flow without being confused by intricate turbulent flow details. Silhouette sketching effectively captures oriented structural information (two-dimensional) from the complex flow.

3.1 Enhanced Flow Illustration Techniques

Our new system extends recent work in interactive volume illustration techniques [18, 20, 26, 27] to emulate these traditional flow illustration techniques and allows the specification of focus regions and context regions that are rendered with different styles and varying detail, similar to MagicSphere techniques [4]. As in traditional illustrations, this technique provides both attentive focus and context for orientation. Within the focus region, the rendering can show emphasis of information through boundary, silhouette, and transfer function enhancements and oriented shading cues through illumination models. Normally, the contextual regions are rendered in a “sketch” style, where only strong oriented features are rendered as sketch lines through silhouetting strong boundary regions and using almost constant shading. To assign different combinations of effects for different parts of the volume, the system can use these spatially varying, focus plus context regions in combination with transfer functions to assign styles based on voxel properties. These illustration techniques are effective for representing structure and oriented flow patterns, as can be seen in Figures 1, 3, 4, 6, and 7. Just as in traditional illustrations of flow, silhouetting and boundary enhancement reduce the visual clutter inherent in complex natural phenomena and provide enough oriented structural information for our visual system to effectively reconstruct the three-dimensional structure of the flow. The images in Figures 1(a-b), 3(a-b), 6(c), and 7(a-b) show examples of simulated drawings, color pencil sketching, and watercolor paintings of three-dimensional flow datasets. In Figure 3(a), the focus region is placed to highlight the bow shock and density waves present at the nose of the X38 spacecraft. Smooth transitions from the focus rendering to the context rendering provide a gradual blurring and softening of the flow structures along the body of the spacecraft. In Figure 3(b), the cores of the vortices are rendered more opaque and a stronger brown color is used to emphasize this data range, while the surrounding region is rendered “softer” by using a washed gray color with less opacity, providing subtle flow directionality context. In Figures 3(a-b) and 7(a-b), data ranges are used for changing the illustration style. Low temperature/density regions are in gray, while higher regions are rendered in brown. In Figures 7(a) and (b), silhouette rendering provides sketches of most of the temperature contour volumes, while a few selected temperature values are rendered as semi-transparent sketched isosurfaces. These regions are selected using our two-dimensional transfer function widgets.

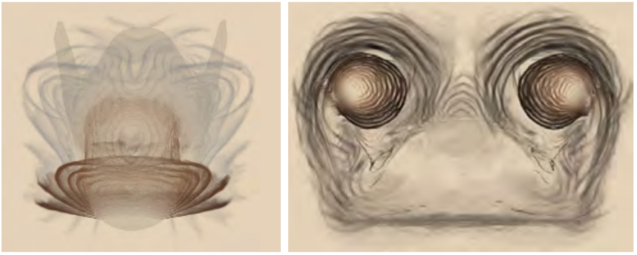

(a) (b)

Figure 3: Volume illustrations of flow around the X38 spacecraft. (a) is an illustration of density flow and shock around the bow, while (b) highlights the vortices created above the fins of the spacecraft.

3.2 Extended Silhouette and Boundary Enhancement Domains

As previously mentioned, boundary and silhouette enhancements are effective techniques for conveying oriented structural information. We have adapted the traditional boundary and silhouette enhancement techniques to allow these calculations to be performed using a vector flow quantity or an additional scalar flow variable’s gradient instead of the gradient of the flow quantity that is used for color and opacity rendering. Table 1 shows the formulas for boundary and silhouette enhancement using the gradient of a scalar field and using a vector field for the enhancements. Figure 4 shows a comparison of boundary and silhouette-enhanced volume rendering (left) and contour-banded volume rendering (right) of the natural convection dataset in (a), a contour-banded volume rendering in (b), using the the gradient for boundary and silhouette enhacement in (c), using the velocity for silhouette enhancement in (d), and using the curl of the velocity field for silhouette enhancement in (e). Figure 4(c) more clearly shows the temperature variation and flow structure than (b) and reduces the visual clutter in the image, while (d) clearly shows the higher velocity flow regions. Figure 4 (e) shows the rotational centers of the convective flow, clearly highlighting the important structures representing the forcing functions in the simulation.

3.3 Extended Two-dimensional Transfer functions

Advanced two-dimensional transfer function widgets can be used to easily select portions of the flow to highlight and to create oriented patterns in smooth regions of the flow. Figure 5 shows our system interface with the focus region highlighted. The user can simply specify pre-defined style templates for both the focus region and context regions. Within each region, styles can also be specified for different portions of the flow (e.g., Figures 7(a) and (b) showing the rendering of a specific temperature region as a thin boundary volume and other temperature regions as sketches). These boundary, silhouette, shading, and two-dimensional transfer function effects can be interactively adjusted and rendered as the user explores the data through the use of fragment shaders, texture-based volume rendering [2], and the latest generation of PC graphics boards. Two-dimensional transfer functions allow us to easily perform boundary enhancement and use both the data and gradient range to determine opacity and apply illustration styles.

We extended the interactive two-dimensional transfer functions of Kniss et al. [15] in several ways. First, we added illustration style transfer functions to select spatial and data value regions for the application of different illustrative rendering styles. Second, we extended the domains of the two-dimensional transfer function widgets. Finally, we added a new banding transfer function widget. This banding widget uses a sine function to map the scalar data value to opacity, creating alternating low and high opacity contour volumes. Users can control the periodicity of the sine function and also apply a power function to create sharper or softer bands (e.g, $sin^{0.5}, sin^2$). This contour banding was used to create most of the illustration and advection results in this paper and can also be used for Schlieren rendering. The combination of our banding transfer function and silhouetting allows us to create effective contour volumes with only oriented structural information, thus reducing visual clutter and overcoming many of the problems with texture advection and LIC techniques. The equation for the banding widget is shown in Table 1.

We also extended the domain of the two-dimensional transfer function to allow several different values to be used as the second domain, including gradient magnitude ($|\nabla U|$), an additional scalar data variable (U_2), the Laplacian ($\nabla^2 U$), $|U|$ (U : vector data), and the curl ($|\nabla \times U|$). These vector calculus operators were chosen because they are fundamental building blocks in constructing functions that are especially useful for analysis and flow visualization. The display of data computed from these operators can be interactively selected and manipulated using simple two-dimensional transfer function widgets. The operators and techniques constructed from these operators are summarized in Table 1. As compared to previous multivariate flow transfer functions [21],

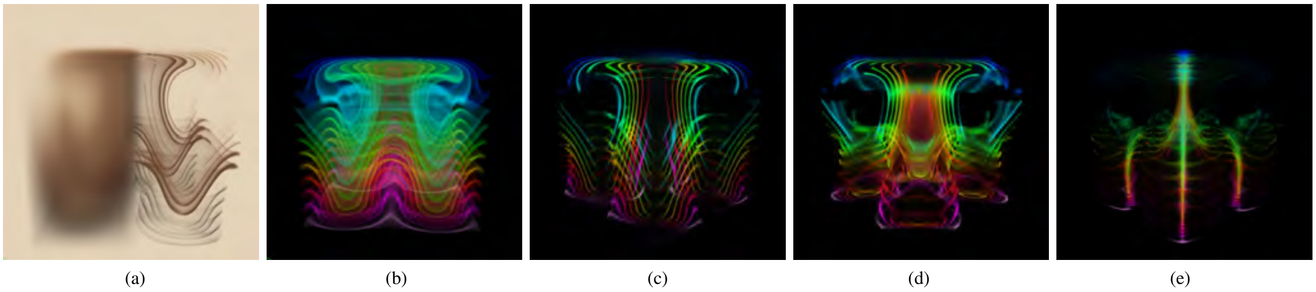

(a) (b) (c) (d) (e)

Figure 4: Comparison of contour volume renderings of the convection dataset with different enhancements. (a) is a boundary and silhouette-enhanced volume rendering (left) and contour-banded volume rendering (right). (b) is a normal contour volume rendering without silhouette enhancement, while (c), (d) and (e) are with silhouette enhancement. The temperature gradient field is used for silhouette enhancement in (c). The velocity field is used in (d), and the curl of the velocity field is used in (e) for silhouetting.

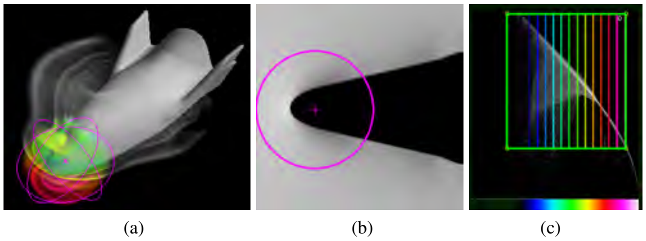

(a) (b) (c)

Figure 5: System interface showing the focus and context selection of illustration styles in the volume (a) and 2D slice viewer (b), and the system's 2D transfer function controls (c).

our work allows several variables and derived flow quantities to be combined into one two-dimensional transfer function widget so that any displayable attribute can be the selective product and enhancement of possibly different two-dimensional domains.

Figures 1, 4, and 6 show several examples of extracting important features and comparing flow variables using these multivariate two-dimensional transfer functions. In Figure 1(b), shear stress and vorticity of the Y-pipe dataset are used in order to emphasize the interesting flow properties near the Y-junction, where backflow is introduced. This image uses a colored pencil rendering style.

Figure 6(a) more clearly visualizes the temperature flow by using the Laplacian to separate inflow, outflow and noflow regions. Heat inflow is color coded as red, heat outflow is color coded as blue and noflow regions are color coded as white. Figure 6(b) uses the same color coding of the Laplacian to highlight velocity regions within the tornado dataset with the zero of the velocity magnitude Laplacian silhouetted, while the vortex core is rendered as an illuminated yellow isosurface. In Figure 6(c), turbulent kinetic energy (TKE), velocity, and the Laplacian of TKE are used to highlight the entrainment and mixing in the cloud. The velocity field of the cloud dataset is used to enhance the silhouette and boundary, while the zero values of the Laplacian are used to further enhance the boundaries, rendered in white. In Figure 6(d), high second derivative magnitude regions of the temperature in the convection dataset are shown in colored bands, showing the origination of the convection poles that form as the flow heats and convects through the domain. Low magnitude regions are rendered in gray scale.

Our ability to render focus regions and context regions using different shading techniques and transfer functions enables global flow visualization and localized flow analysis. Different styles can also be separately applied to different data ranges within the focus and context regions. A key technique for reducing visual clutter is the silhouetting and boundary enhancement of contour volumes. Both illustrations and photographic flow visualization techniques successfully use similar oriented structure representations. For showing variation and orientation of data within relatively homogeneous regions, we utilize our banding transfer function widget to produce contour volumes. As compared to more traditional particle advection techniques, oriented contour volumes more effectively visualize converging and diverging flows and provide coherent animation of time varying flows, as shown in Figures 4, 6, and 7. As long as the scalar or vector volume data variable exhibits relative time coherence, these thin volume contours will provide a smooth animation of the time variation.

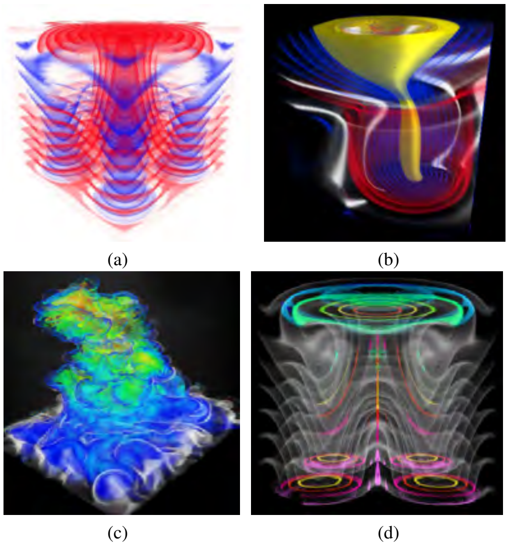

(a) (b) (c) (d)

Figure 6: Use of two-dimensional transfer function with the Laplacian operator and other flow quantities. (a) shows heat inflow (red) and outflow (blue). (b) shows all values of the Laplacian of velocity magnitude in the tornado dataset. (c) visualizes the cloud TKE using the Laplacian to highlight boundaries (white) and velocity for silhouetting. (d) highlights emerging flow structures in the convection dataset using banding of the second derivative magnitude of the temperature field.

4 Photographic Techniques

Our work is inspired by the effectiveness of both illustration and photographic experimental flow visualization techniques. Photographic flow visualization techniques have been used for over one hundred years to capture and allow analysis of three-dimensional flows. While illustrators use oriented silhouettes that capture two-dimensional gradient projections to reduce visual clutter in three-dimensional data, photographic flow techniques use one-

Table 1: Illustration and Rendering Techniques with Physical Analogue and Operators

Technique	Equation	Physical Analogue
Boundary Enhancement	$o_e = o_o \cdot \lvert\overrightarrow{\nabla}(P)\rvert^n$ $o_e = o_o \cdot \lvert\overrightarrow{U}\rvert^n$	Field Lines of Constant K Defined by Operators $U = Vector$
Silhouette Enhancement	$o_e = o_o \cdot (1 - \lvert(\overrightarrow{\nabla}'(P) \cdot \overrightarrow{V})\rvert)^n$ $o_e = o_o \cdot (1 - \lvert(\overrightarrow{U} \cdot \overrightarrow{V})\rvert)^n$	Field Lines of Constant K Oriented in the Direction of V $U = Vector$
Banding Widget	$o_e = (\frac{1+sin(o_o * \pi * m)}{2})^n$	*m* Contour Volumes of Constant K

Operator	Equation	Physical Analogue
$\lvert U\rvert$	$\sqrt{\sum(U_i)^2}$	$U = Scalar \longrightarrow AbsoluteValue$ $U = Vector \longrightarrow VectorMagnitude$
$\nabla \cdot U$	$\frac{\partial U}{\partial x} + \frac{\partial U}{\partial y} + \frac{\partial U}{\partial z}$	$U = Scalar \longrightarrow Gradient$ $U = Vector \longrightarrow Divergence$ $U = ViewVector \longrightarrow Silhouette$
$\nabla \times U$	$(\frac{\partial U_z}{\partial y} - \frac{\partial U_y}{\partial z})\overrightarrow{i} + (\frac{\partial U_x}{\partial z} - \frac{\partial U_z}{\partial x})\overrightarrow{j} + (\frac{\partial U_y}{\partial x} - \frac{\partial U_x}{\partial y})\overrightarrow{k}$	Curl
∇^2	$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2}$	Laplacian
$d\varepsilon$	$(d\varepsilon_x, d\varepsilon_y)$	Schlieren
$\nabla \cdot \varepsilon$	$\frac{\partial \varepsilon_x}{\partial x} + \frac{\partial \varepsilon_y}{\partial y}$	Shadowgraph

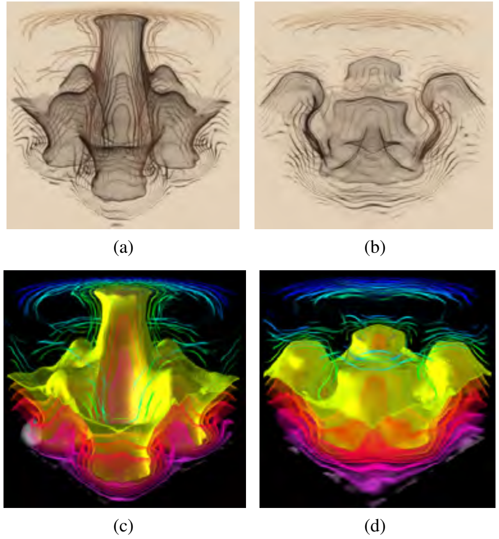


Figure 7: Volume illustrations of two time steps of the temperature variation in the convection dataset (top). Contour volume rendering and silhouetting of the convection dataset (bottom left) and of the next time step (bottom right). The use of contour volumes and sketching provides smooth continuity of time-varying flows.

dimensional gradient variation to further reduce the visual clutter. Even though these techniques reduce the information to oriented one-dimensional variation, they are still effective at capturing enough important oriented flow information from complex three-dimensional flows to allow the scientist to reconstruct the three-dimensional flow structures and properties, as evidenced by their long successful use in experimental fluid dynamics. The two most commonly used flow photographic techniques are shadowgraphs and Schlieren photography [1, 10, 23, 24].

The Schlieren photographic process uses an optical system to capture density gradient variations in inhomogeneous media (Figure 8). When illuminated with a parallel light source, the optical inhomogeneities refract the light rays in proportion to the gradients of the refractive index projected on the *xy*-plane, which is orthogonal to the light direction. This refraction produces a two-dimensional

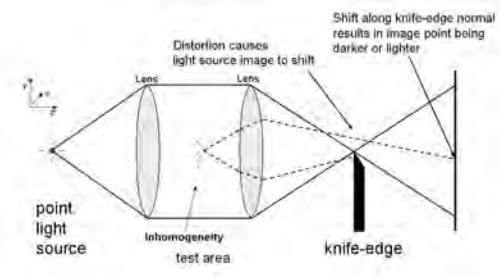

Shift along knife-edge normal results in image point being darker or lighter
Distortion causes light source image to shift
Lens
Lens
point light source
Inhomogeneity
test area
knife-edge

Figure 8: Simple Schlieren system diagram

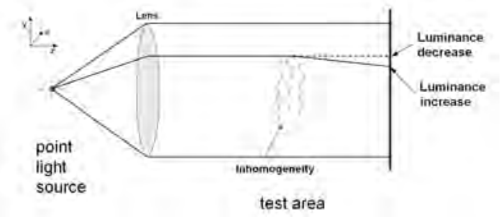


Figure 9: Simple Shadowgraph system diagram

displacement of the light ray (Equation 1), and the Schlieren knife-edge apparatus is able to measure this displacement's projection in a specific one dimensional direction on the *xy*-plane, called the knife-edge angle, which is the main parameter of the process. Therefore, in classic Schlieren photographs, darker regions correspond to negative displacement, brighter regions correspond to positive displacement, and the knife-edge angle is usually chosen to provide the best visualization of particular features. Extensions of Schlieren techniques include using a circular cutoff that visualizes the displacement vector's magnitude, and color Schlieren photographs, that essentially create a "color-wheel" visualization of the displacement.

$$\vec{\varepsilon} = (\varepsilon_x, \varepsilon_y) = k(\int_{z_0}^{z_1} \frac{\partial \rho}{\partial x} dz, \int_{z_0}^{z_1} \frac{\partial \rho}{\partial y} dz) \qquad (1)$$

The shadowgraph observation setup is simpler, since it only contains a parallel light source and a screen (Figure 9). Due to inhomogeneities in the test area, refraction within the affected region displaces the light ray before it reaches the screen, changing the luminance distribution on the screen. Note that this luminance shift only occurs when the displacement vector ε along the x or y axes changes. Thus, the resulting luminance map value can be approximated by the following:

$$L = \int_{z_0}^{z_1} (\frac{\partial \varepsilon_x}{\partial x} + \frac{\partial \varepsilon_y}{\partial y}) dz = \int_{z_0}^{z_1} (\frac{\partial \rho^2}{\partial x^2} + \frac{\partial \rho^2}{\partial y^2}) dz \tag{2}$$

There have been several computer simulations of Schlieren photography of two-dimensional flows using image processing techniques [10]. For three-dimensional flows, the basic non-interactive simulations are described in [28, 33]; however, interactive Schlieren computer visualizations of three-dimensional flows have not been produced, although some commercial packages incorrectly refer to boundary enhancement as Schlieren results.

We have extended our texture-based volume rendering system to simulate and enhance the Schlieren and shadowgraph techniques. The Schlieren integral is calculated using two rendering passes. In the first pass, slice rendering performs the integration of the separate components of the displacement vector $\vec{\varepsilon}$ and stores them in the R and G components of a floating-point pixel buffer. Since the classic Schlieren setup places the light source directly behind the test area, we use front-to-back slicing to integrate along the light path. Note that this integral of positive and negative displacements allows them to cancel out their effects as in the experimental Schlieren system. For visualization of two-dimensional datasets and single slice visualization, only a single slice is rendered in this pass. The second pass transforms the displacement vector map to the actual image, scaling the resulting color by the sensitivity coefficient, s, according to the particular Schlieren method we want to simulate: knife-angle cutoff (Equation 3, where $\vec{k}$ is a normal to the knife edge), or circular cutoff (Equation 4).

$$I = s\frac{(1 + (\vec{\varepsilon} \cdot \vec{k}))}{2} \tag{3}$$

$$I = s|\vec{\varepsilon}| \tag{4}$$

The shadowgraph integral (Equation 2) is computed analogously, except that the value of L is scalar. Therefore, during the second rendering pass, the calculations simplify to the following:

$$I = s\frac{(1 + L)}{2} \tag{5}$$

The most obvious and important limitation of the classic Schlieren photographic approach is that the experimental apparatus does not allow flexible control over which portion of the data is photographed: it always covers the entire test area, thus lacking the ability to focus on a particular location. There is active research in physics to extend the classic Schlieren approach and construct more complicated apparatuses to overcome these limitations. Our Schlieren renderer, however, allows us to easily filter the data according to the value or location, using regular transfer functions and distance-based transfer functions during the first pass, thus overcoming these obstacles and producing improved Schlieren images.

Figures 1(c), 10(a-b), 12(c), and 13(c) show examples of our two-dimensional and volume enhanced Schlieren rendering. Figures 10(a-b) show Schlieren renderings of a two-dimensional slot jet flow with various rendering enhancements. Figure 1(c) is a volume Schlieren rendering applied to the X38 density dataset, clearly showing the bow shock and flow. Figure 13(c) shows another enhanced Schlieren rendering of the same data, simulating wind tunnel effects and clearly showing the bow shock and vortices induced above the fins. Figure 12(c) shows the Schlieren rendering of the vapor water in the cloud dataset. Figures 1(d), 10(c), 12(d), and 13(d-e) show examples of two-dimensional and volume enhanced shadowgraphs. Figure 10(c) shows a shadowgraph rendering of a two-dimensional slot jet flow. Figure 1(d) and Figure 13(d-e) show shadowgraph renderings of the X38 density dataset. Figure 12(d) shows a shadowgraph rendering of the velocity magnitude in the tornado dataset. From Figure 12(d), the center of rotation is clearly visible and the edges of the separate flows are shown.

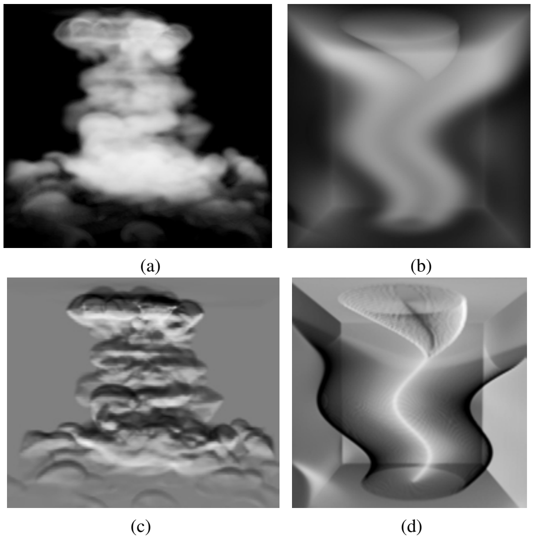
```
PixelOut ivis_Schlieren(ivis_v2f IN, uniform sampler3D GradientMag,
                    uniform samplerRECT Prev_Buffer,
                    uniform float4 schlParams, vectorMin,vectorMax)
{
    PixelOut OUT;
    float4 lookup,integration_by_dz;
    float3 v0, dv, gradient;
    float dz = schlParams[1];

    //Extracting the gradient
    v0 = vectorMin.xyz;
    dv = vectorMax.xyz - vectorMin.xyz;
    lookup = f4tex3D(GradientMag, IN.WorldPos.xyz);
    gradient = (v0+(lookup.xyz*dv))*lookup.a*schlParams[0];

    //Computing Gradient projection on xy-plane
    integration_by_dz.x = (dot(gradient, IN.RightVector))*dz;
    integration_by_dz.y = (dot(gradient, IN.UpVector))*dz;

    //Additive blending with the frame buffer
    float4 previous_pixel = f4texRECT(Prev_Buffer,IN.pix_pos.xy);
    OUT.COL = previous_pixel+integration_po_dz;
    return OUT;
}
```

Figure 11: Sample Schlieren Cg shader


(a)

(b)

(c)

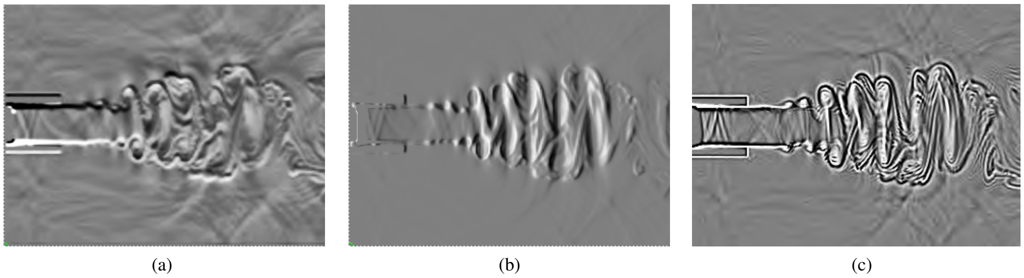
(d)

Figure 12: Comparisons of normal volume renderings (a-b) and enhanced photographic simulation renderings (c-d). (c) is an enhanced Schlieren rendering of water vapor in the cloud dataset and (d) is a shadowgraph rendering of velocity magitude in the tornado dataset.

5 Discussion

Table 1 shows an overview of all of the illustration and photographic techniques with the equations and physical relevance. All of the effects are implemented as fragment programs in our illustrative volume renderer, using Nvidia's Cg language. An example Cg shader for our Schlieren technique is shown in Figure 11.

Our results were generated on a Pentium Xeon 3.20 Ghz workstation with 2.00 Gb RAM and a Nvidia GeForce 7800 GTX (256 Mb VRAM) graphics card. For interactive visualization (image size



(a) (b) (c)

Figure 10: Enhanced Schlieren and Shadowgraph renderings of two dimensional jet flow. (a) is a simulation of standard Schlieren gradient sampling with a horizontal knife-edge angle. (b) alters (a) by using a vertical knife edge angle and removing low gradients. (c) shows a shadowgraph rendering for the same dataset.

350×350 pixels), we use a lower-quality mode (500 slices) and achieve approximately 20 fps for illustrative rendering and 30 fps for Schlieren rendering. Final quality images used in this paper were generated with the highest quality rendering settings (1000 slices) at 6-10 fps for illustrative rendering, depending on the complexity of the effects, and 15 fps for Schlieren rendering.

Initial feedback from domain scientists has been very positive. The visualizations produced from this system have allowed them to see aspects and behavior in the fluids that are not achievable using traditional visualization techniques. Rendering the convection dataset using the silhouette and boundary enhancement allows the scientists to easily separate the flow behavior and clearly shows the forcing function behind the simulation. Sketch and photography inspired visualization of the X38 dataset show properties of the flow that have never been shown before, allowing scientists to see the trip point on the body of the X38 causing the formation of strong vortices between the fins (see Figures 13 (c)-(e) and Figure 14).

6 Conclusion

In summary, our new visualization system addresses the need to improve computational visualization methods by taking advantage of the benefits of illustrations and experimental visualization methods. Both illustrations and experimental visualizations are adept at focusing the viewer's attention to specific facets of the display, while reducing visual clutter by placing emphasis on specific regions of interest and using oriented structures in the display to provide depth and three-dimensional structure. All of our illustrative, experimental, and photographic-based techniques are effective because they only use one-dimensional or two-dimensional oriented structural information and allow both spatially-varying (focus + context) and data-varying specification of rendering styles.

Our new system allows interactive visualization, exploration, and analysis of scalar, vector and time-varying volume datasets. We have successfully taken advantage of key methods from illustration by using oriented structural information and developing images that emulate scientific sketches and watercolors. We have also successfully emulated experimental flow visualization techniques by developing and demonstrating methods for producing enhanced Schlieren integral photographs and shadowgraphs on three-dimensional scalar and vector volumes.

We plan to extend this work by providing more tools for visualizing time-varying datasets, larger and more varied grid structures, and fully developing the comparative visualization tools in our system. We also plan to optimize the interactive shader performance and more fully explore the use of these techniques to volumes generated from other domains.

7 Acknowledgements

Our work is inspired by the great classic flow images by Leonardo da Vinci, the experimental flow photography of Van Dyke [8], and discussions with Penny Rheingans. We would like to thank Dr. Steven H. Collicott (Purdue School of Aeronautics and Astronautics) for the useful discussions about Schlieren photography, Nvidia for their hardware donations to support this research, and the anonymous reviewers for the beneficial suggestions for improving this paper. This material is based upon work supported by the US National Science Foundation under grants NSF ACI-0081581, NSF ACI-0121288, NSF ACI-0328984, NSF IIS-0098443, NSF ACI-9978032, and NSF ACI-0222675. We would like to thank David Marcum (Mississippi State University), Bill Barth (The University of Texas at Austin), Sonia Lasher-Trapp (Purdue University) and Roger Crawfis (Ohio State University) for providing us with the datasets used in this work.

References

[1] D. Brown, T. Cole, B. Peters, J. Wilson, and G. Havener. Shadowgraph/Schlieren photography for aerodynamic applications. In *Proceedings 18th AIAA Aerospace Ground Testing Conference*, 1994.

[2] B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. *ACM Symposium on Volume Visualization*, 1994.

[3] B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 263–270. ACM Press, 1993.

[4] P. Cignoni, C. Montani, and R. Scopigno. Magicsphere: an insight tool for 3D data visualization. *Comput. Graph. Forum*, 13(3):317–328, 1994.

[5] R. Crawfis and N. Max. Texture splats for 3D scalar and vector field visualization. In *IEEE Visualization '93 Proceedings*, pages 261–266. IEEE Computer Society, 1993.

[6] R. Crawfis, N. Max, B. Becker, and B. Cabral. Volume rendering of 3D scalar and vector fields at LLNL. In *Proceedings Supercomputing '93*, pages 570–576. ACM Press, 1993.

[7] Leonardo da Vinci. *Old Man Seated on Rocky Outcrop, Seen in Profile to the Right, with Water Studies*. Windsor Castle, Royal Liberty, RL 12579r, c. 1510-1513.

[8] M. Van Dyke, editor. *An Album of Fluid Motion*. The Parabolic Press, Stanford, California, 1982.

[9] D. Ebert and P. Rheingans. Volume illustration: non-photorealistic rendering of volume models. In *Proceedings of Visualization '00*, pages 195–202. IEEE Computer Society Press, 2000.

[10] L. Guo and L. Zhang. The development of the technique of computer simulating color-schlieren. *Journal of Visualization*, 7(3), 2004.

[11] V. Interrante and C. Grosch. Strategies for effectively visualizing 3D flow with volume LIC. In *VIS 1997: Proceedings of the 8th conference on Visualization 1997*. IEEE Computer Society, 1997.

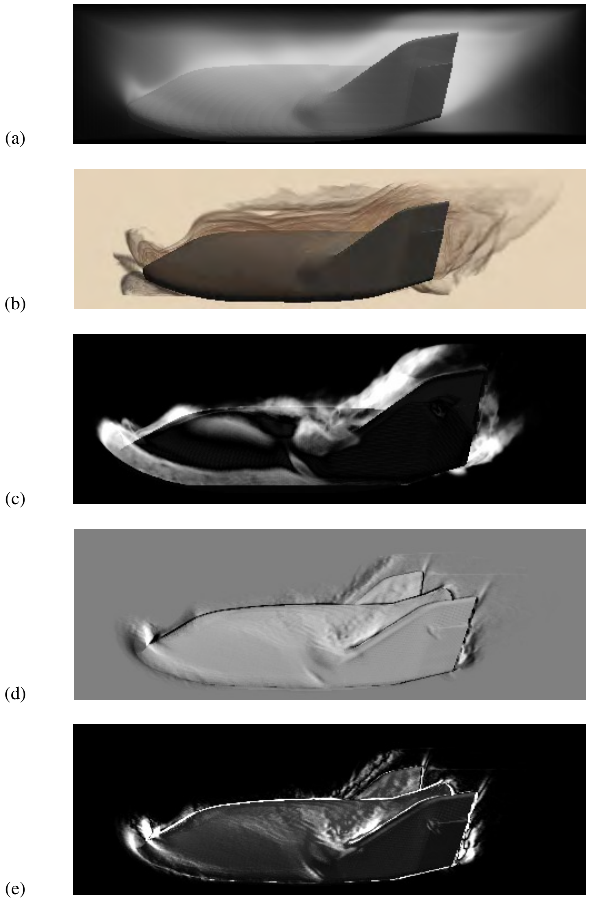


Figure 13: Normal volume rendering (a), an illustrative contour-banded volume rendering (b), enhanced Schlieren with circular cutoff (c), and shadowgraph renderings (d-e) of the X38 dataset from the side, simulating wind tunnel effects. The white band on the nose of the spacecraft (c-e) indicates the presence of a bow shock. The white band behind the fins shows the presence of a wake formed from vortex cores. (e) alters (d) by visualizing the absolute value of the shadowgraph operator in Table 1.

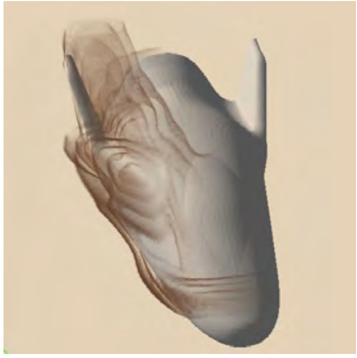


Figure 14: The X38 dataset using boundary and silhouette enhancement clearly shows the origin point and the formation of the vortex tube. The right half is not volume rendered to better show the surface geometry of the X38.

[12] B. Jobard, G. Erlebacher, and M. Hussaini. Hardware-accelerated texture advection for unsteady flow visualization. In *Proceedings of Visualization 2000*. IEEE Computer Society, 2000.

[13] A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. *to appear in IEEE Visualization*, 2005.

[14] R. Kirby, H. Marmanis, and D. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proceedings of the conference on Visualization '99*, pages 333–340, Oct 1999.

[15] 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.

[16] G. Li, U. Bordoloi, and H. Shen. Chameleon: An interactive texture based rendering framework for visualizing three-dimensional vector fields. In *Proceedings of IEEE Visualization 2003*, pages 241–248, 2003.

[17] H. Loffelmann and E. Gröller. Enhancing the visualization of characteristic structures in dynamical systems. In *VIS '96: Proceedings of the 7th IEEE Visualization 1996 Conference (VIS '96)*. IEEE Computer Society, 1996.

[18] E. B. Lum and K.-L. Ma. Hardware-accelerated parallel non-photorealistic volume rendering. In *Hardware-accelerated parallel non-photorealistic volume rendering*, pages 67–74. ACM Press, 2002.

[19] N. Max, L.C. Leedom, and R. Crawfis. Flow volumes for interactive vector field visualization. In *VIS 1993: Proceedings of the 4th conference on Visualization 1993*. IEEE Computer Society, 1993.

[20] Z. Nagy, J. Schneider, and R. Westermann. Interactive volume illustration. In *Proceedings of Vision, Modeling and Visualization Workshop '02*, November 2002.

[21] S. Park, B. Budge, L. Linsen, B. Hamann, and K. Joy. Multi-dimensional transfer functions for interactive 3D flow visualization. In *Pacific Graphics*, 2004.

[22] F. Post, F. Post, T. Van Walsum, and D. Silver. Iconic techniques for feature visualization. In *VIS '95: Proceedings of the 6th conference on Visualization '95*. IEEE Computer Society, 1995.

[23] G. Settles. *Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media*. Springer-Verlag, 2001.

[24] A. Shenoy, A. Agrawal, and S. Gollahalli. Quantitative Evaluation of Flow Computations by Rainbow Schlieren Deflectometry. *AIAA Journal*, 36(11), 1998.

[25] A. Stompel, E. Lum, and K.-L. Ma. Feature-enhanced visualization of multidimensional, multivariate volume data using non-photorealistic rendering techniques. In *Proceedings of Pacific Graphics 2002 Conference*, 2002.

[26] N. Svakhine and D. Ebert. Interactive volume illustration and feature halos. *Pacific Graphics '03 Proceedings*, 15(3):67–76, 2003.

[27] N. Svakhine, D. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3), 2005.

[28] Y. Tamura and K. Fujii. Visualization for Computational Fluid Dynamics and the Comparison with Experiments. In *Proceedings of the AIAA 8th Applied Aerodynamics Conference*, 1990.

[29] S. Treavett and M. Chen. Pen-and-ink rendering in volume visualisation. In *IEEE Visualization 2000*, pages 203–210, October 2000. ISBN 0-7803-6478-3.

[30] D. Weiskopf, G. Erlebacher, and T. Ertl. A texture-based framework for spacetime-coherent visualization of time-dependent vector fields. In *VIS 2003: Proceedings of the 14th conference on Visualization 2003*. IEEE Computer Society, 2003.

[31] A. Wenger, D. Keefe, S. Zhang, and D. Laidlaw. Interactive volume rendering of thin thread structures within multivalued scientific data sets. *IEEE Transactions on Visualization and Computer Graphics*, 10(6), 2004.

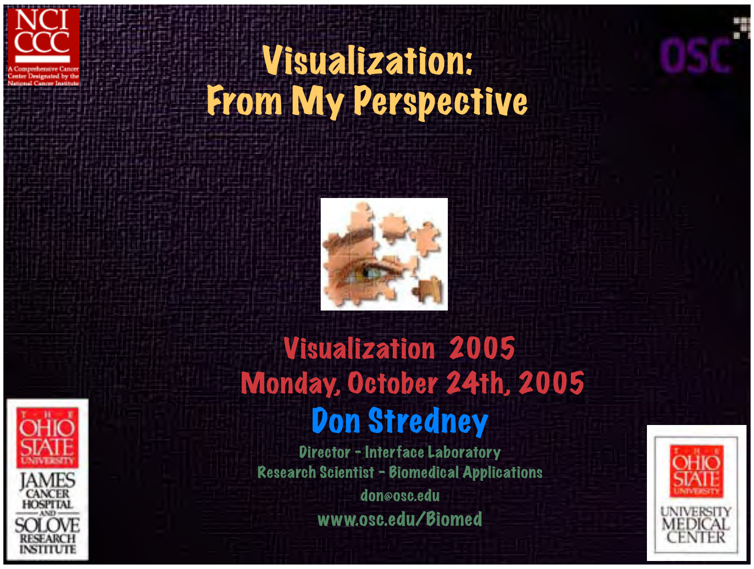
[32] J.J. Van Wijk. Image based flow visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002.

[33] L. Yates. Images constructed from computed flowfields. *AIAA Journal*, 31:1877–1884, 1993.

[34] M. Zöckler, D. Stalling, and H.-C. Hege. Interactive visualization of 3D-vector fields using illuminated stream lines. In *VIS '96: Proceedings of the 7th conference on Visualization '96*. IEEE Computer Society Press, 1996.

[35] F. Zöllner. *Leonardo da Vinci Sketches and Drawings*. Taschen, 2004.

Visualization: From Illustrator’s Perspective



Visualization: From My Perspective

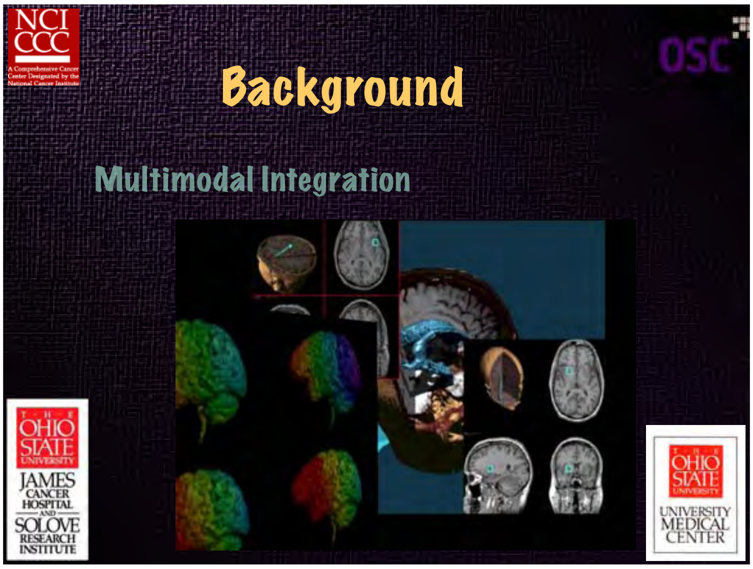
Visualization 2005
Monday, October 24th, 2005

Don Stredney
Director - Interface Laboratory
Research Scientist - Biomedical Applications
don@osc.edu
www.osc.edu/Biomed

Overview

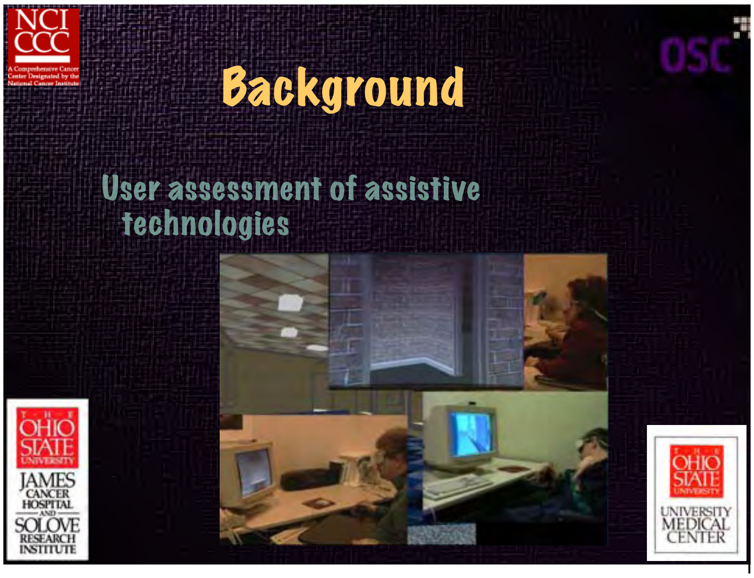
- Background
- Sensemaking
- Representation
- Relevant Applications - Surgical Simulation
- Integration of NPR Techniques
- Aesthetic Relevance
- Cultivating Adoption & Creating Diverse Teams
- References

Background



Background

Multimodal Integration



Background

User assessment of assistive technologies

Background

Residential Training in Regional Anesthesia

Background

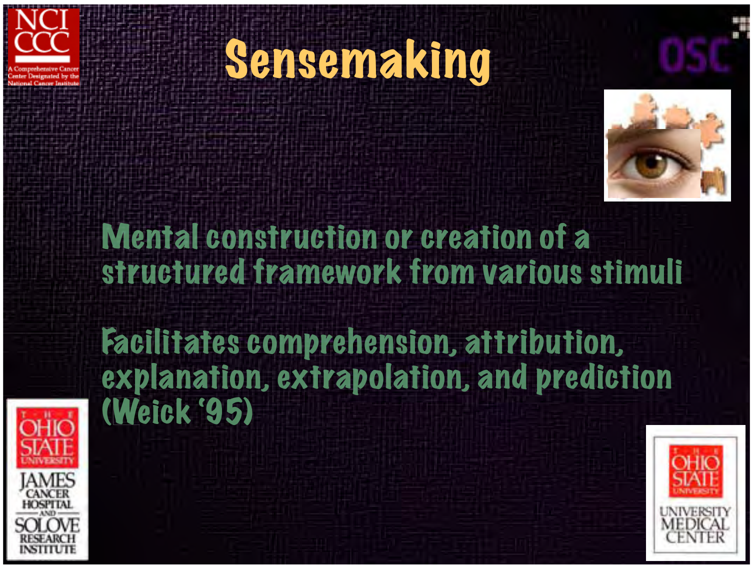
Residential Training in
Regional Anesthesia

Background

Pre-operative assessment

Background

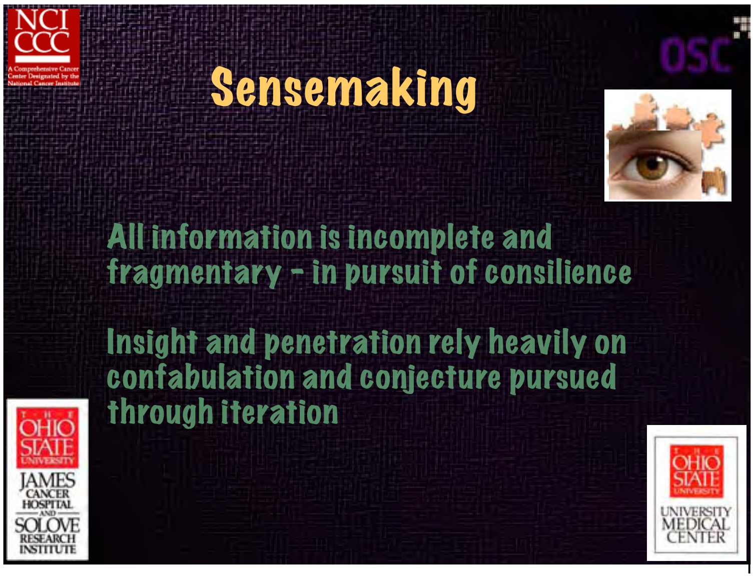
Functional Sinus Surgery
Simulaion



Sensemaking

Mental construction or creation of a structured framework from various stimuli

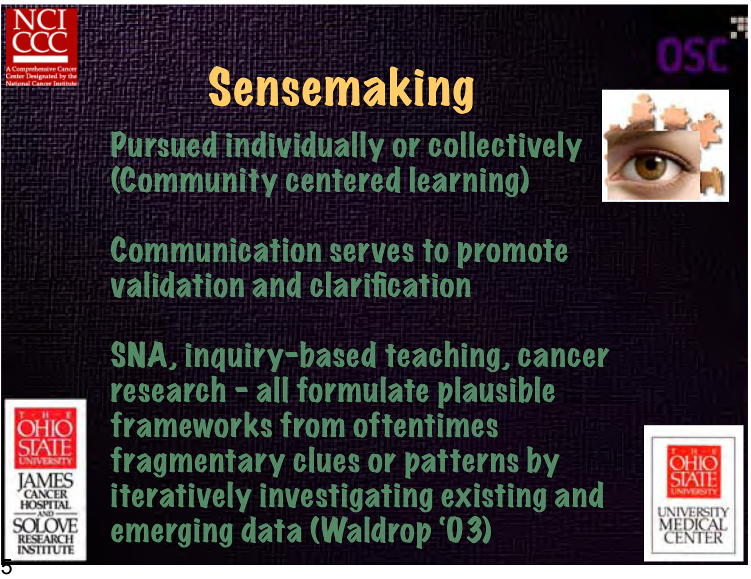
Facilitates comprehension, attribution, explanation, extrapolation, and prediction (Weick '95)



Sensemaking

All information is incomplete and fragmentary - in pursuit of consilience

Insight and penetration rely heavily on confabulation and conjecture pursued through iteration

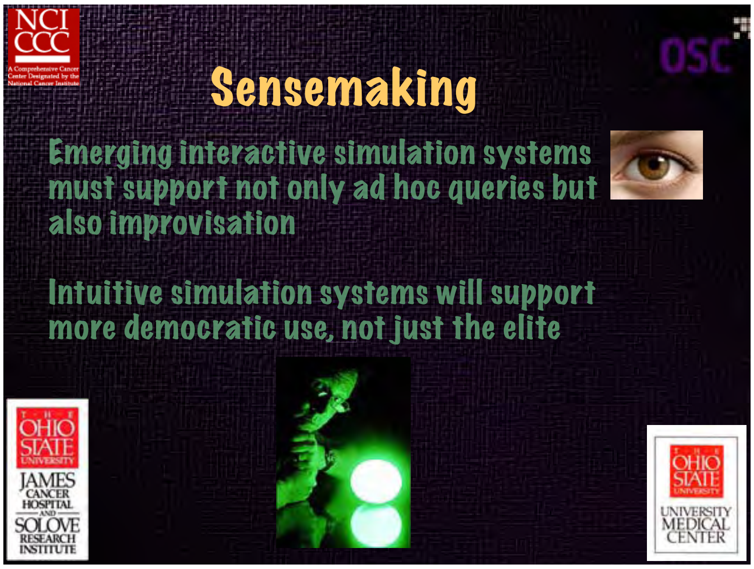


Sensemaking

Pursued individually or collectively (Community centered learning)

Communication serves to promote validation and clarification

SNA, inquiry-based teaching, cancer research - all formulate plausible frameworks from oftentimes fragmentary clues or patterns by iteratively investigating existing and emerging data (Waldrop '03)



Sensemaking

Emerging interactive simulation systems must support not only ad hoc queries but also improvisation

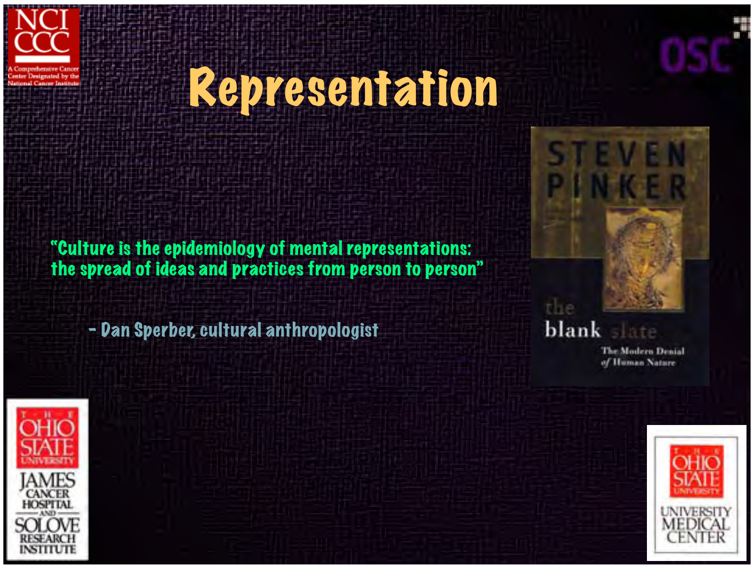
Intuitive simulation systems will support more democratic use, not just the elite

Sensemaking

To support sensemaking, simulations should be:

Self-directed (student centered)
Interactive (non-deterministic)
Declarative and Prodedural

These help promote self-education



Representation

"Culture is the epidemiology of mental representations: the spread of ideas and practices from person to person"

– Dan Sperber, cultural anthropologist

Representation

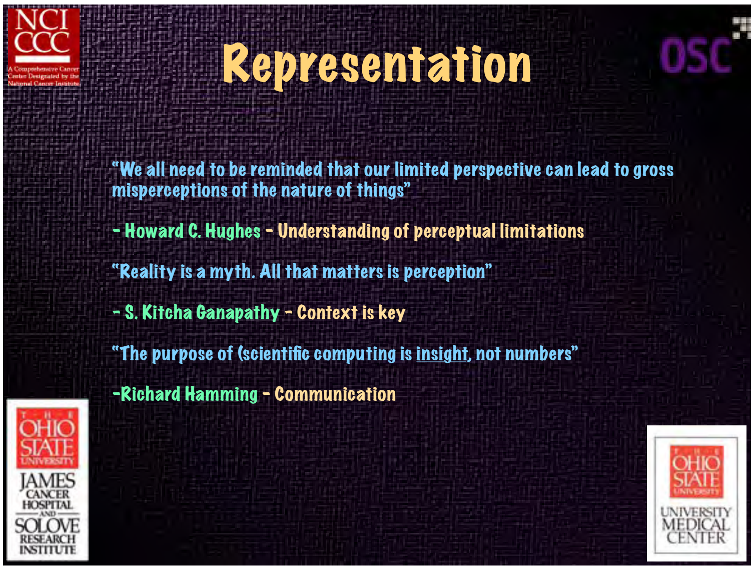
Representation is pervasive

There is a great need to democratize visual literacy

We need to develop systems that allow for a wide range of represntations for communication, contextual realism, and expression

Representation

Power of Representations



Representation

"We all need to be reminded that our limited perspective can lead to gross misperceptions of the nature of things"

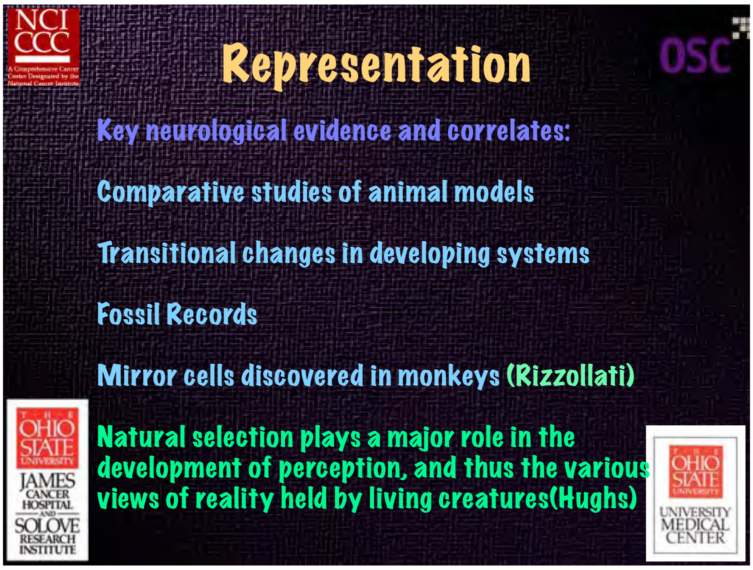
- Howard C. Hughes - Understanding of perceptual limitations

"Reality is a myth. All that matters is perception"

- S. Kitcha Ganapathy - Context is key

"The purpose of (scientific computing is insight, not numbers"

-Richard Hamming - Communication



Representation

Key neurological evidence and correlates:

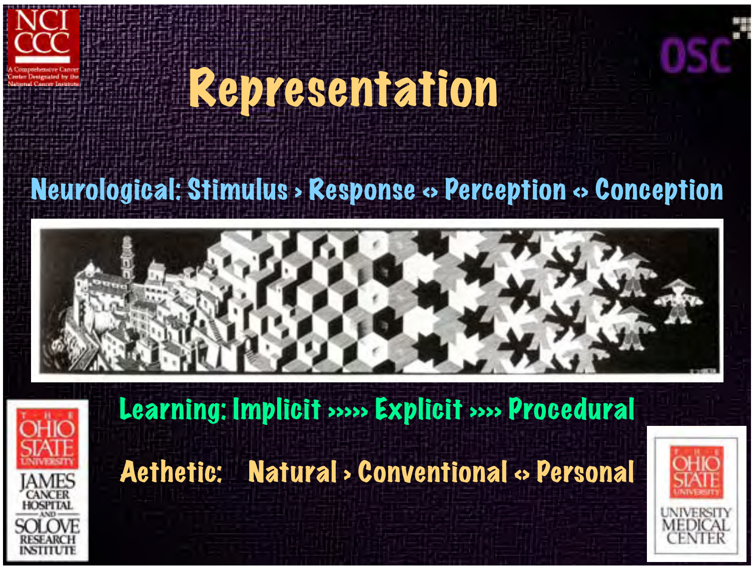
Comparative studies of animal models

Transitional changes in developing systems

Fossil Records

Mirror cells discovered in monkeys (Rizzollati)

Natural selection plays a major role in the development of perception, and thus the various views of reality held by living creatures(Hughs)

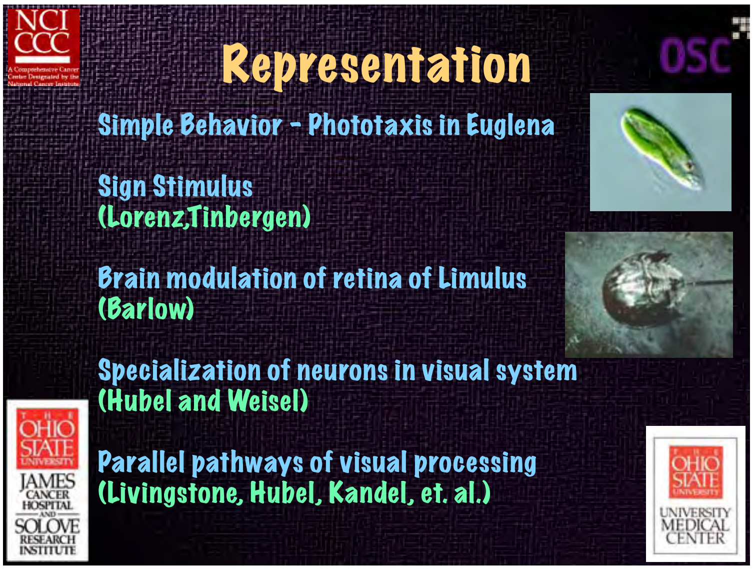


Representation

Neurological: Stimulus › Response ‹› Perception ‹› Conception

Learning: Implicit ››››› Explicit ›››› Procedural

Aethetic: Natural › Conventional ‹› Personal



Representation

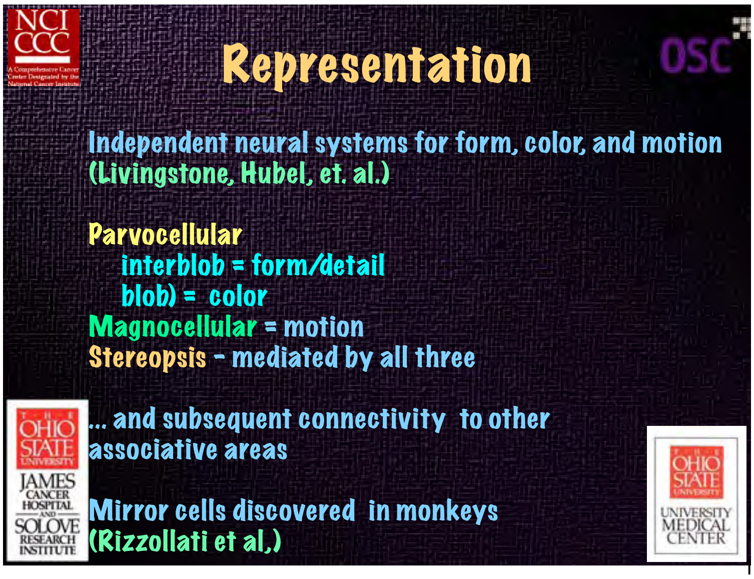
Simple Behavior - Phototaxis in Euglena

Sign Stimulus
(Lorenz,Tinbergen)

Brain modulation of retina of Limulus
(Barlow)

Specialization of neurons in visual system
(Hubel and Weisel)

Parallel pathways of visual processing
(Livingstone, Hubel, Kandel, et. al.)



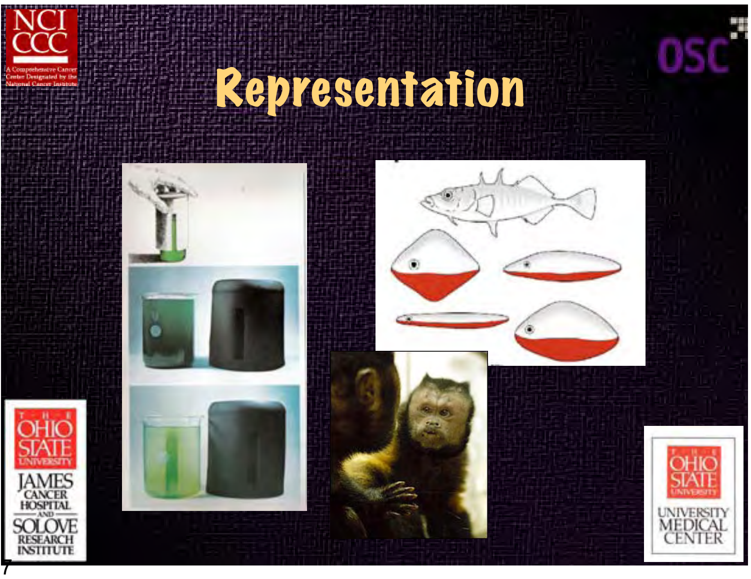
Representation

Independent neural systems for form, color, and motion
(Livingstone, Hubel, et. al.)

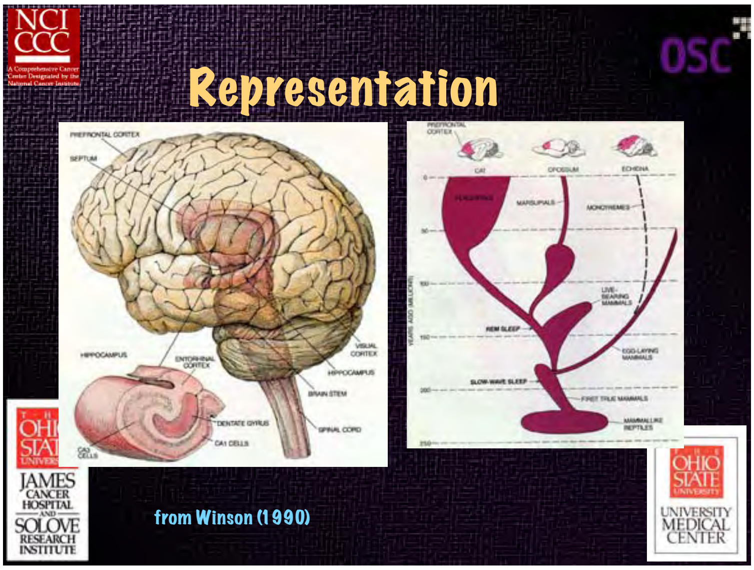
Parvocellular
interblob = form/detail
blob) = color
Magnocellular = motion
Stereopsis - mediated by all three

... and subsequent connectivity to other associative areas

Mirror cells discovered in monkeys
(Rizzollati et al,)



Representation



Representation

from Winson (1990)

Representation

The same neural mechanisms are used during memory processing under general experience

Dreams are the means by which animals form strategies for survival and evaluate current experience in light of those strategies (Winson 1990)

"We conclude that consciousness depends upon the microclimate of neuronal networks throughout the forebrain."
(Hobson & Stickgold in Gazzaniga)

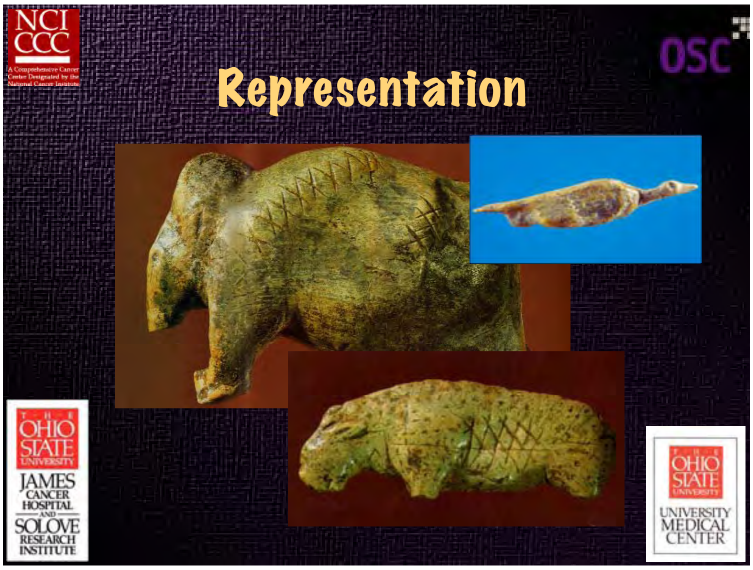
Representation

Emergence of symbolic representation in humans

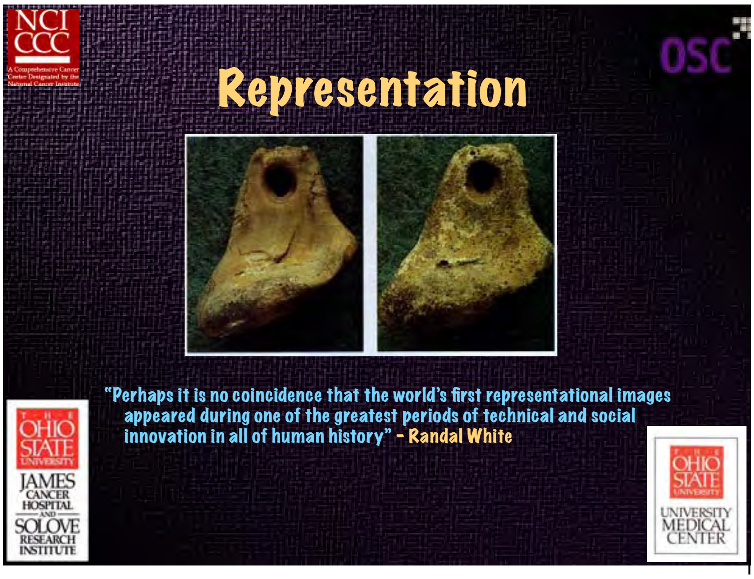
2.5 million years ago - purely utilitarian
40,000 years ago - Aurignacian Period - sudden appearance of objects with intentional decorative or aesthetic purpose

Cultural processes linked to an increase in symbolic form (White)

Perhaps expansion (qualitative/quantitative) in Mirror Cells (Ramachandran)

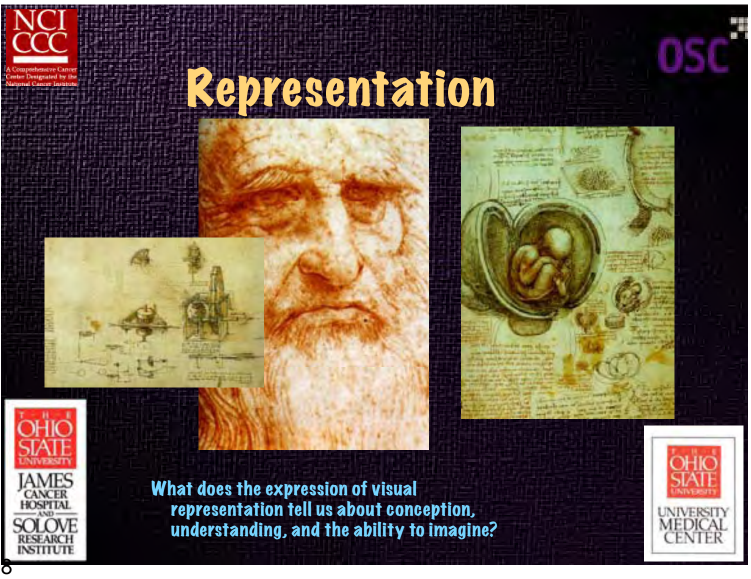


Representation



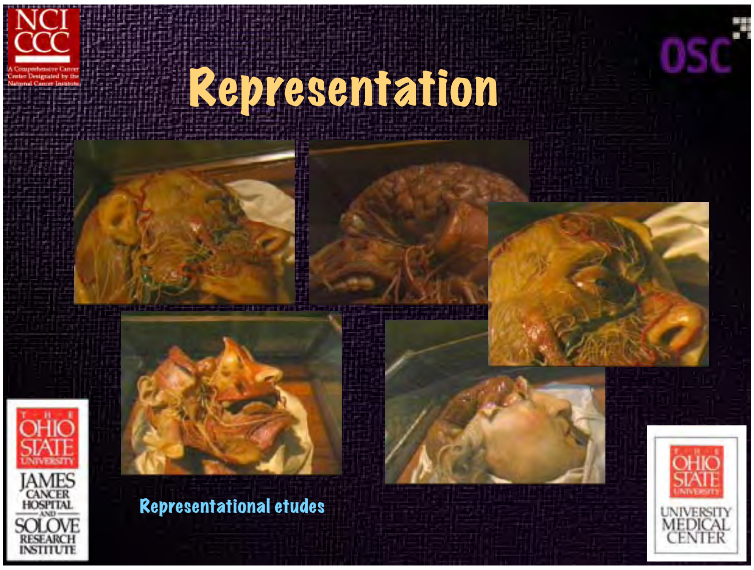
Representation

"Perhaps it is no coincidence that the world's first representational images appeared during one of the greatest periods of technical and social innovation in all of human history" - Randal White



Representation

What does the expression of visual representation tell us about conception, understanding, and the ability to imagine?

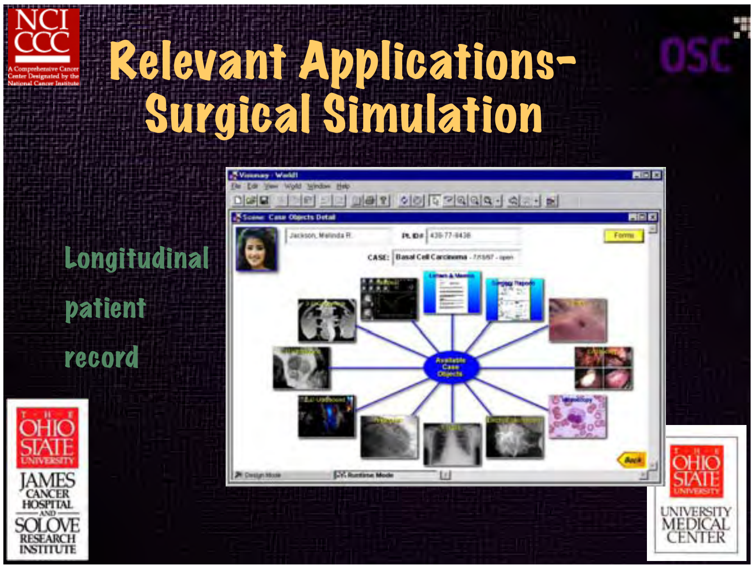


Representation

Representational etudes

Representation

VISUALIZATION IN SCIENTIFIC COMPUTING
ALL BUT WAR IS SIMULATION



Relevant Applications- Surgical Simulation

Longitudinal patient record

Visionary - World1

Scene: Case Objects Detail

Jackson, Melinda R.

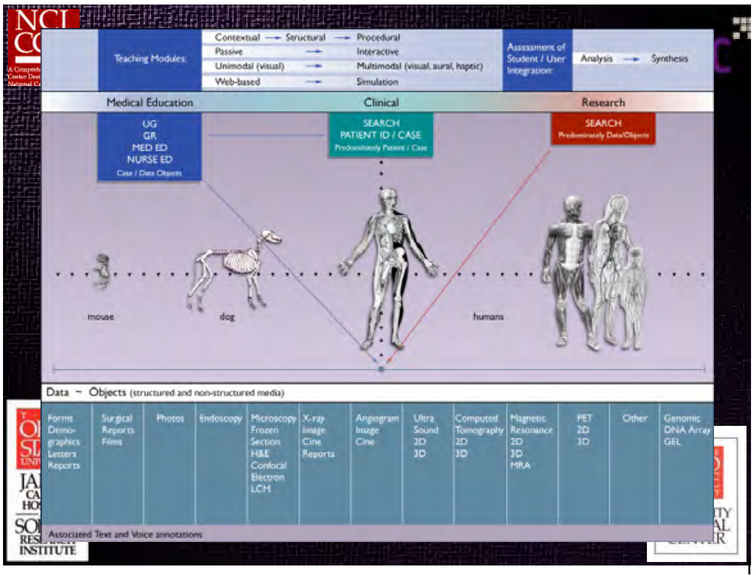
Pt. ID# 430-77-9438

CASE: Basal Cell Carcinoma - 7/15/97 - open

Relevant Applications- Surgical Simulation

E-Science - large-scale science that studies very complex micro to macro-scale problems over time and space

NSF CISE Grand Challenges in E-Science Workshop Report, April 2002



Teaching Modules: Contextual → Structural → Procedural; Passive → Interactive; Unimodal (visual) → Multimodal (visual, aural, haptic); Web-based → Simulation

Assessment of Student / User Integration: Analysis → Synthesis

Medical Education — Clinical — Research

UG
GR
MED ED
NURSE ED
Case / Data Objects

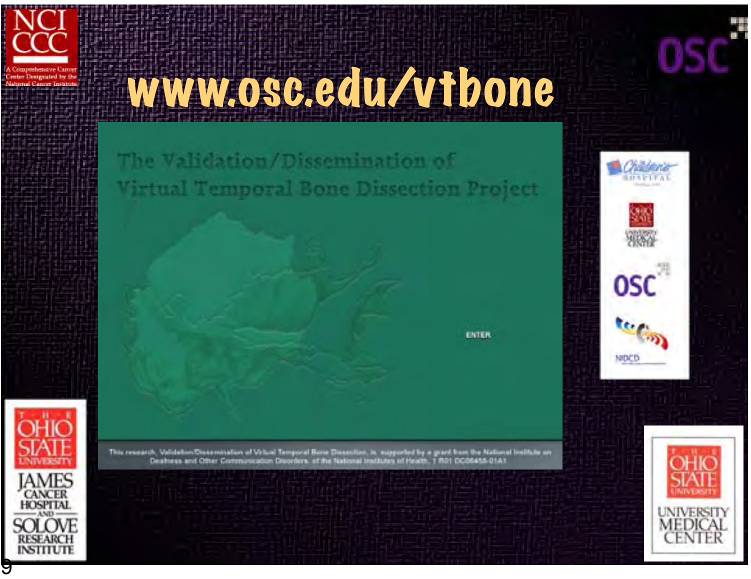
SEARCH PATIENT ID / CASE Predominately Patient / Case

SEARCH Predominately Data/Objects

Data ~ Objects (structured and non-structured media)

Forms Demo-graphics Letters Reports	Surgical Reports Films	Photos	Endoscopy	Microscopy Frozen Section H&E Confocal Electron LCM	X-ray Image Cine Reports	Angiogram Image Cine	Ultra Sound 2D 3D	Computed Tomography 2D 3D	Magnetic Resonance 2D 3D MRA	PET 2D 3D	Other	Genomic DNA Array GEL

Associated Text and Voice annotations



www.osc.edu/vtbone

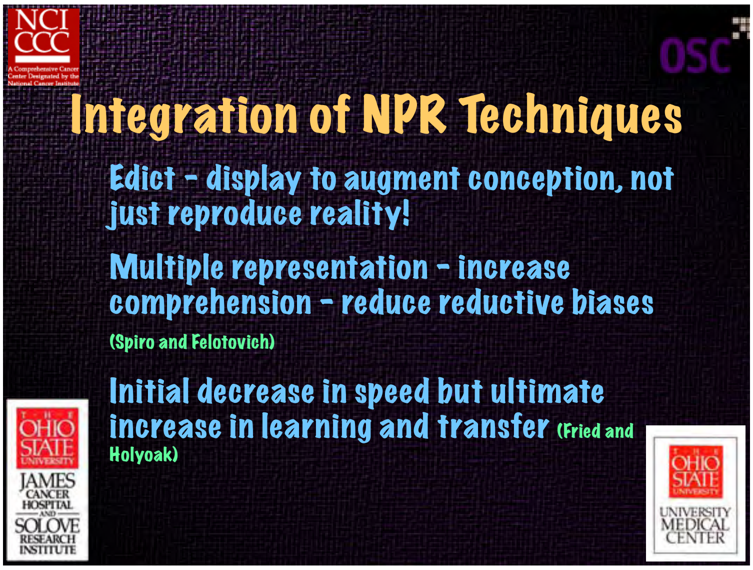
The Validation/Dissemination of Virtual Temporal Bone Dissection Project

ENTER

This research, Validation/Dissemination of Virtual Temporal Bone Dissection, is supported by a grant from the National Institute on Deafness and Other Communication Disorders, of the National Institutes of Health, 1 R01 DC06458-01A1

Issues in Simulation

Temporal Bone Dissection Simulation

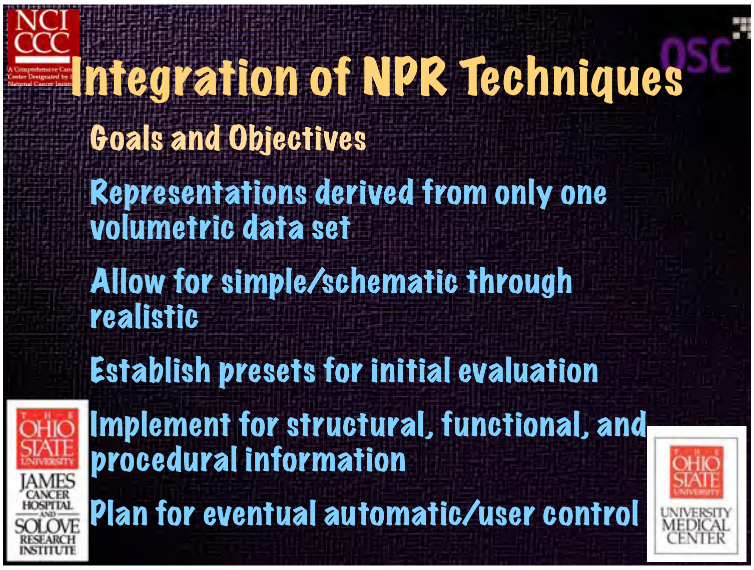


Integration of NPR Techniques

Edict - display to augment conception, not just reproduce reality!

Multiple representation - increase comprehension - reduce reductive biases (Spiro and Felotovich)

Initial decrease in speed but ultimate increase in learning and transfer (Fried and Holyoak)



Integration of NPR Techniques

Goals and Objectives

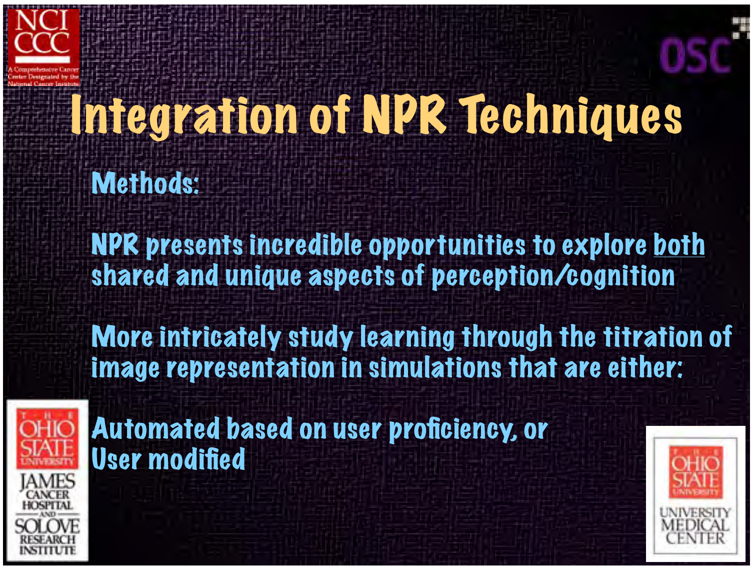
Representations derived from only one volumetric data set

Allow for simple/schematic through realistic

Establish presets for initial evaluation

Implement for structural, functional, and procedural information

Plan for eventual automatic/user control



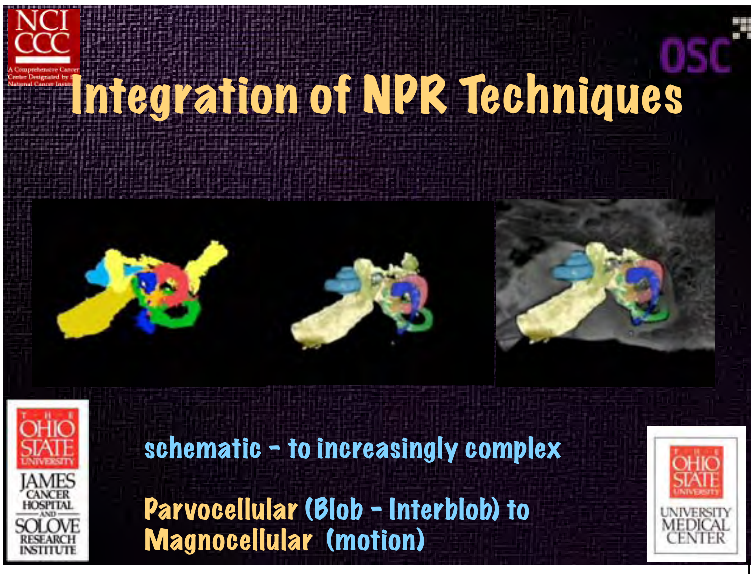
Integration of NPR Techniques

Methods:

NPR presents incredible opportunities to explore both shared and unique aspects of perception/cognition

More intricately study learning through the titration of image representation in simulations that are either:

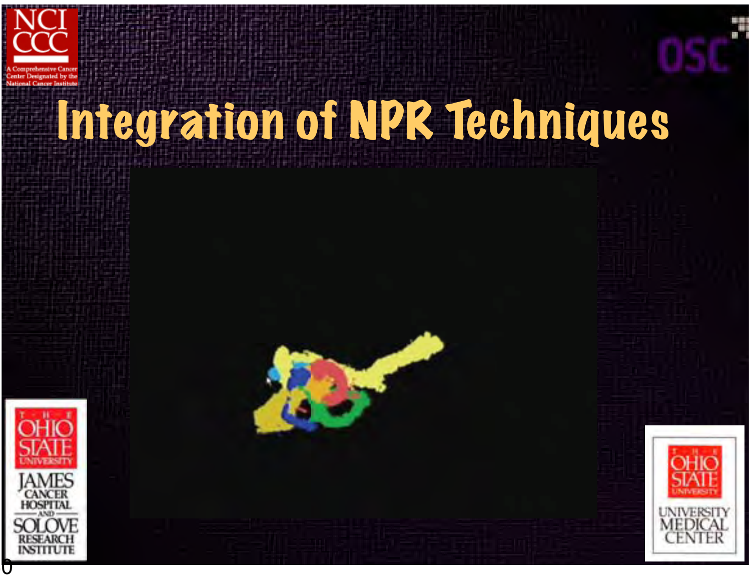
Automated based on user proficiency, or
User modified



Integration of NPR Techniques

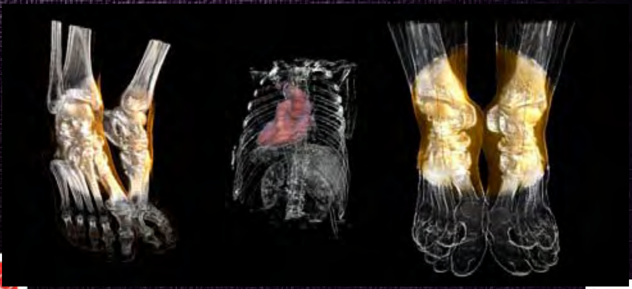
schematic - to increasingly complex

Parvocellular (Blob - Interblob) to Magnocellular (motion)



Integration of NPR Techniques

Integration of NPR Techniques



PURPL
Purdue University Rendering & Perceptualization Lab

Aesthetic Relevance

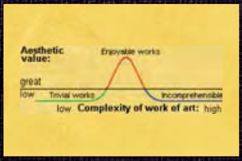
Facilitate natural inquiry at user's pace

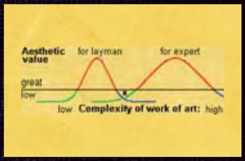
Through user synthesis -

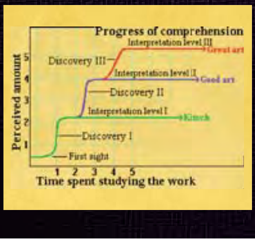
more qualitative understanding of the user's comprehension

provide a more critical assessment of the constructive proficiency of the user

Aesthetic Relevance







Aesthetic Relevance

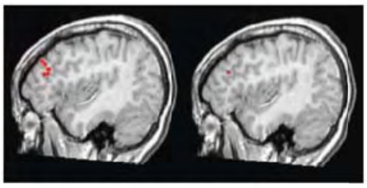


Fig. 1. Activation of the PDC under stimuli qualified as beautiful by participant (*Left*) and under stimuli qualified as not beautiful (*Right*).

Cela-Conde *et al.*

from Cele-Conde et al. (2004)

Acknowledgements

Portions of this research are funded under a grant from the National Institutes of Health, The National Institute on Deafness and Other Communicative Disorders

1 R01 DC06458-01A1

Cultivating Adoption & Creating Diverse Teams

Understand the agenda - current practice and vision

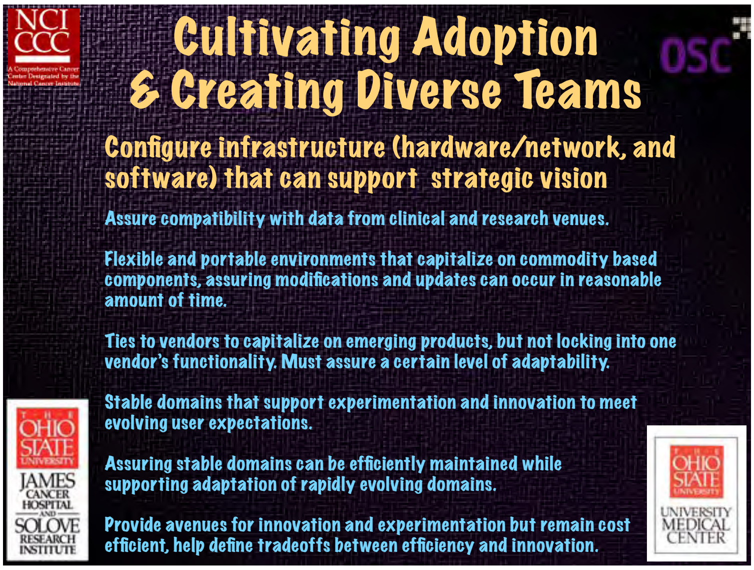
Assist in defining strategic vision to develop and integrate the use of modeling and simulation and maintain an environment that supports early adoption of technologies.

Canvas user community to determine variance of expectations.

Determine and exploit homogeneities that exist between various content domains.

Evaluate curriculum to determine best initial approach to amortize early developments.

Help define standards that can be readily implemented, but that do not suppress innovation.



Cultivating Adoption & Creating Diverse Teams

Configure infrastructure (hardware/network, and software) that can support strategic vision

Assure compatibility with data from clinical and research venues.

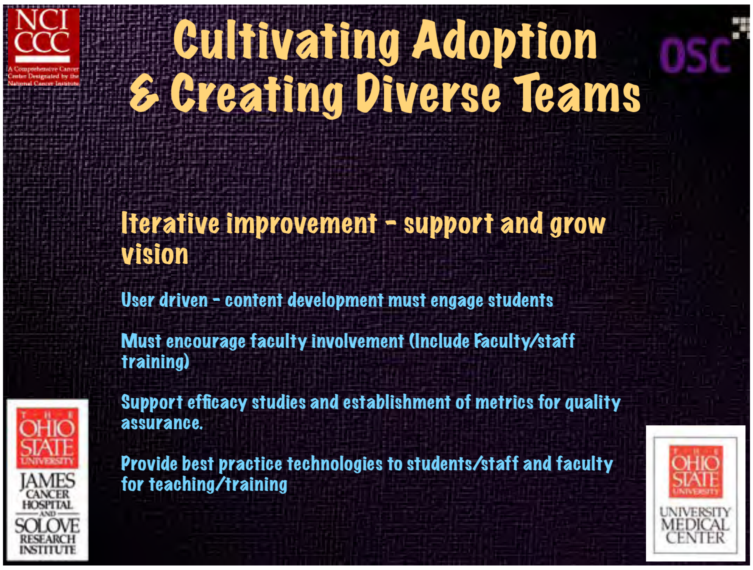
Flexible and portable environments that capitalize on commodity based components, assuring modifications and updates can occur in reasonable amount of time.

Ties to vendors to capitalize on emerging products, but not locking into one vendor's functionality. Must assure a certain level of adaptability.

Stable domains that support experimentation and innovation to meet evolving user expectations.

Assuring stable domains can be efficiently maintained while supporting adaptation of rapidly evolving domains.

Provide avenues for innovation and experimentation but remain cost efficient, help define tradeoffs between efficiency and innovation.



Cultivating Adoption & Creating Diverse Teams

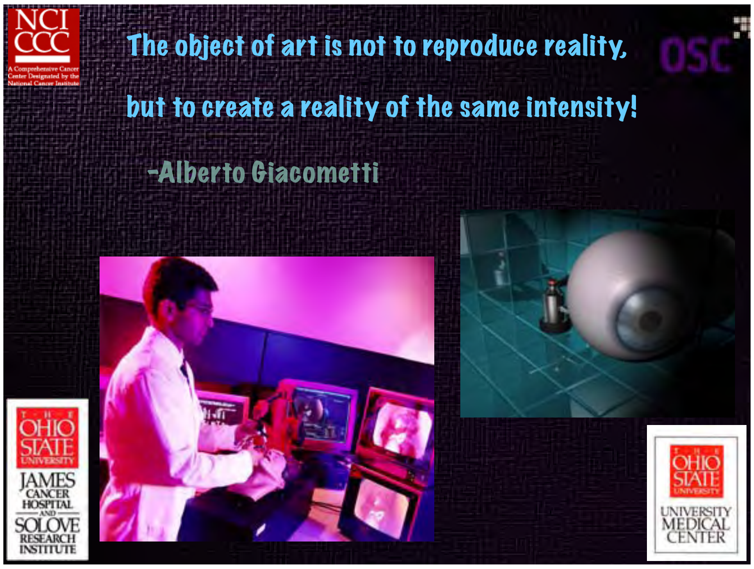
Iterative improvement - support and grow vision

User driven - content development must engage students

Must encourage faculty involvement (Include Faculty/staff training)

Support efficacy studies and establishment of metrics for quality assurance.

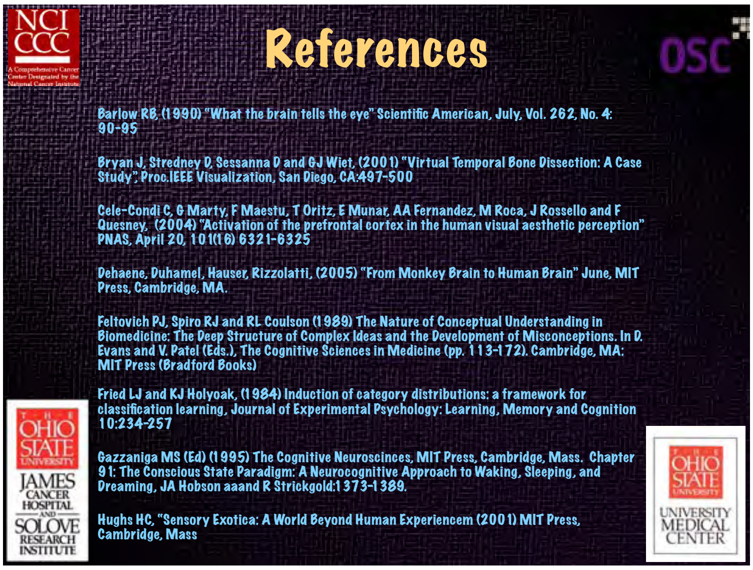
Provide best practice technologies to students/staff and faculty for teaching/training



The object of art is not to reproduce reality,

but to create a reality of the same intensity!

-Alberto Giacometti



References

Barlow RB, (1990) "What the brain tells the eye" Scientific American, July, Vol. 262, No. 4: 90-95

Bryan J, Stredney D, Sessanna D and GJ Wiet, (2001) "Virtual Temporal Bone Dissection: A Case Study", Proc.IEEE Visualization, San Diego, CA:497-500

Cele-Condi C, G Marty, F Maestu, T Oritz, E Munar, AA Fernandez, M Roca, J Rossello and F Quesney, (2004) "Activation of the prefrontal cortex in the human visual aesthetic perception" PNAS, April 20, 101(16) 6321-6325

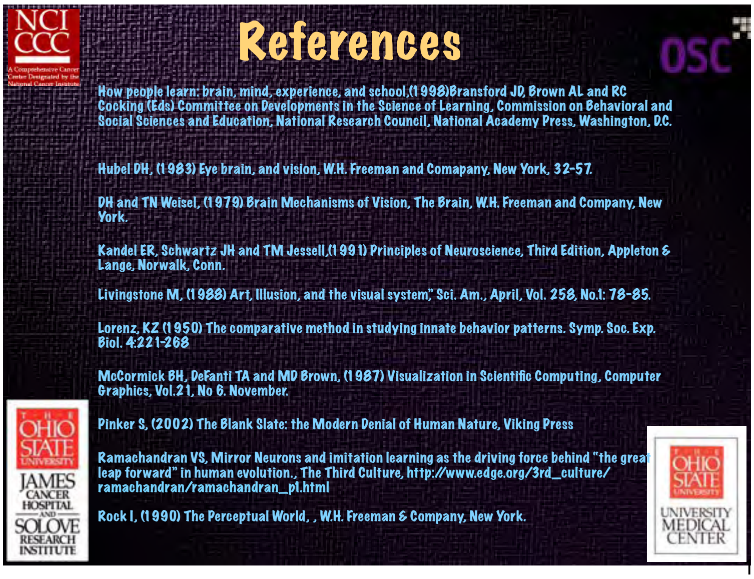
Dehaene, Duhamel, Hauser, Rizzolatti, (2005) "From Monkey Brain to Human Brain" June, MIT Press, Cambridge, MA.

Feltovich PJ, Spiro RJ and RL Coulson (1989) The Nature of Conceptual Understanding in Biomedicine: The Deep Structure of Complex Ideas and the Development of Misconceptions. In D. Evans and V. Patel (Eds.), The Cognitive Sciences in Medicine (pp. 113-172). Cambridge, MA: MIT Press (Bradford Books)

Fried LJ and KJ Holyoak, (1984) Induction of category distributions: a framework for classification learning, Journal of Experimental Psychology: Learning, Memory and Cognition 10:234-257

Gazzaniga MS (Ed) (1995) The Cognitive Neuroscinces, MIT Press, Cambridge, Mass. Chapter 91: The Conscious State Paradigm: A Neurocognitive Approach to Waking, Sleeping, and Dreaming, JA Hobson aaand R Strickgold:1373-1389.

Hughs HC, "Sensory Exotica: A World Beyond Human Experiencem (2001) MIT Press, Cambridge, Mass



References

How people learn: brain, mind, experience, and school,(1998)Bransford JD, Brown AL and RC Cocking (Eds) Committee on Developments in the Science of Learning, Commission on Behavioral and Social Sciences and Education, National Research Council, National Academy Press, Washington, D.C.

Hubel DH, (1983) Eye brain, and vision, W.H. Freeman and Comapany, New York, 32-57.

DH and TN Weisel, (1979) Brain Mechanisms of Vision, The Brain, W.H. Freeman and Company, New York.

Kandel ER, Schwartz JH and TM Jessell,(1991) Principles of Neuroscience, Third Edition, Appleton & Lange, Norwalk, Conn.

Livingstone M, (1988) Art, Illusion, and the visual system," Sci. Am., April, Vol. 258, No.1: 78-85.

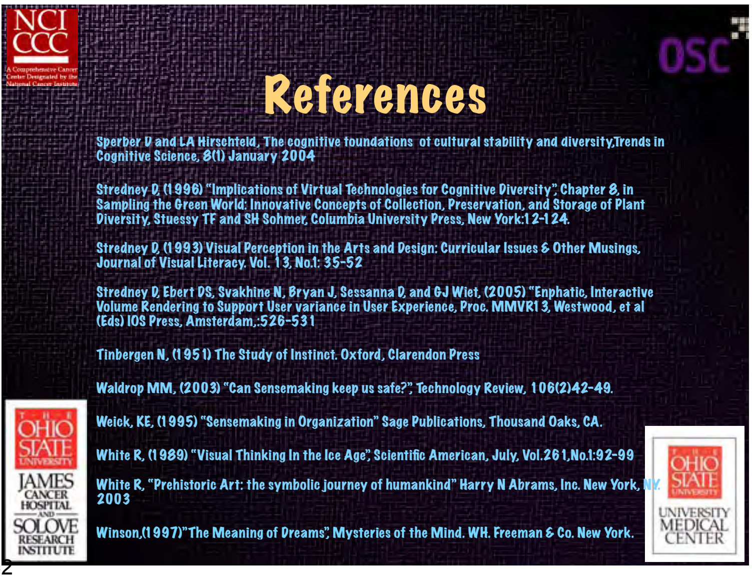
Lorenz, KZ (1950) The comparative method in studying innate behavior patterns. Symp. Soc. Exp. Biol. 4:221-268

McCormick BH, DeFanti TA and MD Brown, (1987) Visualization in Scientific Computing, Computer Graphics, Vol.21, No 6. November.

Pinker S, (2002) The Blank Slate: the Modern Denial of Human Nature, Viking Press

Ramachandran VS, Mirror Neurons and imitation learning as the driving force behind "the great leap forward" in human evolution., The Third Culture, http://www.edge.org/3rd_culture/ramachandran/ramachandran_p1.html

Rock I, (1990) The Perceptual World, , W.H. Freeman & Company, New York.



References

Sperber D and LA Hirschfeld, The cognitive foundations of cultural stability and diversity,Trends in Cognitive Science, 8(1) January 2004

Stredney D, (1996) "Implications of Virtual Technologies for Cognitive Diversity", Chapter 8, in Sampling the Green World: Innovative Concepts of Collection, Preservation, and Storage of Plant Diversity, Stuessy TF and SH Sohmer, Columbia University Press, New York:12-124.

Stredney D, (1993) Visual Perception in the Arts and Design: Curricular Issues & Other Musings, Journal of Visual Literacy. Vol. 13, No.1: 35-52

Stredney D, Ebert DS, Svakhine N, Bryan J, Sessanna D, and GJ Wiet, (2005) "Enphatic, Interactive Volume Rendering to Support User variance in User Experience, Proc. MMVR13, Westwood, et al (Eds) IOS Press, Amsterdam,:526-531

Tinbergen N, (1951) The Study of Instinct. Oxford, Clarendon Press

Waldrop MM, (2003) "Can Sensemaking keep us safe?", Technology Review, 106(2)42-49.

Weick, KE, (1995) "Sensemaking in Organization" Sage Publications, Thousand Oaks, CA.

White R, (1989) "Visual Thinking In the Ice Age", Scientific American, July, Vol.261,No.1:92-99

White R, "Prehistoric Art: the symbolic journey of humankind" Harry N Abrams, Inc. New York, NY. 2003

Winson,(1997)"The Meaning of Dreams", Mysteries of the Mind. WH. Freeman & Co. New York.

Illustrative Visualization for Surgical Planning

Illustrative Visualization for Surgical Planning

Bernhard Preim[1] Christian Tietjen[1] Arno Krüger[1]

[1] Department of Simulation and Graphics
Otto-von-Guericke-University of Magdeburg, Germany
{preim|tietjen|krueger}@isg.cs.uni-magdeburg.de

Keywords: Medical visualization, illustrative rendering, neck dissection, operation planning

1. Introduction

Visualizations are generated for a certain purpose. In medical applications this purpose is often a diagnostic question or a therapy planning scenario. In these scenarios, it is essential to adapt the appearance of objects or regions to their relevance for the specific task. As an example, it is often useful to focus on certain anatomic structures whereas other objects or regions only serve as orientation aid which might be displayed less pronounced. A medical visualization system might "know" what is relevant, after the user selected an object either immediately within the visualization or indirectly via its name in a list. Emphasis techniques modify the selected object or other objects such that its shape can be clearly recognized and its location in the overall model becomes obvious.

Most of the techniques discussed here require segmentation information concerning relevant objects. As a family of visualization techniques suitable for emphasis in medical visualization, we discuss so-called non-photorealistic rendering techniques where points and lines are employed to display and augment objects. These are inspired by traditional illustration techniques in scientific and medical applications. The potential of these techniques is to emphasize important features and to remove extraneous detail [Hod89]. Among the large variety of possible visualization techniques, those are preferred which are recognized at a first glance (*preattentive vision*, [Tre85]). Research in visual perception indicates that there are visualization parameters which are recognized without attention. As an example, objects shown with highly saturated colors are recognized immediately. Another effective focussing technique is blurring where only important objects are rendered sharply whereas others appear blurred, similar to blurred regions in photographies [KMH02].

2. Illustrative Surface Rendering

For a long time computer graphics has been focussed on photorealistic rendering. It was the goal to compute an image from a description of the geometry by simulating optical effects such as reflection, absorption and refraction as closely as possible. In 1990, a new direction emerged and meanwhile gained much acceptance - non-photorealistic rendering (NPR). The goal here is to provide a wider range of rendering techniques to express various effects and to simulate styles from traditional scientific and medical illustration. The term "non-photorealistic" is widespread in computer graphics although it is not expressive. Due to the inspiration from traditional illustrations, these methods are also called *illustrative rendering*. According to our experience, illustrative rendering is a better term for the communication between computer scientists and medical doctors.

Directing attention to relevant features, on the one hand, and omitting unimportant details on the other hand may be achieved by rendering strokes and points instead of shading surfaces. Illustrative rendering also provides facilities to expose features which have been obscured. This is essential in the context of emphasis techniques.

The pioneering work [ST90] was entitled "Comprehensible rendering of 3D shapes". Silhouette and feature lines were generated to emphasize the shape of objects, and hatchings were employed to convey the curvature and the texture of objects. Also, discontinuities in the depth-buffer were analyzed and visualized by means of lines. Silhouettes are essential in shape recognition because they provide cues for figure-to-ground distinction. However, since they are view dependent, they need to be determined for every viewing direction [IFH*03].

It is interesting and worth wile to note, that the goals of scientific visualization and illustrative rendering are very similar: to convey information effectively and to emphasize features in the data. Therefore, it is not surprising that illustrative rendering techniques have been adopted in visualization in general and in medical visualization in particular [NSW02].

Shape perception. Psychological studies clearly revealed that silhouette and hatching lines might improve the comprehensibility of images. As an example, Kim et al. investigated the effect of textured lines superimposed on shaded surfaces [KHSI03b] [KHSI03a]. In their study, users had the task to estimate surface normals. It turned out that the 3d shape was better perceived with hatching lines in the direction of maximum curvature. Slightly improved results were achieved with hatching lines in two directions. These and other studies reveal that texture may improve shape perception. While illustrative rendering has many subareas and applications, such as artistic applications or games, we focus on techniques to improve the perception of anatomic and pathologic structures. The perception improvement of this structures is essential for surgical planning. Illustrative techniques have the potential to convey complex information, such as anatomic and functional information.

2.1. Emphasis and Illustrative Rendering

Illustrative rendering provides a wide range of techniques which might be employed for emphasis purposes . In photorealistic rendering, emphasis might be achieved by adapting the position of the virtual camera or by placing a spot-light source. The rendering process itself, however, regards all edges and faces as similarly important. Nothing is left blank even if it is less relevant. Partial visibility of an outer object to reveal inner structures can only be achieved by semitransparent rendering. This method, however, strongly degrades shape perception.

Illustrative rendering offers more degrees of freedom to emphasize objects or regions. Important objects might be enhanced by silhouette and hatching lines, while others are not. As has been pointed out by [VKG04], illustrative rendering permits *sparse visual representations* of objects which consume less screen space than shaded surfaces. Outlines or silhouettes are probably the sparsest meaningful visualization which allows to roughly understand the object shape. The addition of prominent *feature lines* or hatching lines leads to a denser representation which reveals more detail on the object shape. Finally, the combination of such illustration techniques with conventional rendering techniques represents a dense representation which depicts an object clearly at the expense of obstructed objects behind.

The drawback of this freedom is that good choices for many parameters are required. More degrees of freedom make it more difficult to adjust a visualization. While artists may take hours to produce expressive images, surgical planning is carried out under time-pressure. Also, medical visualizations should be precise and reliable. Our view on illustrative rendering and its potential is focussed on these aspects. Techniques which require considerable and non-trivial input by the user (for example specification of hatching directions) are not considered. Also, rendering styles which are more artistic than precise are omitted. Browsing through anatomic illustrations and surgical textbooks gives an idea about useful rendering styles for medical education and therapy planning. Silhouettes and points are frequently used to convey the shape of objects. The local density of points is adapted to the curvature of objects and simulates a lighting effect. This rendering style is called *"stippling"*. Small dots of ink are placed onto paper such that their density gives the impression of tone. Besides their expressive power, stippling is attractive since the rendering of points is very fast and facilitates an interactive exploration.

2.2. Silhouette and Feature Lines from Surface Models

Surface models in medical visualization are generated by thresholding medical volume data or by transforming segmentation information into (polygonal) surfaces. For continuous objects, such as B-Spline surfaces, the silhouette S is defined as the set of points on the object's surface where the surface normal is perpendicular to the vector from the viewpoint [HZ00]. At these points p_i, the dot product of the normal n_i with the view vector is zero (Eq. 1):

$$\{S\} = \{P | n_i \Delta (p_i - c) = 0\} \tag{1}$$

with c being the camera position (for perspective projections).

For polygonal models, the definition above cannot be directly applied because normals are only defined for faces and not for arbitrary points. However, silhouettes can be interpolated along edges in a polygonal model that lie on the border between changes of surface visibility. Thus, silhouette edges of a polygonal model are edges that share a front- and a back-facing polygon.

Other significant lines are creases, which are defined by comparing the angle between its two adjacent polygons. If this angle is above a certain threshold the common edge represents a crease. Creases on smooth surfaces are also called *crest lines*. They represent ridges as well as valleys on such surfaces. Together with some other lines considered as important, creases are also referred to as *feature lines*.

Silhouette algorithms solve two tasks: They determine silhouette edges and determine the visible subset of them. In general, *image-* and *object-space* methods are available for these tasks (see [IFH*03] for a recent survey).

Image-space methods operate on image buffers which contain per pixel information. In particular, the z-buffer and the normal-buffer (representing the z-coordinate and the normal of the polygon rendered at each pixel) are useful to find silhouette edges. Strong discontinuities in these buffers indicate the borders between objects and thus silhouettes. Edge detection methods from conventional image processing are employed for this purpose (recall [HZ00]). Image-space methods efficiently compute silhouettes because graphics hardware may be exploited. Fig. 1 illustrates the use of images-space silhouettes and feature lines to convey the shape of context objects. Object-space methods, on the other hand, analyze the 3d model and produce an analytical description of the silhouette which is essential if silhouette lines have to be flexibly parameterized.

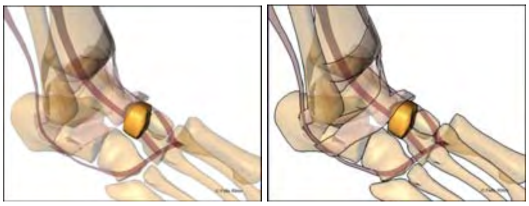


Figure 1: *An anatomic illustration with emphasis on a user-selected bone. The context objects are rendered strongly transparent (left). On the right, transparent objects are enhanced with silhouette and feature lines to better convey their shape. (Courtesy of Felix Ritter, MeVis Bremen).*

Subpolygon silhouettes. In medical visualization (flat) polygonal meshes usually approximate curved anatomic structures with a smooth appearance. Therefore, silhouettes of polygonal meshes can differ obviously from the silhouette that an algorithm based on polygonal edges yields. Therefore, methods have been developed to determine silhouettes more precisely.

As an example, [HZ00] consider the silhouette of a free-form surface approximated by a polygonal mesh. To find this silhouette, they recompute the normal vectors of the approximated free-form surface at the vertices of the polygonal mesh. Using this normal, they compute its dot product with the respective viewing direction. Then, for every edge where the vertices have opposite polarity, they use linear interpolation along this edge to calculate the point where the surface normal is zero. The connection of these points yields a piecewise linear subpolygon silhouette line which probably has fewer artifacts and is closer to the real silhouette.

Data structures for silhouette and feature line determination. For the efficient extraction of lines and strokes from 3d meshes, it is essential to use a data structure which provides local connectivity information. For conventional surface rendering, it is sufficient to represent the affiliation of vertices to polygons. For silhouette and feature line determination, it is essential to access adjacent polygons (polygons which have a common edge or a common vertex). This information is represented for example in a Winged Edge data structure [Bau72] which is useful for object-oriented rendering methods. As the name suggests, it is based on an (global) edge list and consists of locally linked edges. For every object, the data structure additionally stores a list of polygons and vertices. As an example, the winged-edge data structure allows to select adjacent polygons with strong deviation in the surface normal efficiently. This data structure is created as a preprocessing step.

Suggestive contours. An interesting extension of the more traditional silhouettes are so-called *suggestive contours* introduced by [DFRS03]. They include contours derived from adjacent viewpoints to generate expressive renderings which convey complex shapes with concavities convincingly. Suggestive contours are computed based on Eq. 1. Instead of drawing pixels if the dot product of surface normal and view vector is zero; pixels are drawn if the dot product represents a local minimum.

An essential advantage of suggestive contours is their temporal coherence. While conventional silhouettes may strongly change after small rotations, suggestive contours are more constant [DFR04]. Temporal coherence is an advantage for animations as well as for interactive 3d renderings. Suggestive contours have not been used for surgical planning so far.

2.3. Hatching Surface Models

Feature lines may effectively convey prominent features of a surface, such as ridges and valleys in the shape of the human brain. In the absence of such prominent features they are not applicable. The surface of organs, for example, has only a very few landmarks which might be emphasized with feature lines. In particular, for such smooth objects, hatching may convey shape information. Hatching techniques support a continuous perception of a surface encompassing rapidly changing as well as relatively constant areas.

Hatching may be utilized in isolation or in combination with surface rendering, in particular with strongly transparent surfaces. Strongly transparent surfaces are often used in surgical planning to show outer structures such as organs and inner structures such as vasculature or pathology simultaneously. The drawback of this strategy is that most of the depth-cues to convey shape have a minimal effect (at best) for transparent surfaces. This is well-known in psychophysics as well as in computer graphics and visualization [IFP96].

The challenge however is to develop algorithms for the optimal placement and scaling of hatching lines and strokes to convey shape information best. While hatching techniques designed for artists rely on many parameters which have to be supplied by the user, in surgical planning, an easy or even automatic parametrization is essential. Artists have recognized that the direction of strokes has a strong influence on our perception of surfaces. A uniform direction is not preferable as shapes tend to be perceived flattened. [IFP96] found that medical illustrators often use the curvature of surfaces to guide hatching: strokes are oriented along the strongest curvature. They described a viable approach for defining principal curvature directions by approximating partial derivatives based on finite differences of voxel values [IFP95]. The driving application for their work is radiation treatment planning where isosurfaces represent equal intensity of radiation dose. For treatment planning, the radiation beams and several of the surfaces are shown as transparent isosurfaces together with the tumor which should be destroyed, is rendered opaque. Hatching techniques enhance the interpretation of the transparent isointensity surfaces. This work started in 1990 [LFP*90] and represents the first application of illustrative rendering techniques in medical visualization.

2.4. Reconstruction of Surfaces for Illustrative Rendering

Medical volume data consists of slices where the in-plane resolution is usually higher than the distance between slices. This is referred to as *anisotropic data*. A typical resolution is $0.8 \times 0.8 \times 4mm$. Segmentation results are usually stored as volume data where the values indicate which object the corresponding voxel belongs to - voxels belong either completely to a certain structure or not at all.

A surface may be generated by the Marching Cubes (MC) algorithm or one of its refinements. The MC algorithm involves linear interpolation to compute intersections of the surface with the volume data. If the MC algorithm is applied to anisotropic data, aliasing effects occur. These effects strongly hamper the expressiveness of illustrative renderings. This is a general problem for any kind of medical visualization, however, it is more serious in case of silhouette and feature line rendering since the artifacts appear pronounced (see Fig. 2, left).

To overcome this problem two strategies are possible:

- Interpolation of additional slices, and
- Smoothing surfaces

On the one hand, higher order interpolation techniques can be employed to compute in-between slices. Ideally, the number of interpolated slices is chosen such that an isotropic resolution arises. Cubic interpolation is an appropriate method to compute these additional slices. The drawback of this strategy is the considerable increase in memory consumption. In our example with a voxel size of $0.8 \times 0.8 \times 4mm$, the number of slices and thus the overall size of the dataset is increased by a factor of 5.

Smoothing segmentation results for illustrative rendering. A large variety of techniques exist to remove high frequency noise in polygonal models. These methods differ in their computational effort and in the quality of the resulting surfaces. Some of the smoothing techniques allow to fulfill certain constraints. For surgical planning, it is often desirable that structures do not shrink after smoothing. Therefore, volume conservation is an essential constraint. Usually, smoothing modifies the position of vertices. Alternatively, surface normals could be modified. Shaded surface visualizations benefit from the modification of surface normals, however for silhouette rendering modifying the surface normals does not matter.

A widespread and simple algorithm is Laplacian smoothing where each vertex is moved in the geometric center of its local neighborhood [YOB02]. The filter is applied iteratively and has a smoothness factor and the number of iterations as parameters. Laplacian smoothing does not prevent shrinkage. Other simple smoothing techniques are Gaussian and Median smoothing [Tau95a], [Tau95b].

Similar to image processing techniques used to reduce noise in 2d data, there are more advanced techniques which reduce noise but preserve features better. A family of these advanced techniques is based on the physical process of diffusion. In particular, anisotropic diffusion leads to excellent results [TWBO02]. Other advanced smoothing techniques which turned out to be useful for smoothing anatomic structures are described in [DMSB99] and [Kob00].

In Fig. 2 and Fig. 3, the effect of smoothing a surfaces on the resulting silhouettes and feature lines is shown. In both cases, the same filter was applied. However, the two parameters had to be selected differently to cope with the peculiarities of these structures.

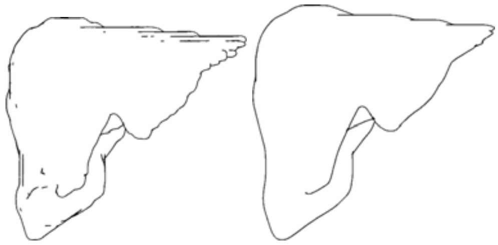


Figure 2: *Silhouette generation of the liver. Left: the isosurface representing the original segmentation result is employed. The resulting staircase artifacts are distracting and confusing since they have no anatomical basis. Right: the triangle mesh of the isosurface was smoothed with a relaxation filter.*

An essential aspect of smoothing is the assessment of the result. In a qualitative sense, the surface should appear smooth. Quantitatively, the smoothed surface should be evaluated with respect to the distance to the original surface. Strong differences are not acceptable for clinical applications. The Hausdorff distance provides a worst case approx-

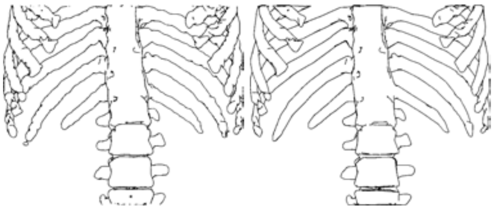


Figure 3: *Silhouette generation of skeletal structures. The unprocessed isosurface leads to noisy artifacts which can be corrected by appropriate smoothing.*

imation of how strongly the smoothed surface differs from the original one. A general problem is that different smoothing techniques are appropriate for different anatomic structures. Smoothing techniques which are appropriate for thin and elongated structures are different compared to those appropriate for compact larger structures, such as organs. In case of thin structures, special care is necessary to avoid topological changes (separation of a previously connected structure). Pathologic structures, such as lesions, again require different techniques. For surgical planning, it is essential that for each class of anatomic structure suitable default techniques and parameters are selected to reduce the interaction effort of the user.

3. Combining Line-, Surface-, and Volume Visualization

In surgical planning, illustrative techniques need to be combined with other - more conventional - rendering techniques. The combination of different rendering styles is a challenging problem. Hybrid surface and volume renderings can be generated e.g. by converting surface data to volume data (voxelization) and subsequently creating a combined volume rendering. The alternative approach is to apply a two-pass rendering where surfaces and volume data are rendered independently and are combined in a final step considering depth values of surface and volume data. Similarly, there are two different strategies, how line rendering can be integrated with surface and volume rendering.

1. Integration of lines and surfaces in the volume rendering process.
2. Application of different rendering passes to render lines, surfaces and volume data.

Integration of lines and surfaces in the volume rendering process. This strategy is computationally faster and easier to implement. An example for this approach is given in [VKG04].

Combining different rendering passes. The second strategy leads to an object-space method where line strips are considered as individual graphics primitives which might be parameterized flexibly. This strategy has been suggested by [NSW02]. This strategy is also realized in [TIP05] where the geometry and appearance of lines, surfaces and a volume rendering are integrated in an extended OpenInventor-scene graph. This object-space approach, however, poses some technical problems to achieve a correct rendering with respect to the depth-order of lines, surfaces and volume data. In a scene graph representation, the ordering of nodes in the graph is essential for a correct rendering.

3.1. Hybrid Rendering with Object-Based Methods

In this subsection, we describe the object-space method in more detail. This description is based on [TIP05].

In order to use direct volume rendering (DVR), we integrate this rendering technique into the scene graph. This is achieved by a specialized DVR node coupled with a transfer function (TF) node. However, since DVR always renders the whole volume into the z-buffer, it is not possible to add surface shading afterwards. Thus, the sequence in which the modules for the individual rendering techniques are added to the rendering pipeline is important.

For surface shading the OpenInventor architecture relies on normal z-buffer rendering. Thus, no special order of nodes is required for a correct rendering of the scene. Due to the nature of the remaining two techniques, the resulting scene graph may get fairly complex.

Direct volume rendering may be combined with surface rendering by adding the respective nodes to the scene graph. However, the DVR node fills the z-buffer for the entire volume that is rendered regardless of the opacity transfer function as explained above. Therefore, it has to be added to the scene graph after the surface shading has been completed, i.e., as the last node in the scene graph traversal. Hence, after the DVR, the z-buffer contains no more sensible depth information.

For the efficient extraction of lines and strokes from 3d meshes, a Winged Edge data structure is employed (recall Sect. 2.2). In order to render and stylize these objects, each object's mesh data is subsequently used to extract the lines and stylize them. The stylization process is divided into several steps represented by nodes in the scene graph. This concept also allows the reuse of certain line stylization pipelines because the respective nodes may be linked into the scene graph at several positions. This ensures a coherent appearance of the objects that use the same pipeline.

The object-space silhouette rendering approach that we employ comprises three steps: geometry data preparation, line extraction and stroke generation, and stylization.

After extracting the significant edges (silhouettes and feature lines), these edges are assembled into strokes and are processed in a stroke pipeline for stylization. As an essential step, this involves that hidden lines are removed. A fast and simple method for hidden line removal might be accomplished in the following way:

1. Objects are rendered into the z-buffer (while the frame buffer remains unchanged).
2. In a second step, all extracted lines are scan-converted individually and stepwise classified as hidden or visible using the previously generated z-buffer data [IHS02].

Then, stylization may be applied such as changing the stroke's width, dashing and color. The edge extraction and stroke generation is thus independent from the final stroke rendering. If surface rendering is used in addition to the line graphics, the surface objects have to be rendered into the final image prior to the lines. However, this approach is only applicable for opaque objects because transparent objects do not change the z-buffer. Thus, lines that lie behind a transparent object would not be removed. In order to prevent that distant lines are rendered on top of closer transparent model parts, the z-buffer rendering must be carried out for the transparent objects as well. The steps of the process are outlined in Figure 4.

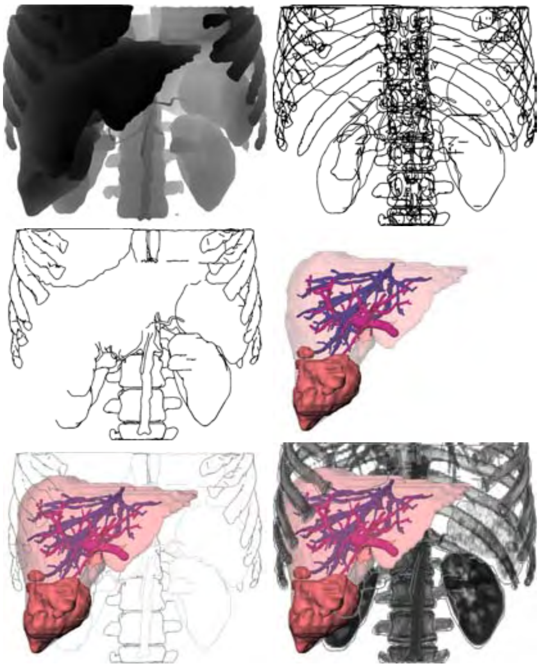


Figure 4: *Sequence for combining all rendering styles. (From: [TIP05]).*

3.2. Emphasis with Hybrid Visualizations

The different rendering styles are not equally well suited to visualize objects. Thus, they will be used to depict different structures according to their relevance for the visualization. With respect to relevance, in the following we will refer to three types of structures or objects:

Focus objects (FO): objects in the center of interest are emphasized in a particular way.

Near focus objects (NFO): important objects for the understanding of the functional interrelation or spatial location. Their visualization depends on the particular problem.

Context objects (CO): all other objects.

The combination of different rendering styles provides facilities to adapt a medical visualization to the importance of anatomic and pathologic structures. Illustrative rendering might be used to depict anatomic context, either as a single rendering mode or in combination with strongly transparent surface rendering. As an alternative, volume rendering might be employed for the anatomic context. Surface rendering and high opacity might be used for FOs. In Fig. 5 and 6, we show some combinations of rendering styles designed for a medical education system. Focus objects are the liver and the intrahepatic structures; the skeletal structures serve as anatomic context.

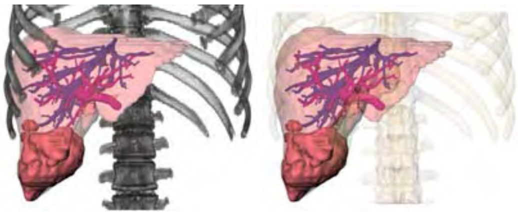


Figure 5: *Important structures as surface rendering. Context shown as direct volume rendering (left) and as strongly transparent surfaces (right). (From: [TIP05]).*

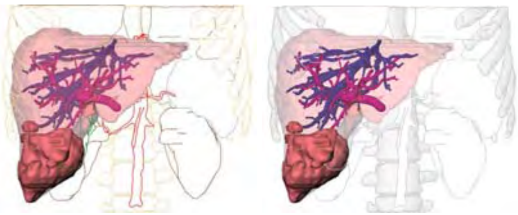


Figure 6: *Important structures as surface rendering. Context displayed exclusively with silhouettes (left) and as strongly transparent surfaces combined with silhouette rendering (right). (From: [TIP05]).*

A relevant yet difficult question concerns the appropriateness of the images in Fig. 5 and 6. Many variants have been discussed with surgeons and the four images in Fig. 5 and 6 have been regarded as suitable. But is there any best image? Probably, different viewers strongly differ in their choice. As a consequence, it should be easy to generate such images and to save parameters as individually preferred style which is reused for other images.

Line stylization may be used to control whether objects are conceived as FO or CO. One type of stylization is color which is widely used to emphasize objects. An interesting type of stylization refers to the visibility of lines. Hidden and visible lines may be depicted in a different manner to convey the spatial relationships (see Fig. 7). This is only feasible with the object-space approach to the visualization of lines.

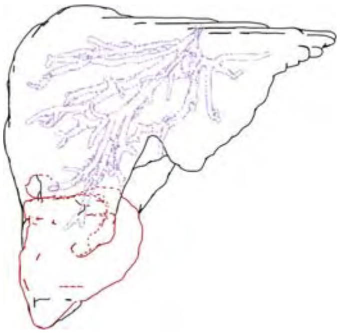


Figure 7: *Hidden feature lines are rendered as dashed lines. With this technique additional information concerning a tumor and the intrahepatic vasculature are included in a visualization.*

Default values. Hybrid visualizations consisting of lines, surfaces and volume data provide excellent facilities for fine-tuning visualizations of medical volume data. However, the huge space of possible renderings is due to many parameters. To make hybrid visualizations feasible, it is essential to carefully select appropriate default values for the visualization of categories of anatomic and pathologic structures, such as organs, lesions or vascular structures. Users should be able to modify these values, but in general the interaction effort should be reduced.

4. Emphasis with Illustrative Rendering

With conventional local emphasis techniques, the currently selected object (CSO) is emphasized with an appropriate color or with additional lines superimposed or by placing symbols which direct to the CSO. Local techniques might also be successful if parts of the CSO are hidden. If, however, larger portions or the CSO as a whole are hidden, local techniques are insufficient. In this section, we discuss cutaway and ghost views which ensure the visibility of the CSO. We also discuss regional techniques to color selection; in particular, we consider contrasts between the CSO and adjoined objects.

4.1. Cutaway and Ghost Views

An useful illustration technique is the modification of occluding regions to make an important object visible. These regions are removed or at least shown strongly transparent. Such visualizations are called cutaway illustrations.

A slightly different illustration technique is referred to as *ghost view*. In these views, a region is shown semitransparently to reveal hidden structures. The generation of cutaway and ghost views is very similar since in both cases a cut region has to be defined and considered. Compared to ghostviews cutaway views lead to a sharp contrast between foreground and background objects. Thus, ambiguities with respect to spatial ordering are avoided [DWE03]. Cutaway views have been introduced by [FS92] in computer-based illustration systems with applications in maintenance.

To indicate that an illustration technique is applied, the shape of the regions should differ strongly from the shape of anatomic or pathologic structures. While technical illustrators often create zig-zag-shaped cutaway views , regular shapes such as prisms or cylinders are useful for medical visualization. Cutaway views may be generated in the context of volume rendering as well as in the context of surface rendering. Cutaways in volume rendering require to voxelize the clip geometry. Based on a voxel representation, volume rendering is modified such that voxels in the clip region are discarded.

Fig. 8 shows a ghostview based on a cylindrical cutaway in a visualization which emphasizes enlarged lymph nodes relevant for neck dissection planning (see Sect. 5 for a discussion of these surgical interventions). The circular cross-section is parallel to the view plane and located around the tumor. The cylinder height is chosen such that the lymph nodes become completely visible. Together, the circular clip region is scaled such that the tumor itself as well as a 5 mm margin in x-, y-, and z-direction is cut out. The borderline of the cut is depicted to be clearly recognizable.

Similar to other emphasis techniques, cutaway and ghost views have a limited applicability: Primarily small and compact objects may be emphasized with cutaway views. For objects with a branching structure and a large axis aligned bounding box this technique is less appropriate.

Cutaway views of small pathologic lesions may be combined with an interaction to systematically explore them. With a special key, for example the Tab-key, the user may step through emphasized visualizations of these objects. Additional information, such as size and volume, might be included for the currently selected object.

Realization of cutaway and ghost views. In the context of medical visualization (large models, up to 100,000 vertices per structure), an efficient approach is necessary to compute the cutaway region and to adapt it immediately to a changing viewing direction. A useful intermediate result is the convex hull $ch(o)$ of the object o which should be displayed. The convex hull of a 3d pointset P is the smallest convex polygon which contains all points of P. If the cut-region has a convex shape, for example a cylinder, the cut-region of the $ch(o)$ is the same as the cut-region of o itself. $ch(o)$ has considerably less vertices than o (e.g. 200 vs.

20,000) and may be further geometrically simplified since the accuracy requirements are not distinctive.

In a second step the convex hull is projected onto the viewing plane and the position and size of the cut shape in the viewing plane is determined. To reduce vertices again, the convex hull of the projection is calculated in 2d. In case of cylindrical cut-regions, an algorithm which determines minimally enclosing circles of a pointset (around 20 vertices) is applied. After changing the viewpoint, only the second step has to be carried out.

Convex hull determination is one of the most essential problems studied in computational geometry. For example [BKOS00] provide a chapter on convex hull algorithms. Minimally enclosing circle determination (or more general: minimally enclosing discs) are also discussed in computational geometry books [SE02, e.g.] and [BYB98]. The latter problem can be solved in $O(n)$ time which means that the computational effort only linearly increases with the number of points involved.

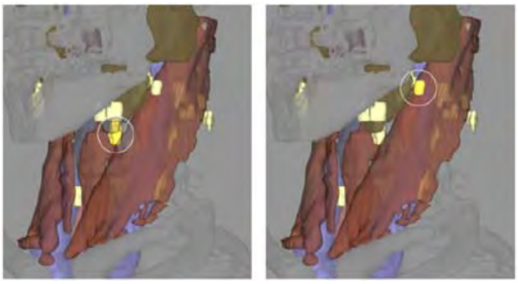


Figure 8: *Ghost view of enlarged lymph nodes for neck dissection planning. (From: [KTH*05]).*

Recently, cutaway views have been generalized to importance-driven volume rendering [VKG04]. Here, less important parts of the data are suppressed, for example by employing transparency. The general concept here is to transform *importance* to *visibility*. Importance-driven rendering is based on a visibility priority which is assigned to each object. It is thus more general, than an emphasis technique for the CSO only. Similar to cutaways, importance-driven rendering is most suitable to focus on smaller objects. If objects with high priorities occlude each other, the object with maximum priority is drawn (maximum importance projection). The straightforward application of this concept leads to a cylinder-shaped countersink for an important object. This variant suffers from similar problems as the maximum intensity projection where the depth relations are difficult to recognize.

5. Case Study: Neck Dissection Planning

In this section, we discuss how conventional and illustrative rendering techniques might be employed to support a particular surgical intervention: neck dissection planning. Neck dissection planning poses challenging visualization problems due to the enormous density of crucial anatomic structures: Muscles, vascular structures and nerves share the same small space.

This discussion is based on an ongoing research project and describes experiences gained in the first two years. In total, 20 CT and MRI datasets have been analyzed and visualized to support preoperative planning. This case study is presented in more detail in [KTH*05].

5.1. Medical Background

Neck dissections are carried out for patients with malignant tumors in the head and neck region. These surgical procedures are necessary because the majority of the patients develops lymph node metastases in the neck region.

The extent of the intervention depends on the occurrence and location of enlarged (and probably) malignant lymph nodes. In particular, the infiltration of a large muscle (*M. sternocleidomastoideus*), a nerve (*N. facialis*) or blood vessel determine the surgical strategy. If for example the *A. carotis interna* is infiltrated, the patient is regarded as not resectable. The identification and the quantitative analysis of lymph nodes with respect to size and shape is crucial for the surgeon's decision. Visualization techniques should be developed to support decisions regarding the resectability and the surgical strategy for neck dissections.

5.2. Conventional Surgical Planning

Surgical planning in general as well as neck dissection planning as a special problem are carried out by means of 2d slices. Computer support allows to browse quickly through the slices, to change brightness and contrast and to perform measurements such as determining distances between selected points. 3d renderings are rarely used and many surgeons are not convinced of the additional value of 3d renderings at all. This attitude is not only due to their habits but has some serious arguments: in 2d slices each and every voxel is visible - it can be selected and its intensity value can be inquired. Instead, 3d visualizations provide an overview which is often too coarse for an in-depth planning of surgical strategies.

Since conventional surgical planning relies on 2d slices it is a good strategy to include 2d slices and the related manipulation techniques in advanced surgical planning systems. With this strategy, surgeons can plan their interventions as usual and can use the advanced techniques additionally. The most benefit can be achieved if 2d and 3d visualizations are carefully synchronized. A simple yet very efficient synchronization relates to the selection of an object. Whenever an object is selected in one view it should also be selected and emphasized in the other view (Fig. 9). This coordination allows surgeons to recognize which structures of the original volume data correspond to a 3d surface.

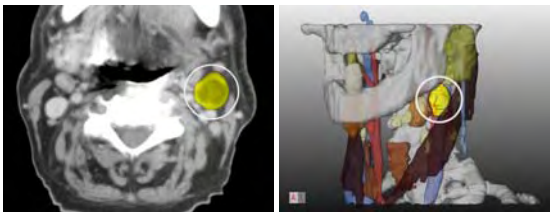


Figure 9: *The lymph node emphasized in the 3d visualization is simultaneously emphasized in the original slices.*

5.3. Advanced Surgical Planning

Advanced surgical planning requires reliable segmentation results. Segmentation of the relevant structures is a challenging task and computer support for minimally-interactive solutions is an area of ongoing research. We omit this important problem here and assume that segmentations of all relevant structures are available. For most anatomic structures the segmentation result comprises a connected set of voxels. For some structures, such as nerves, only selected voxels can be identified in a few slices. Nevertheless, it is desirable for surgical planning to see the rough course of the nerves. A special rendering technique is required for such incomplete structures.

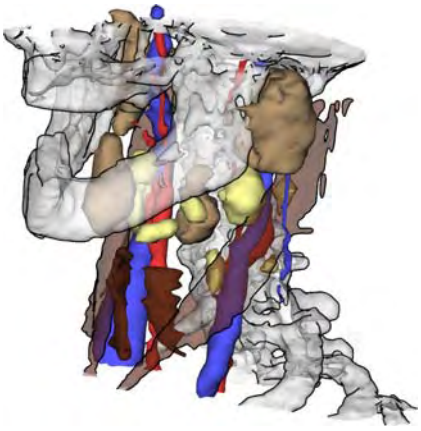


Figure 10: *Illustrative rendering for neck dissection planning. Silhouettes are generated for the bones which serve as anatomic context.*

Illustrative rendering. Silhouette rendering is employed for two purposes. The obvious use is to indicate the context objects, such as bones (Fig. 10). In addition, silhouettes may be used to discriminate two classes of objects; those which exhibit a certain feature are rendered with silhouettes enabled whereas the remaining objects are drawn without silhouettes. A reasonable use of this strategy is to enable silhouettes for objects which are more "interesting" since silhouettes direct the user's attention. In neck surgery planning, many lymph nodes have to be shown and explored by the user. In particular, lymph nodes which are enlarged and touch a critical structure are essential. In Fig. 11, these lymph nodes are rendered with silhouettes. Surgical users regard this as a substantial help since otherwise it is not recognizable whether the lymph node is (only) close to a critical structure or touches it.

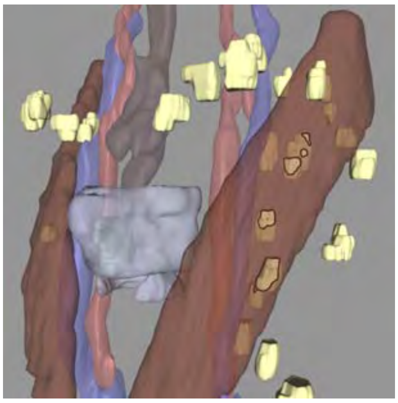


Figure 11: *Silhouette rendering indicates lymph nodes which touch and potentially infiltrate a critical structure.*

Approximative rendering of nerves. Nerves are very small structures compared to the spatial resolution of the data. Therefore, a single voxel contains nerve tissue and other adjacent tissue resulting in an intensity value which is very hard to distinguish from its surrounding. As a consequence, only in some slices a nerve could be identified at all. Anatomic experience indicates that nerves proceed almost linearly and do not deviate strongly from the straight connection between positions found in some slices. Since it is an important goal to prevent the injury of nerves we decided together with surgical partners that approximate visualizations should be generated where the segmented portions are emphasized and the part in between is reconstructed as a linear connection. The emphasis of the segmented portions is accomplished by means of small cylindrical disks (see Fig. 12).

5.4. Discussion

From an applications point of view, illustrative renderings target only a portion of the overall problem. Whether or not all relevant lymph nodes are detected and correctly delineated is probably more important than the details of their visualization. Valuable computer support for surgical planning requires high-quality and dedicated image acquisition, reliable and fast image analysis techniques *and* comprehensible visualizations. With respect to the visualization, conventional as well as more recent rendering techniques have their merits if appropriately combined. The combination of 2d and 3d renderings is an essential aspect for the acceptance and use of computer-supported surgical planning systems.

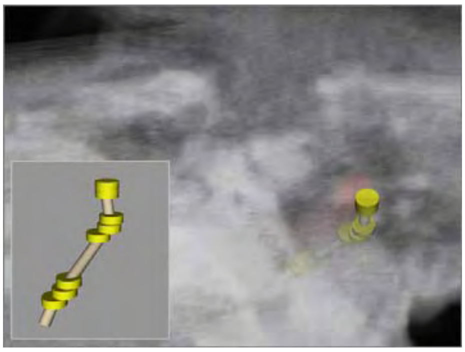


Figure 12: *Approximate visualization of N. facialis for neck dissection planning.*

6. Concluding Remarks

We presented a variety of illustration techniques applicable and viable to surgical planning in particular. The case study on neck dissection planning is based on a close collaboration with clinicians. The ideas how to use illustrative techniques were developed in discussions with them and the informal feedback is positive and encouraging.

Illustrative techniques are not widespread in surgical planning. Our research results indicate however, that illustrative techniques have a potential to improve surgical planning. The need for illustrative techniques will likely increase since more and more information is available preoperatively. The development of illustrative techniques should be directed to support the integrated and comprehensible visualization of these different sources of information.

The great advantage of using illustrative techniques is the additional freedom to fine-tune visualizations with respect to task-specific needs. The major drawback is that additional effort is required to process segmentation results and to select appropriate techniques and parameters. In clinical settings, these steps need to be strongly supported and automized if possible since the time for surgical planning remains severely restricted.

Future work. Illustrative rendering of medical volume data is one of the most active research areas in medical visualization. We discussed emphasis and illustration in static volume data. We briefly described the idea by [VKG04] to adjust visualization parameters to the importance of objects. This idea can be extended in many ways; in particular, the importance may be different for different parts of objects (the shape of a long muscle near a tumor is highly important whereas other areas are less important). More insight is necessary to automatically choose and combine illustrative techniques.

The most important work to be done concerns an evaluation of illustrative techniques by means of controlled user studies (see [KHI*03] for a discussion of issues in the preparation, execution, and analysis of user studies of visualization techniques). Such studies should investigate which visualization techniques provide additional insight into the patient's anatomy and should explore whether these insights influence the surgical strategy.

Further reading. For an overview on illustrative rendering, two dedicated books can be recommended: [SS02] and [GG01]. Since 2000, the conference Non-Photorealistic Rendering and Animation (NPAR) has been established as biannual conference (see the conference website http://www.npar.org). Concerning hatching, [PHWF01], [ZISS04] and [JEGPO02] are recommended; concerning silhouettes and feature lines we refer to [RC99]. [IHS02] give an overview discussing requirements and appropriate silhouette detection methods. [XNYC04] describe high quality silhouette rendering based on point-based models. [NSW02] describe interactive volume illustration where expressive hatchings are generated based on the estimated curvature, transparency and lightings. The placement of hatchings is guided by seedpoints specified by the user which are used as starting points for tracking lines with high curvature. [KWTM03] designed curvature-based transfer functions which make it possible to emphasize regions with large values for the major curvature κ_1 and κ_2. With this approach, ridges and valleys can be emphasized effectively. The use of line-based illustrations for displaying several isosurfaces and the value of line-based illustration for emphasis is discussed by [TC00].

We omitted a discussion of stippling techniques although these techniques are relevant for emphasis in medical visualizations. Stippling is based on points or small filled circles as rendering primitives. Meanwhile, complex objects can be stippled sufficiently fast and they can be easily parameterized. Stippling techniques are described for example in [DHvOS00], [Sec02] and [PS02]. Recent work by [LME*02] and [LME*03]) also considered stippling based on medical volume data. Gradient direction and gradient magnitude are estimated to adjust the local resolution of points so that silhouettes and local features are highlighted. Silhouette rendering and stippling is combined by [SBS05]. Expressive visualizations arise in particular if shading effects are integrated for controlling the density of points. It is recommended to look at applications of illustrative renderings in other application areas, such as illustration of terrain data [WV03] and illustration in technical areas [GSG*99]. Shape perception by means of computerized line drawings has been investigated—among others—by [Koe84] and [KvDCL96].

Illustrative rendering, focussing and emphasis techniques are also discussed in [PTD05]. Cutaway illustrations are discussed in depth in [DWE03]. Different variants of cutaway illustrations are considered, the combination with illustrative rendering is tackled and the mapping of cutaway rendering to modern graphics hardware is described there.

References

[Bau72] BAUMGART B. G.: *Winged Edge polyhedron representation*. Tech. rep., Stanford University, 1972.

[BKOS00] BERG M. D., KREVELD M. V., OVERMARS M., SCHWARZKOPF O.: *Computational Geometry*, 2 ed. Springer, 2000.

[BYB98] BOISSONNAT J.-D., YVINEC M., BRONNIMA H.: *Algorithmic Geometry*. Cambridge University Press, New York, 1998.

[DFR04] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S.: Interactive Rendering of Suggestive Contours with Temporal Coherence. In *Proc. of the third international symposium on Non-photorealistic animation and rendering* (2004), ACM Press, pp. 15–24.

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive Contours for Conveying Shape. In *Proc. of 30th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (2003), pp. 848–855.

[DHvOS00] DEUSSEN O., HILLER S., VAN OVERFELD C., STROTHOTTE T.: Floating points: A method for computing stipple drawings. *Computer Graphics Forum 19*, 3 (2000), 40–51. Proc. of Eurographics.

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 317–324.

[DWE03] DIEPSTRATEN J., WEISKOPF D., ERTL T.: Interactive Cutaway Illustrations. *Computer Graphics Forum 22*, 3 (2003), 523–532. Proc. of Eurographics.

[FS92] FEINER S., SELIGMANN D. D.: Cutaways and ghosting: satisfying visibility constraints in dynamic 3d illustrations. *The Visual Computer 8*, 5&6 (1992), 292–302.

[GG01] GOOCH B., GOOCH A. A.: *Non-Photorealistic Rendering*. AK Peters Ltd, July 2001.

[GSG*99] GOOCH B., SLOAN P., GOOCH A. A., SHIRLEY P., RIESENFELD R.: Interactive technical illustration. In *Proc. of ACM Symposium on Interactive 3D Graphics* (1999), ACM SIGCHI, pp. 31–38.

[Hod89] HODGES E. R. S.: *The Guild Handbook of Scientific Illustration*. Van Nostrand Reinhold, 1989.

[HZ00] HERTZMANN A., ZORIN D.: Illustrating Smooth Surfaces. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (June 2000), Brown J. R., Akeley K., (Eds.), ACM Press/Addison-Wesley Publishing Co., pp. 517–526.

[IFH*03] ISENBERG T., FREUDENBERG B., HALPER N., SCHLECHTWEG S., STROTHOTTE T.: A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications 23*, 4 (2003), 28–37.

[IFP95] INTERRANTE V. L., FUCHS H., PIZER S.: Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In *Proc. of IEEE Visualization* (1995), pp. 52–59.

[IFP96] INTERRANTE V. L., FUCHS H., PIZER S.: Illustrating Transparent Surfaces with Curvature-Directed Strokes. In *Proc. of IEEE Visualization* (1996), pp. 211–218.

[IHS02] ISENBERG T., HALPER N., STROTHOTTE T.: Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes. *Computer Graphics Forum 21*, 3 (2002), 249–258. Proc. of Eurographics.

[JEGPO02] JODOIN P.-M., EPSTEIN E., GRANGER-PICHÉ M., OSTROMOUKHOV V.: Hatching by example: a statistical approach. In *Proc. of the second international symposium on Non-photorealistic animation and rendering* (2002), ACM Press, pp. 29–36.

[KHI*03] KOSARA R., HEALEY C. G., INTERRANTE V., LAIDLAW D. H., WARE C.: Thoughts on User Studies: Why, How, and When. *IEEE Computer Graphics and Applications 23*, 4 (2003), 20–25.

[KHSI03a] KIM S., HAGH-SHENAS H., INTERRANTE V. L.: Conveying Shape with Texture: an experimental investigation of the impact of texture type on shape categorization judgments. In *Proc. of the IEEE Symposium on Information Visualization* (Oktober 2003), Munzner T., North S., (Eds.), ACM Press, pp. 163–170.

[KHSI03b] KIM S., HAGH-SHENAS H., INTERRANTE V. L.: Showing Shape with Texture: two directions seem better than one. In *Human Vision and Electronic Imaging VIII* (2003), Rogowitz B. E., Pappas T. N., (Eds.), vol. 5007, SPIE, pp. 332–339.

[KMH02] KOSARA R., MIKSCH S., HAUSER H.: Focus+context taken literally. *IEEE Computer Graphics and Applications 22*, 1 (2002), 22–29.

[Kob00] KOBBELT L.: Discrete fairing and variational subdivision for freeform surface design. *The Visual Computer 16*, 3-4 (2000), 142–158.

[Koe84] KOENDERINK J. J.: What does the occluding contour tell us about solid shape. *Perception 13* (1984), 321–330.

[KTH*05] KRÜGER A., TIETJEN C., HINTZE J., PREIM B., HERTEL I., STRAUSS G.: Interactive Visualization for Neck-Dissection Planning. In *Proc. of IEEE/Eurographics Symposium on Visualization* (2005), pp. 295–302.

[KvDCL96] KOENDERINK J. J., VAN DOORN A. J., CHRISTOU C., LAPPIN J. S.: Shape constancy in pictorial relief. *Perception 25* (1996), 155–164.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MÖLLER T.: Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proc. of IEEE Visualization* (2003), pp. 513–520.

[LFP*90] LEVOY M., FUCHS H., PIZER S. M., ROSENMAN J., CHANEY E. L., SHEROUSE G. W., INTERRANTE V., KIEL J.: Volume Rendering in Radiation Treatment Planning. In *Visualization in Biomedical Computing* (May 1990), pp. 4–10.

[LME*02] LU A., MORRIS C. J., EBERT D. S., RHEINGANS P., HANSEN C.: Non-photorealistic volume rendering using stippling techniques. In *Proc. of IEEE Visualization* (2002), IEEE Computer Society, pp. 211–218.

[LME*03] LU A., MORRIS C. J., EBERT D. S., RHEINGANS P., HANSEN C.: Illustrative Interactive Stipple Rendering. *IEEE Transactions on Visualization and Graphics 9*, 2 (2003), 1–12.

[NSW02] NAGY Z., SCHNEIDER J., WESTERMANN R.: Interactive volume illustration. In *Proc. of Vision, Modelling and Visualization* (2002), pp. 497–504.

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-Time Hatching. In *Proc. of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (Juni 2001), Pocock L., (Ed.), ACM Press, pp. 579–584.

[PS02] PASTOR O. M., STROTHOTTE T.: Frame-Coherent Stippling. *Computer Graphics Forum 22*, 3 (2002), 145–152. Proc. of Eurographics.

[PTD05] PREIM B., TIETJEN C., DÖRGE C.: NPR, Focussing and Emphasis in Medical Visualization. In *Proc. of Simulation und Visualisierung* (March 2005), vol. I, SCS.

[RC99] RASKAR R., COHEN M.: Image Precision Silhouette Edges. In *ACM Symposium Interactive 3D Graphics* (1999), Spencer S. N., (Ed.), ACM Press, pp. 135–140.

[SBS05] SALAH Z., BARTZ D., STRASSER W.: Illustrative Rendering of Segmented Anatomical Data. In *Proc. of Simulation und Visualisierung* (March 2005), SCS, pp. 175–184.

[SE02] SCHNEIDER P., EBERLY D. H.: *Geometric Tools for Computer Graphics*. Morgan Kaufmann, 2002.

[Sec02] SECORD A.: Weighted Voronoi stippling. In *Proc. of the second international symposium on Non-photorealistic animation and rendering* (2002), ACM Press, pp. 37–43.

[SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. Morgan Kaufmann, San Francisco, April 2002.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-D shapes. In *Proc. of the 17th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (August 1990), Baskett F., (Ed.), ACM Press, pp. 197–206.

[Tau95a] TAUBIN G.: Curve and surface smoothing without shrinkage. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision* (Washington, DC, USA, 1995), IEEE Computer Society, p. 852.

[Tau95b] TAUBIN G.: A signal processing approach to fair surface design. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 351–358.

[TC00] TREAVETT S. M. F., CHEN M.: Pen-and-ink rendering in volume visualization. In *Proc. of IEEE Visualization* (October 2000), pp. 203–211.

[TIP05] TIETJEN C., ISENBERG T., PREIM B.: Combining Silhouettes, Shading, and Volume Rendering for Surgery Education and Planning. In *Proc. of IEEE/Eurographics Symposium on Visualization* (2005), Springer, pp. 303–310.

[Tre85] TREISMAN A.: Preattentive Processing in Vision. *Computer Vision, Graphics and Image Processing 31*, 2 (1985), 156–177.

[TWBO02] TASDIZEN T., WHITAKER R., BURCHARD P., OSHER S.: Geometric surface smoothing via anisotropic diffusion of normals. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 125–132.

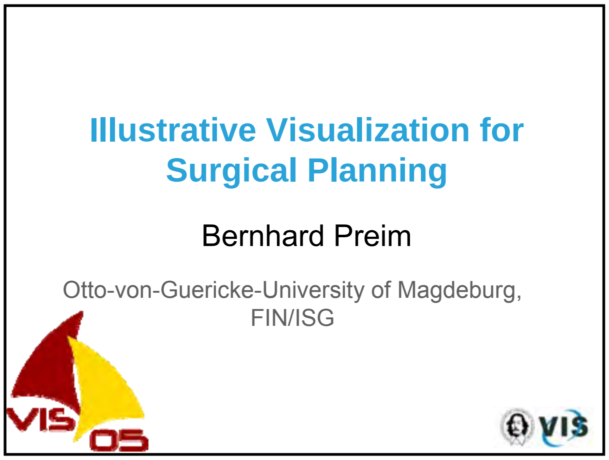
[VKG04] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-driven volume rendering. In *Proc. of IEEE Visualization* (Los Alamitos, CA, 2004), IEEE, pp. 139–145.

[WV03] WHELAN J. C., VISVALINGAM M.: Formulated Silhouettes for Sketching Terrain. In *Theory and Practice of Computer Graphics* (2003), pp. 90–96.

[XNYC04] XU H., NGUYEN M. X., YUAN X., CHEN B.: Illustrative Silhouette Rendering for Point-Based Models. In *Proc. of EG Symposium on Point-Based Graphics* (2004), Eurographics Association, pp. 13–18.

[YOB02] YAGOU H., OHTAKE Y., BELYAEV A.: Mesh smoothing via mean and median filtering applied to face normals. In *GMP '02: Proceedings of the Geometric Modeling and Processing âýý Theory and Applications (GMP'02)* (Washington, DC, USA, 2002), IEEE Computer Society, p. 124.

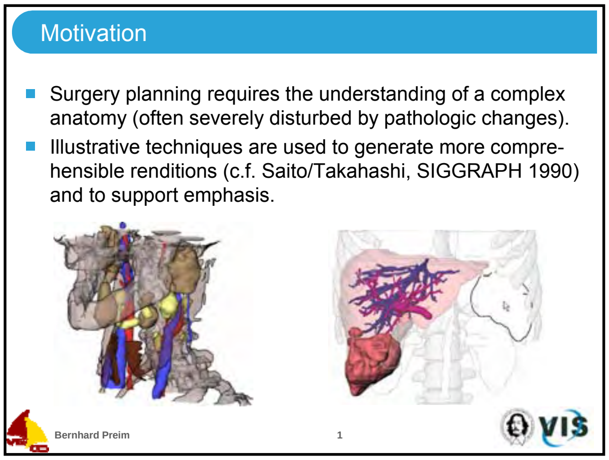
[ZISS04] ZANDER J., ISENBERG T., SCHLECHTWEG S., STROTHOTTE T.: High quality hatching. *Computer Graphics Forum 23* (2004), 421.



Illustrative Visualization for Surgical Planning

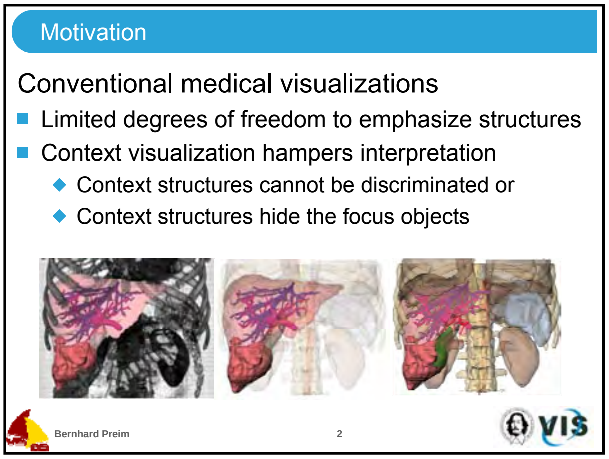
Bernhard Preim

Otto-von-Guericke-University of Magdeburg, FIN/ISG



Motivation

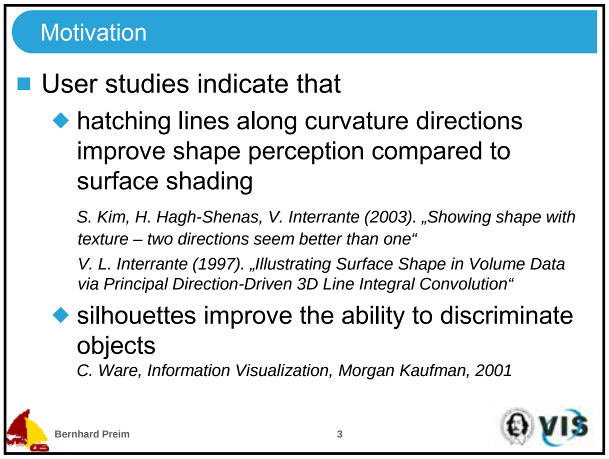
- Surgery planning requires the understanding of a complex anatomy (often severely disturbed by pathologic changes).
- Illustrative techniques are used to generate more comprehensible renditions (c.f. Saito/Takahashi, SIGGRAPH 1990) and to support emphasis.



Motivation

Conventional medical visualizations

- Limited degrees of freedom to emphasize structures
- Context visualization hampers interpretation
  - Context structures cannot be discriminated or
  - Context structures hide the focus objects



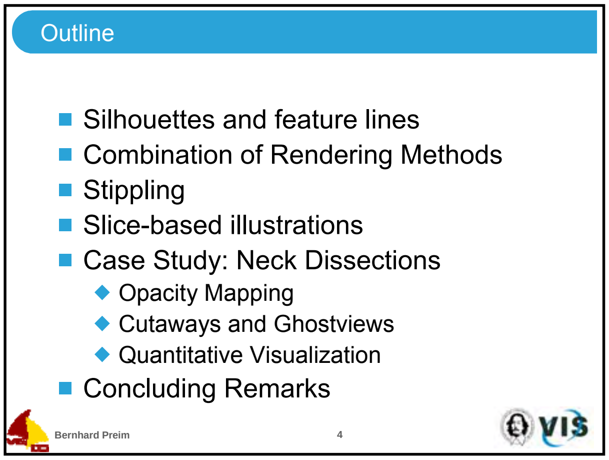
Motivation

- User studies indicate that
  - hatching lines along curvature directions improve shape perception compared to surface shading

    S. Kim, H. Hagh-Shenas, V. Interrante (2003). „Showing shape with texture – two directions seem better than one“

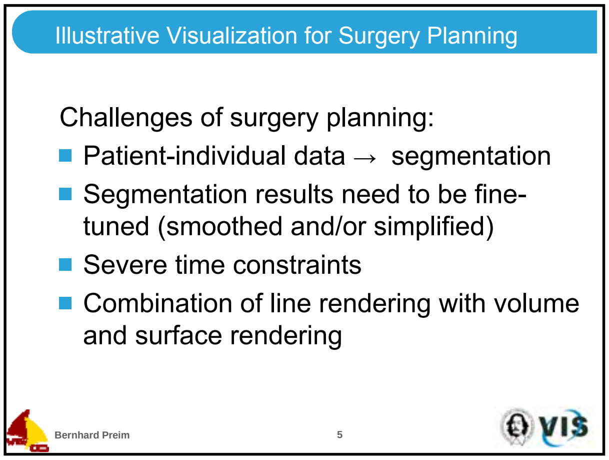
    V. L. Interrante (1997). „Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution“
  - silhouettes improve the ability to discriminate objects

    C. Ware, Information Visualization, Morgan Kaufman, 2001



Outline

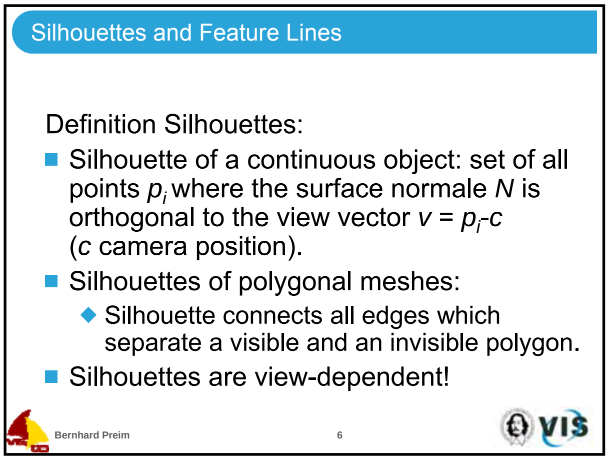
- Silhouettes and feature lines
- Combination of Rendering Methods
- Stippling
- Slice-based illustrations
- Case Study: Neck Dissections
  - Opacity Mapping
  - Cutaways and Ghostviews
  - Quantitative Visualization
- Concluding Remarks



Illustrative Visualization for Surgery Planning

Challenges of surgery planning:

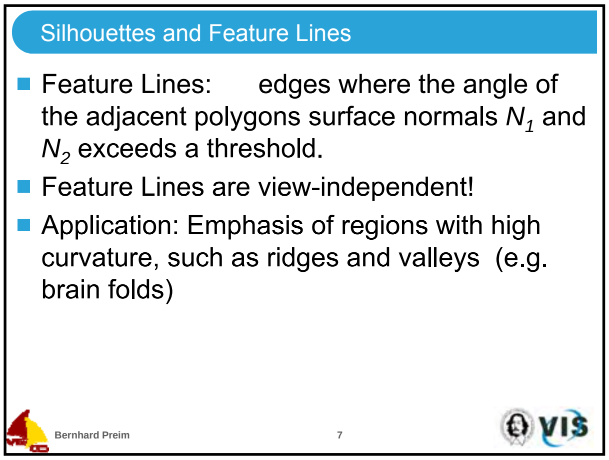
- Patient-individual data → segmentation
- Segmentation results need to be fine-tuned (smoothed and/or simplified)
- Severe time constraints
- Combination of line rendering with volume and surface rendering



Silhouettes and Feature Lines

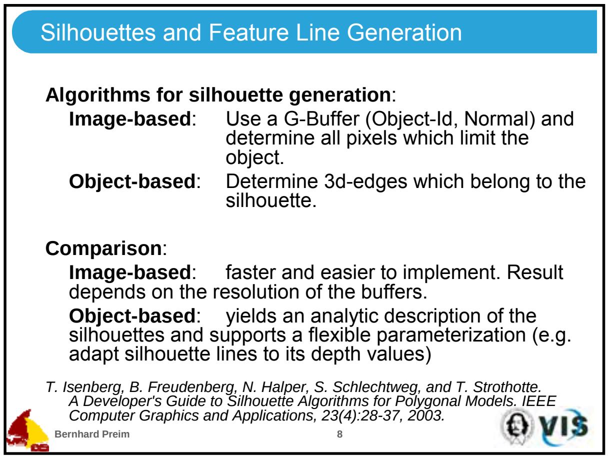
Definition Silhouettes:

- Silhouette of a continuous object: set of all points p_i where the surface normale N is orthogonal to the view vector $v = p_i$-c (c camera position).
- Silhouettes of polygonal meshes:
  - Silhouette connects all edges which separate a visible and an invisible polygon.
- Silhouettes are view-dependent!



Silhouettes and Feature Lines

- Feature Lines: edges where the angle of the adjacent polygons surface normals N_1 and N_2 exceeds a threshold.
- Feature Lines are view-independent!
- Application: Emphasis of regions with high curvature, such as ridges and valleys (e.g. brain folds)



Silhouettes and Feature Line Generation

Algorithms for silhouette generation:

Image-based: Use a G-Buffer (Object-Id, Normal) and determine all pixels which limit the object.

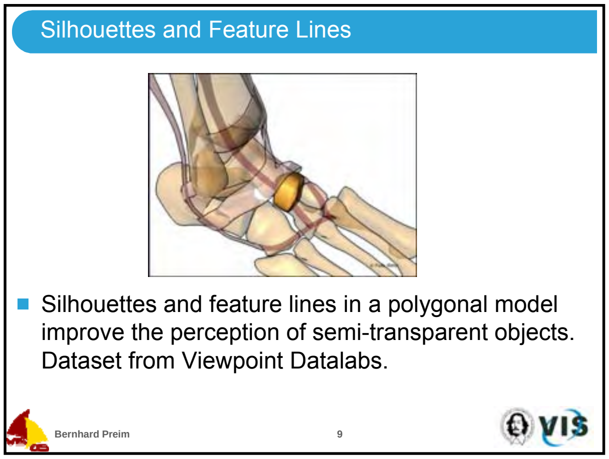
Object-based: Determine 3d-edges which belong to the silhouette.

Comparison:

Image-based: faster and easier to implement. Result depends on the resolution of the buffers.

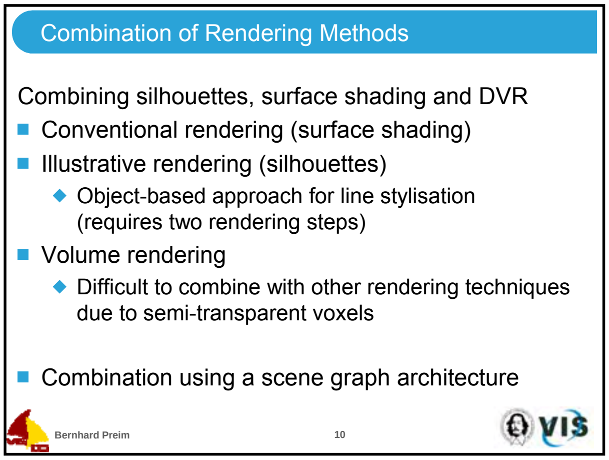
Object-based: yields an analytic description of the silhouettes and supports a flexible parameterization (e.g. adapt silhouette lines to its depth values)

T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte. A Developer's Guide to Silhouette Algorithms for Polygonal Models. IEEE Computer Graphics and Applications, 23(4):28-37, 2003.



Silhouettes and Feature Lines

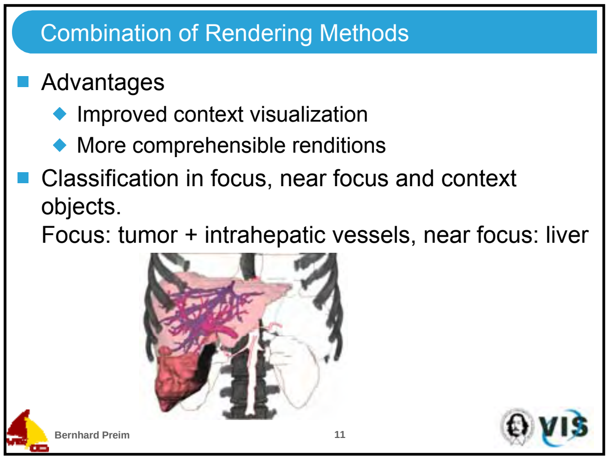
- Silhouettes and feature lines in a polygonal model improve the perception of semi-transparent objects. Dataset from Viewpoint Datalabs.



Combination of Rendering Methods

Combining silhouettes, surface shading and DVR

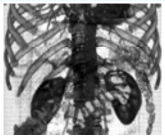
- Conventional rendering (surface shading)
- Illustrative rendering (silhouettes)
  - Object-based approach for line stylisation (requires two rendering steps)
- Volume rendering
  - Difficult to combine with other rendering techniques due to semi-transparent voxels
- Combination using a scene graph architecture

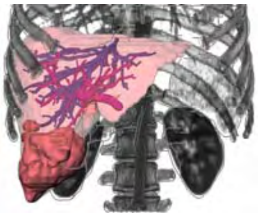


Combination of Rendering Methods

- Advantages
  - Improved context visualization
  - More comprehensible renditions
- Classification in focus, near focus and context objects.
  Focus: tumor + intrahepatic vessels, near focus: liver

Combination of Rendering Methods

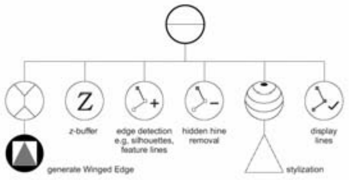


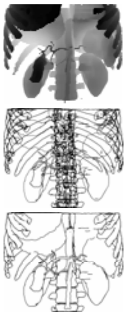


Combination of Rendering Methods

Silhouette rendering

1. z-buffer rendering
2. Generation of the silhouettes
3. Hidden line removal
4. Rendering of the silhouettes

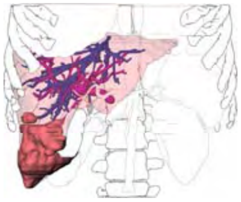




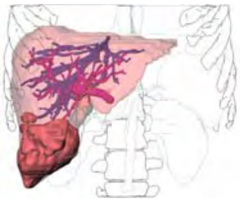
Combination of Rendering Methods

Combination of silhouette and surface rendering

1. Surface rendering
2. Silhouette rendering



Wrong order (Silhouettes first)



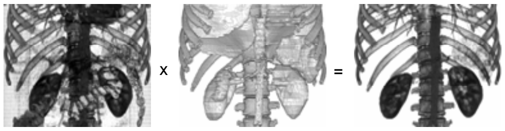
Correct order

Combination of Rendering Methods

Combination of surface and volume rendering

1. Rendering of the polygonal objects
2. Rendering of the volume dataset

Tagged volume rendering

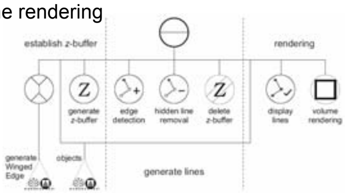


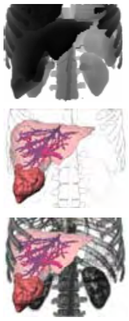
Avoid unwanted occlusions by masking

Combination of Rendering Methods

Combination of all three rendering styles

1. z-buffer rendering of all segmented objects
2. Generation of silhouettes and feature lines (including HLR)
3. Clear the z-buffer
4. Rendering line- and surface shaded objects
5. Volume rendering



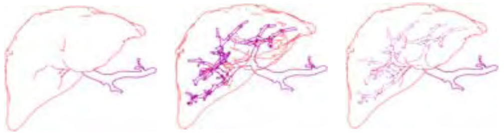


Combination of Rendering Methods

Removing self-occluding lines:

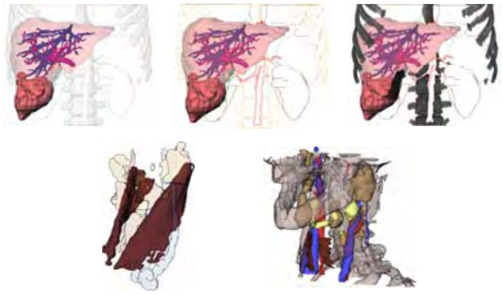
For each object do:

1. Rendering the z-buffer of the current object
2. Line generation and HLR
3. Clearing the z-buffer



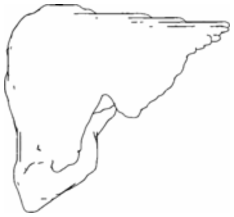
Combination of Rendering Methods

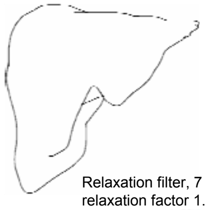
Visualization examples



Smoothing

- Stair artefacts on isosurfaces
- Produces wrong "silhouette" lines
- Interpolate intermediate slices or smooth surface afterwards





Relaxation filter, 7 iterations, relaxation factor 1.0

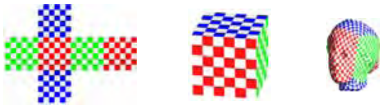
Efficient Texture-based Stippling

- Stippling is an effective visualization technique for medical objects (recall David Eberts talk)
- Provides an adjustable level of sparseness
- For surfaces of anatomic structures efficient methods are desirable to support interactive handling and combination with other vis. techniques.
- Hardware support is feasible if stippling can be accomplished with texture mapping.
- **Challenges**:
  - Controlling distortions when textures are mapped to surfaces with arbitrary curvature and orientation
  - Limitations of shader languages

Efficient Texture-based Stippling

Cube Mapping:

- Cubes as intermediate geometry
- Cube Map: Texture consisting of 6 equally sized 2d-Textures



Emil Praun and Hugues Hoppe. „Spherical Parametrization and Remeshing", ACM Transactions on Graphics, 22(3):340–349, 2003.

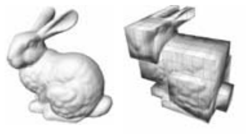
Idea :

- Several cubes with different textures (depending on the local lightness)

Efficient Texture-based Stippling

Polycube Mapping:

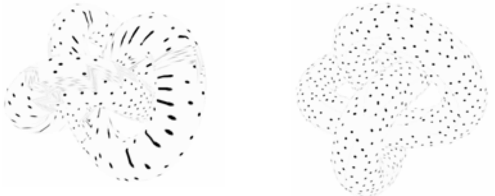
- A polycube is generated in a preprocessing step (equally sized cubes)
- Space is subdivided into cubes
- Texturemapping is separated for each cube to minimize the distortion
-



Marco Tarini et al. „PolyCube-Maps", ACM Transactions on Graphics, 23(3):853–860, 2004.

Efficient Texture-based Stippling

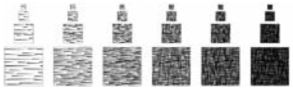
- Distortion effects (elongated disks) with Cubemapping can be minimized with polycube mapping.



Master thesis of Alexandra Baer, University of Magdeburg

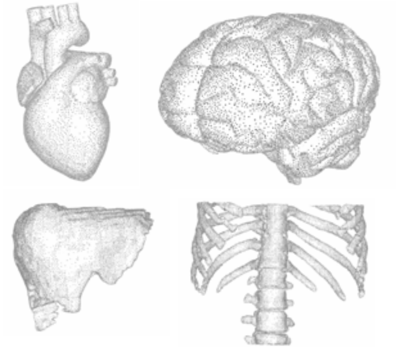
Efficient Texture-based Stippling

- Appropriate size of points for different distances
- Textures in different resolutions (tonal art maps)



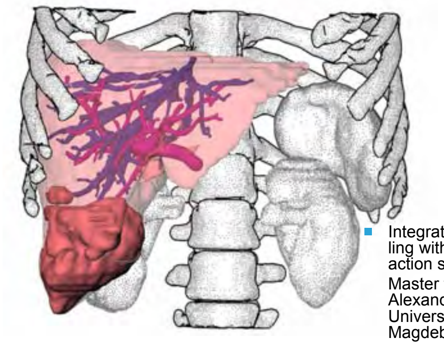
Emil Praun et al. „Lapped Textures“, Proc. of SIGGRAPH, pp. 465–470, 2000.

Efficient Texture-based Stippling



Examples:
Master thesis of Alexandra Baer, University of Magdeburg

Efficient Texture-based Stippling



- Integration of stippling with other interaction styles

Master thesis of Alexandra Baer, University of Magdeburg

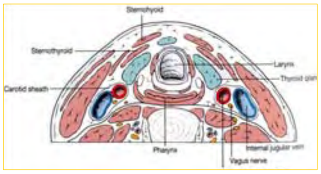
Efficient Texture-based Stippling: Summary

- Stippling of surfaces representing segmented surfaces.
- Acceleration by means of shader programing
- Texture-based approach
- Polycubemapping allows to minimize texture distortion

Slice-based Illustrations

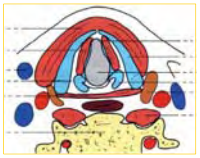
- Medical doctors often use slice-based viewing (for diagnosis, also for intraoperative support)
- Consider the enhancement of these visualizations.
- Inspiration: textbooks on cross-sectional anatomy

Galanski/Prokop, *Ganzkörper-CT*, [1998]

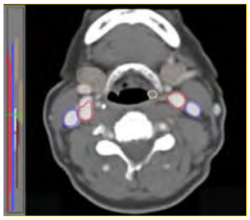


Möller/Reif, *Taschenatlas Schnittbildanatomie*, Thieme,1993

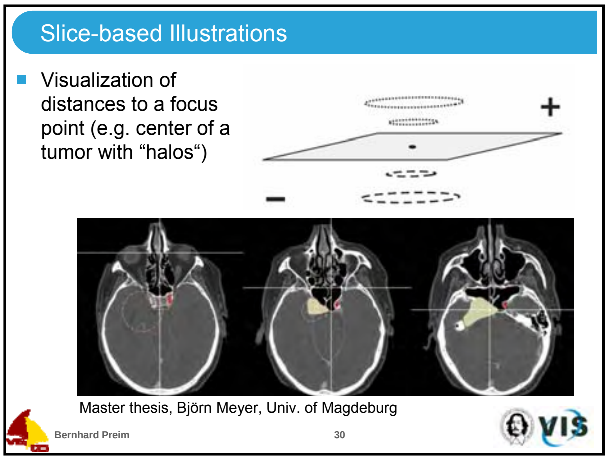
Slice-based Illustrations



Möller/Reif, *Taschenatlas Schnittbildanatomie*, Thieme,1993



Master thesis, Björn Meyer, Univ. of Magdeburg

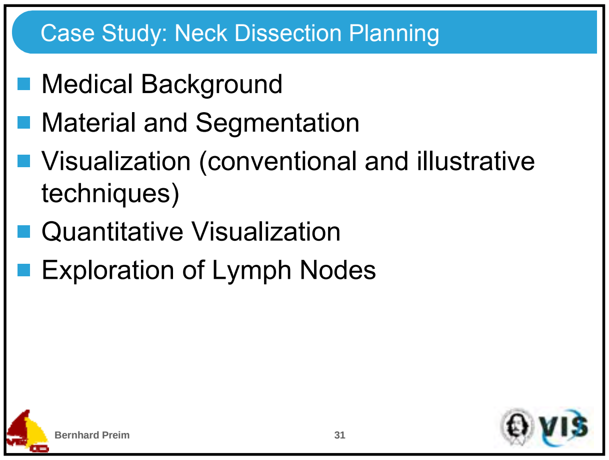


Slide-based Illustrations

- Visualization of distances to a focus point (e.g. center of a tumor with “halos“)

Master thesis, Björn Meyer, Univ. of Magdeburg

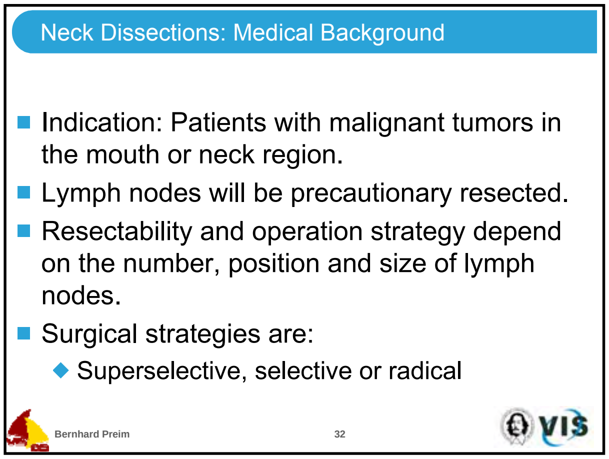
Bernhard Preim 30



Case Study: Neck Dissection Planning

- Medical Background
- Material and Segmentation
- Visualization (conventional and illustrative techniques)
- Quantitative Visualization
- Exploration of Lymph Nodes

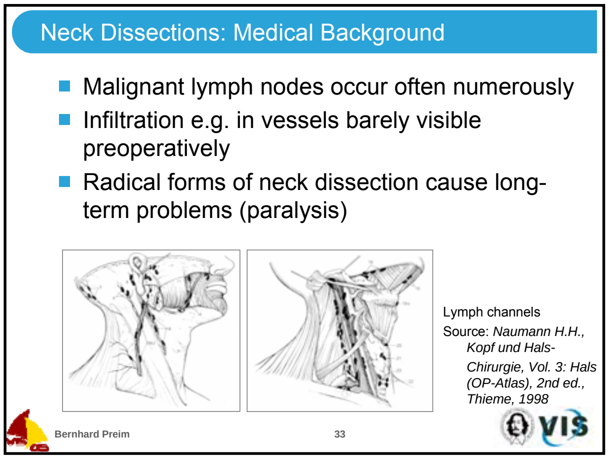
Bernhard Preim 31



Neck Dissections: Medical Background

- Indication: Patients with malignant tumors in the mouth or neck region.
- Lymph nodes will be precautionary resected.
- Resectability and operation strategy depend on the number, position and size of lymph nodes.
- Surgical strategies are:
  - Superselective, selective or radical

Bernhard Preim 32



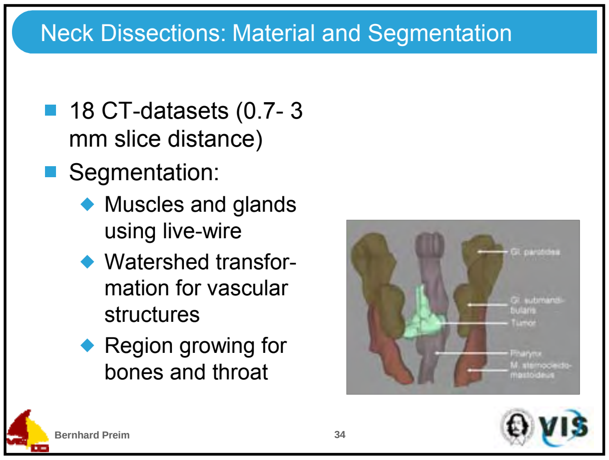
Neck Dissections: Medical Background

- Malignant lymph nodes occur often numerously
- Infiltration e.g. in vessels barely visible preoperatively
- Radical forms of neck dissection cause long-term problems (paralysis)

Lymph channels

Source: *Naumann H.H., Kopf und Hals-Chirurgie, Vol. 3: Hals (OP-Atlas), 2nd ed., Thieme, 1998*

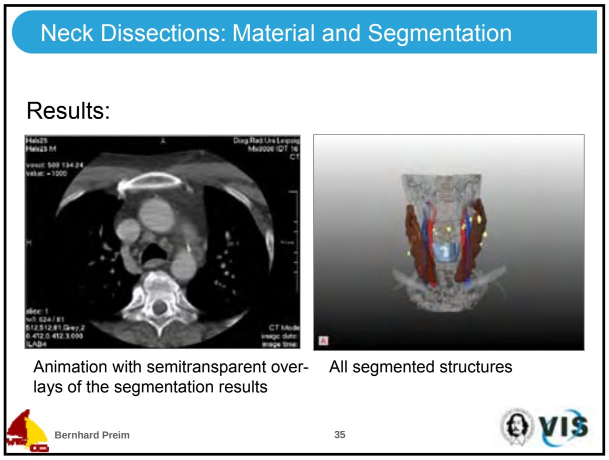
Bernhard Preim 33



Neck Dissections: Material and Segmentation

- 18 CT-datasets (0.7- 3 mm slice distance)
- Segmentation:
  - Muscles and glands using live-wire
  - Watershed transformation for vascular structures
  - Region growing for bones and throat

Bernhard Preim 34



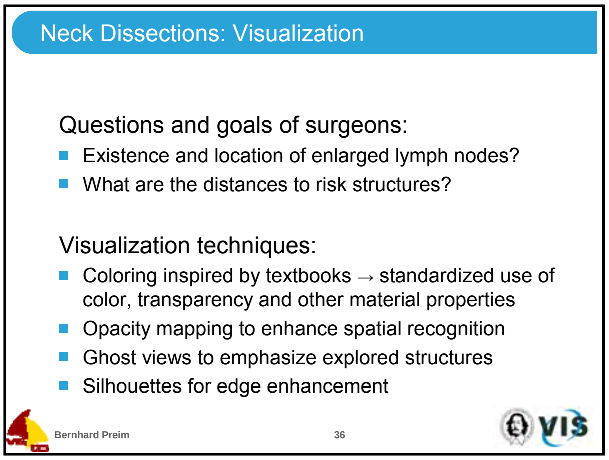
Neck Dissections: Material and Segmentation

Results:

Animation with semitransparent overlays of the segmentation results

All segmented structures

Bernhard Preim 35



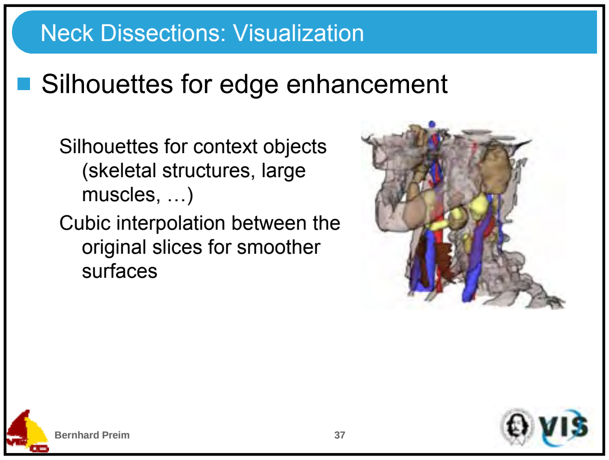
Neck Dissections: Visualization

Questions and goals of surgeons:

- Existence and location of enlarged lymph nodes?
- What are the distances to risk structures?

Visualization techniques:

- Coloring inspired by textbooks → standardized use of color, transparency and other material properties
- Opacity mapping to enhance spatial recognition
- Ghost views to emphasize explored structures
- Silhouettes for edge enhancement

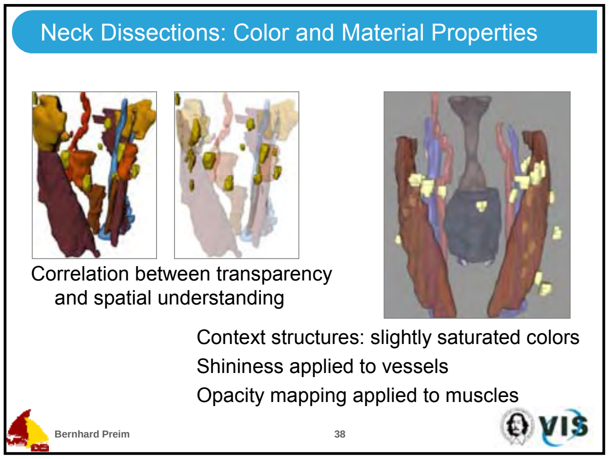


Neck Dissections: Visualization

- Silhouettes for edge enhancement

Silhouettes for context objects (skeletal structures, large muscles, …)

Cubic interpolation between the original slices for smoother surfaces



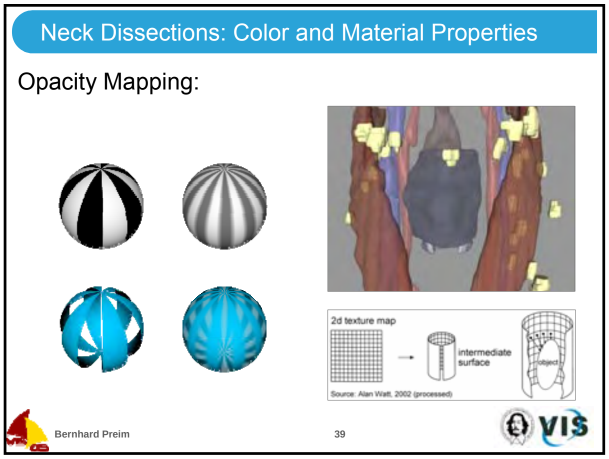
Neck Dissections: Color and Material Properties

Correlation between transparency and spatial understanding

Context structures: slightly saturated colors

Shininess applied to vessels

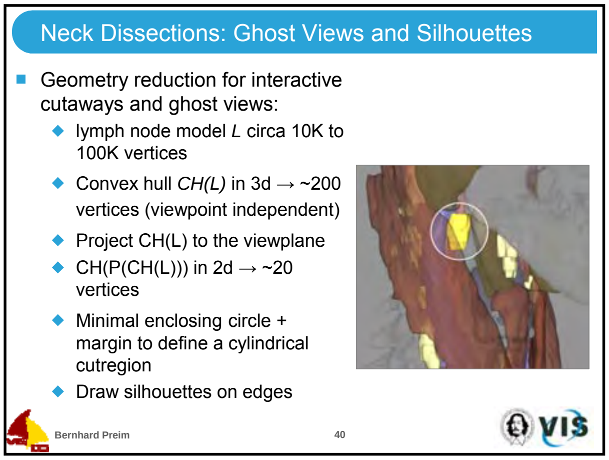
Opacity mapping applied to muscles



Neck Dissections: Color and Material Properties

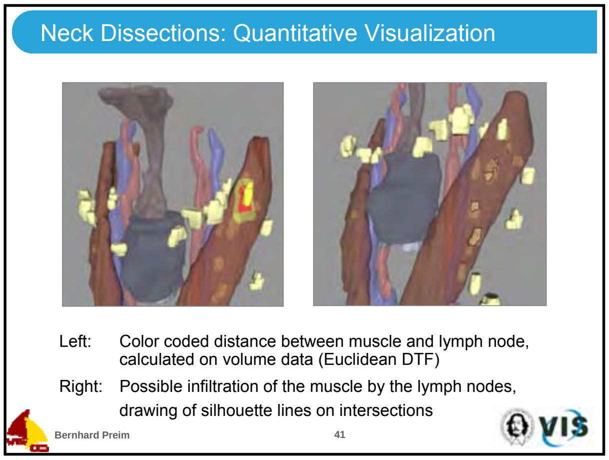
Opacity Mapping:

Source: Alan Watt, 2002 (processed)



Neck Dissections: Ghost Views and Silhouettes

- Geometry reduction for interactive cutaways and ghost views:
  - lymph node model *L* circa 10K to 100K vertices
  - Convex hull *CH(L)* in 3d → ~200 vertices (viewpoint independent)
  - Project CH(L) to the viewplane
  - CH(P(CH(L))) in 2d → ~20 vertices
  - Minimal enclosing circle + margin to define a cylindrical cutregion
  - Draw silhouettes on edges



Neck Dissections: Quantitative Visualization

Left: Color coded distance between muscle and lymph node, calculated on volume data (Euclidean DTF)

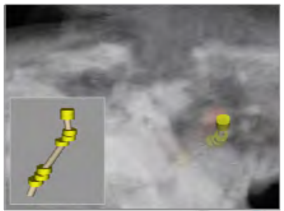
Right: Possible infiltration of the muscle by the lymph nodes, drawing of silhouette lines on intersections

Neck Dissections: Approximative Visualization

- Small structures can only be partially segmented due to the partial volume effect.
- Approximative visualizations are helpful provided that the uncertainty is encoded.

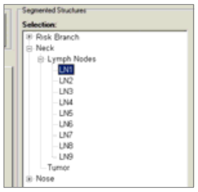
Example:
Nerves are dected in a few slices. Corresponding positions are marked with disks and connected with straight lines.

Surgeons interpret these images using anatomic knowledge.



Neck Dissections: Exploration of Lymph Nodes

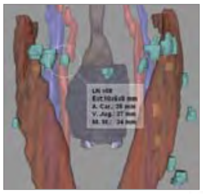
- Selection by name is not feasible.
- Better: sequential order e.g. starting with the lymph node with maximal extent or lowest distance to risk structures.



Segmented Structures

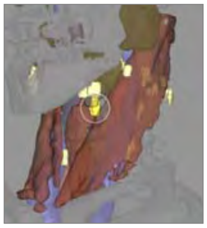
Selection:

- Risk Branch
- Neck
  - Lymph Nodes
    - LN1
    - LN2
    - LN3
    - LN4
    - LN5
    - LN6
    - LN7
    - LN8
    - LN9
  - Tumor
- Nose



Neck Dissections: Exploration of Lymph Nodes

- Ghost views for the sequential emphasis of lymph nodes
- Cylindrical cutting volume, color saturation, transparency and silhouettes



Concluding Remarks

- Visualization and interaction for surgery planning:
  - Standardized visualizations for surgical planning (time savings, reproducibility)
  - Include quantitative information in visualizations
  - Adjust material properties and silhouettes for focus control and enhanced spatial recognition
  - Sequential emphasis of pathologic structures (lymph nodes, lung nodules, …) using ghost views

Concluding Remarks (cont`d)

- Other illustrative techniques are promising provided that their use could be highly standardized.

F. Dong, G. J. Clapworthy, H. Lin und M. A. Krokos (2003). "Non-Photo-realistic Rendering of Medical Volume Data", IEEE CG& A, Vol. 23(4): 44-52

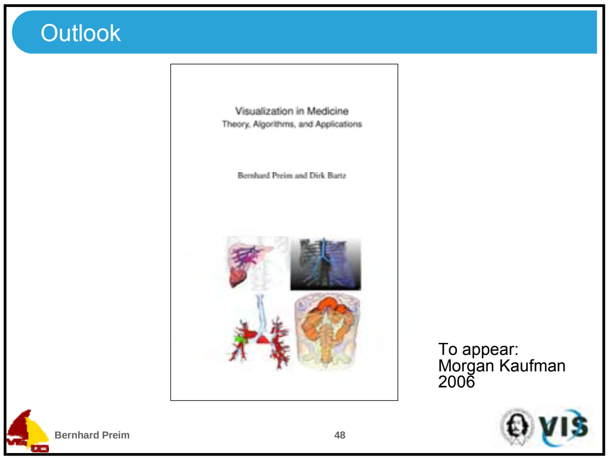
Concluding Remarks (cont`d)

- User studies are required to compare visualization options with respect to task-specific problems (e.g. exploration of vasculature around a tumor).
- Investigate usefulness:

  Does your (new) visualization technique

  provides additional insight?

  influence surgical strategies?



Outlook

Visualization in Medicine
Theory, Algorithms, and Applications

Bernhard Preim and Dirk Bartz

To appear:
Morgan Kaufman
2006

Acknowledgements

Department of Simulation and Graphics
Faculty of Computer Science, University of Magdeburg, Germany
Visualization Group
http://wwwisg.cs.uni-magdeburg.de/cv/

ENT Department
University Hospital of Leipzig, Germany
Innovation center for computer assisted surgery
http://www.iccas.de

The case study was carried out in a project supported by the Deutsche Forschungsgemeinschaft (DFG) (Priority Programme 1124, PR 660/3-1).

VIS
HNOLEIPZIG
Hals-Nasen-Ohren-Universitätsklinik
iccas innovation center computer assisted surgery
Deutsche Forschungsgemeinschaft DFG

Smart 3d visualizations in clinical applications

Bernhard Preim[1], Heinz-Otto Peitgen[2]
[1]Institute for Simulation and Graphics, Department for Computer Science
Otto-von-Guericke-University of Magdeburg, Universitaetsplatz 2, 39106 Magdeburg
e-Mail: preim@isg.cs.uni-magdeburg.de
[2]Center for Medical Diagnosic Systems and Visualization, Universitaetsallee 29,
28359 Bremen, e-Mail: peitgen@mevis.de

Abstract. We discuss techniques for the visualization of medical volume data dedicated for their clinical use. We describe the need for rapid dynamic interaction facilities with such visualizations and discuss emphasis techniques in more detail. Another crucial aspect of medical visualization is the integration of 2d and 3d visualizations. In order to organize this discussion, we introduce 6 „Golden" rules for medical visualizations.

1 Introduction

Many tasks in medical applications can be supported with appropriate visualizations. In addition to visualizations from textbooks, computer-generated and highly interactive 3d visualizations are employed in medical education. In anatomy education, for example, the VOXELMAN [9-11] is used to allow the exploration of medical volume data with advanced cut and clip functions. The volume data was enriched by a sophisticated knowledge representation about anatomical taxonomy ("intelligent" voxels). The pioneering work of the VOXELMAN has inspired some of the ideas presented here. Another long-term effort has been carried out in surgery simulation by KÜHNAPFEL et al. [1, 12]. They developed the KISMET system which simulates bleeding and smoke caused by surgical devices besides a variety of other interactions between surgical devices and human tissue.

"Idealized" and clinical data. The discussion of appropriate visualizations has to consider the quality of the underlying data which differs strongly between the abovementioned educational applications and applications for the routine clinical use. For educational purposes, idealized data, such as the Visible Human dataset of the National Library of Medicine are employed. The use of a cadaver dataset has several advantages: high radiation can be applied and motion artifacts for example from breathing do not occur. These datasets are carefully prepared and enriched with additional information. Such a time-consuming approach is not feasible for medical diagnosis and treatment planning. The underlying data have less quality (because they are acquired from living patients). This paper is focused on clinical applications and discusses visualization and interaction techniques that are designed to support diagnostic processes and treatment decisions, such as operability of a patient.

This paper is organized as follows: In Sect. 2 visualization techniques currently used for medical diagnosis and treatment planning are described. This section is followed by the brief Sect. 3 which names six "golden rules" for smart medical visualizations which are discussed in the remainder of this paper.

2 3d Visualizations for clinical applications

3d visualizations in medicine are based on radiological image data. Primarily CT (computed tomography) data,and MRI (Magnetic Resonance Imaging) are used. The traditional – film-based or soft-copy – "reading" of radiological image data represents a slice-by-slice inspection. Due to the growing resolution of image scanners this is no longer feasible as the only inspection method. As an example, CT data of the human thorax acquired with multislice image devices produce some 500 slices. Therefore, 3d visualizations, such as direct volume rendering, isosurface rendering, and maximum-intensity-projections (MIP) are becoming more commonly used. MIP images depict the brightest voxel along each line of sight and thus do not require any user interaction with respect to opacity settings ([3] gives an excellent overview on these basic techniques). Direct volume rendering and isosurface rendering, on the other hand, strongly depend on user settings. Predefined transfer functions for certain visualization tasks, such as highlighting bony structures in CT data, are used to reduce the interaction effort. Unfortunately, such predefined settings are not reliable for MR data (which do not exhibit a standardized intensity scale and are less homogeneous) and for the visual delineation of structures with similar intensity values.

If the diagnosis is confirmed, radiologists demonstrate the images to the referring physician in order to discuss the therapy. Furthermore, if an intervention is considered radiologists and surgeons discuss the strategy. Still, it is common to use static images, either slices from the original data or rendered images for this discussion. If interactive 3d visualizations are used at all, it is the radiologist who operates the system in order to show what is important. In order to understand the 3d reality of a particular patient's organ, it is desirable that interactive 3d visualizations are used and that the surgeon is in control to explore the data in order to answer his or her questions.

3 The six golden rules for the design of smart medical visualization

SHNEIDERMAN succeeded in convincing people of his 8 "golden rules" for user interface design [20]. Inspired by these we present 6 "golden rules" for the design of smart medical visualizations. "Smart" means that the visualizations are useful for clinical tasks which includes that they are recognizable and dedicated to a particular task. Moreover, smart medical visualizations are fast to generate and can be interactively explored. The following rules are based on our experience with the development and evaluation of clinical applications.

- Integrate 2d and 3d visualization with interaction facilities in both views
- Provide useful defaults for visualization parameters
- Provide dynamic interaction facilities
- Use model-based visualizations, for example for thin elongated structures
- Provide appropriate emphasis techniques
- Include anatomical context in the visualization of the relevant structures

The rest of the paper is devoted to the discussion of these 6 rules.

4 Integration of 2d and 3d visualizations

On the first glance it seems that smart medical visualizations should be 3d exclusively. The human body is a three-dimensional system; therapy planning and surgery simulation based on 3d visualizations are therefore more realistic than visualizations based on 2d slice visualizations. 3d visualizations with appropriate depth-cues (first of all shading) are intuitively comprehensible and seem to be superior. Despite of these obvious advantages, clinically relevant therapy planning systems contain 2d and 3d visualizations and interactive manipulations are allowed in both often with some kind of synchronization between the views.

The question arises whether 2d visualizations are only needed for conservative medical doctors not accustomed to the great benefit of 3d visualizations, so-to-say to be upward compatible to slice-based reading. Researchers in the field agree on real benefits of and needs for 2d visualizations, too.

In general, 2d-visualizations better support precise interaction. Precision is required, for example when measurements are derived, implants are placed in the geometric model of the patient and drilling procedures are planned. 2d visualizations are useful for these tasks because selection tasks which are part of the discussed tasks can be accomplished more precisely. Each voxel can be selected if the appropriate slice is chosen for the 2d visualization. 3d visualizations provide an overview; the "big" picture of the data. They are used, for example, to decide which range of slices is displayed in the respective 2d view in more detail. As a consequence, 2d and 3d visualizations allow for interactions in both views.

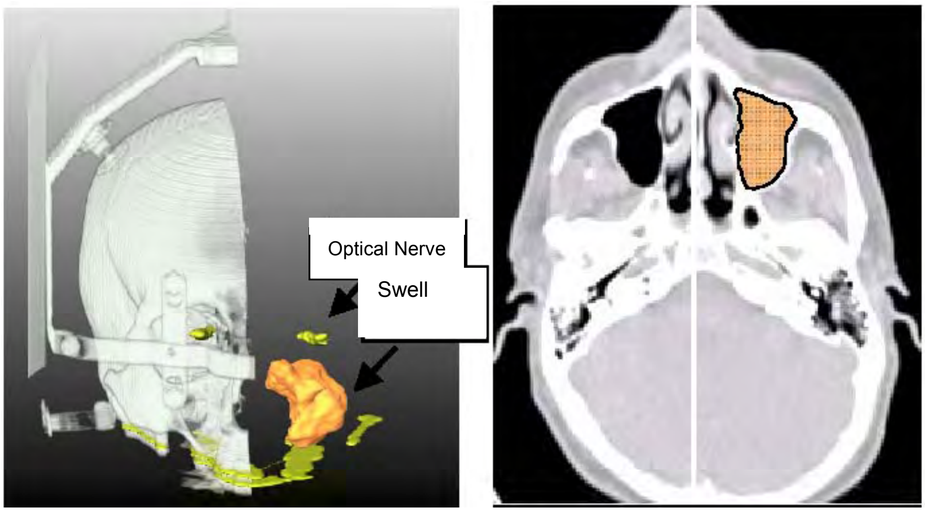


Fig. 1: The 3d visualization of a human head in CT data and the corresponding 2d visualization. The clipping plane is also included in the 2d visualization and can be manipulated in both views. The segmentation results are transparently overlaid to the original data in the 2d view. Radiological data kindly provided by University of Leipzig (Dr. Strauß).

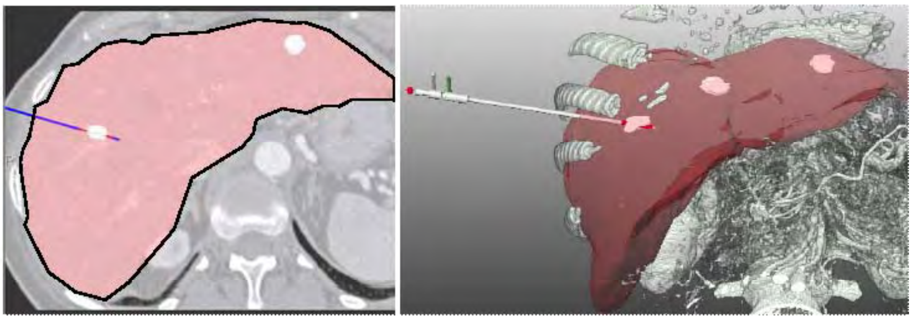


Fig. 2: Synchronized 2d and 3d visualizations of a CT liver data set with the segmented liver and three liver lesions (white). The visualization is used for the planning of a minimally-invasive therapy with a laser applicator. Radiological data kindly provided by the Technical University of Munich (Prof. Feussner)

While this general principle has been realized in several medical visualization systems, we have applied it consequently to all interactions frequently required for therapy planning ([16]). For example, 2d and 3d visualizations are employed for the placement of applicator devices used for minimally-invasive operations. The precise definition of the target point (the point where the applicator produces a maximum of energy to destroy the pathology around) is usually selected in 2d. The entry point (where the applicator penetrates the skin) might also be selected in a 2d visualization. Whether important structures are hurt by this access path, however, becomes more obvious in 3d visualizations (see Fig. 2). In a similar manner, 2d and 3d visualizations are combined for measurements, such as distances and angles for the quantitative analysis of spatial relations. Another frequent task in therapy planning is the specification of resection areas (those parts of the body which should be removed later in surgery). Resection specification can be accomplished in 2d (by drawing into selected slices and interpolating between them) and in 3d by moving surgical tools through the data which virtually remove tissue. Again, it is crucial to evaluate the virtual resection in the other view. If the resection area was specified in 2d, the 3d view shows its general shape – which is important to assess whether this shape can be resected. On the other hand, if the resection was specified in 3d, the 2d view is useful to asses the affected regions.

5 Provide useful defaults for visualization parameters

Visualizations, such as these presented in Figs. 1-2 are controlled by a variety of parameters. These include object colors, transparency values, the viewing direction as well as smoothness and quality parameters. It is essential that these parameters are available via the graphical user interface. However, it is tedious if they have to be specified again and again for individual objects. Hence, meaningful default values which can be accepted in the majority of the cases are crucial for the usability of such a system. How can these be specified? First it is important to classify objects, because such a classification provides a useful base for the assignment of default values. Examples of object classes are organs, bony structures, vascular structures and tumors. Organs are large structures: usually it is desirable to see details of the organ interior. Therefore, by default they are rendered semitransparently whereas lesions and vascular structures are not. Note, that different transparency values are useful for 2d and 3d visualizations. In 2d visualizations (Fig. 2, left) the original data should be visible to some extent. Therefore, even lesions and vascular structures are not displayed fully opaque. Large compact structures such as organs can be displayed with lower resolution compared to thin elongated structures.

Finally, and that is an aspect which was often discussed with medical doctors: the initial viewing direction should be a natural one, preferably the surgeon's view. The two images in Fig. 2 for example are generated without changing any visualization parameter. All colors and transparency values, the transfer functions for the volume rendering correspond to the default values. An individual user may have slightly different preferences. Therefore, it should be possible to customize visualizations by storing personalized default values. The benefit of well-selected default values are not only time-savings but also better quality and reproducibility of the resulting visualization.

6 Dynamic medical visualizations

The understanding of the complex spatial phenomena inside the human body requires the interactive manipulation of images. Dynamic visualizations – interactive visualizations with rapid feedback after continuous interaction – are particularly useful for this understanding because changes between images can be observed and need not to be interpreted (see [21] for a discussion of active and dynamic visualizations).

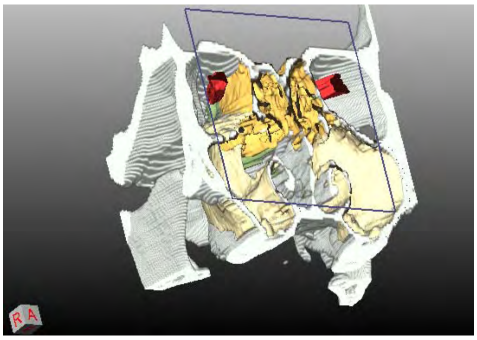


Fig. 3: Exploration of a CT data set of the human head. The continuous movement of the clipping plane is used to gain insight into the nasal cavities. The images were acquired to prepare surgery in this area. The spatial relations to critical structures, in particular the optical nerves, are essential. The cube in the lower left corner indicates the viewing direction. Radiological data kindly provided by University of Leipzig (Dr. Strauß).

There are several wide-spread interaction tasks in medical visualization which can be supported with dynamic interactions initiated with the pointing device on the image viewer (see Fig. 3). Clipping planes and clip boxes are translated and rotated in order to suppress parts of the underlying data. Radiologists are used to adapt brightness and contrast of volume rendered images by simply dragging the mouse (the *x*- and *y*-position of the mouse specify the parameters for the widely used ramp transfer functions). Again, the interaction can be initiated without moving the pointing device outside the visualization to some control element. Also, the traditional 2d slice view can be enhanced with dynamic interaction such that the displayed slice changes continuously as long as a certain combination of mouse buttons is pressed. This interaction is referred to as the cine-mode.

Another effective dynamic interaction facility is included in the VoxelMan (recall [9-11]). It is based on a sorting process of the object's relation to the skin. With a continuous mouse movement objects are hidden starting from those closer to the skin giving sight to more distant ones. Compared to the traditional interaction required to hide objects (select them individually and press some button to hide them) this is much faster and again allows to concentrate on the visualization.

7 Integration of anatomical context

Medical visualizations for sophisticated treatment planning procedures are based on prior image analysis. Examples are planning processes for total hip replacements [2, 8], facial surgery [4], neurosurgery [6] and liver surgery [19]. In the image analysis stage, important anatomical and pathological structures are identified and delineated (segmented). The segmented objects are usually visualized by means of surface rendering. With appropriate smoothing and shading this often yields easy-to-interpret visualizations.

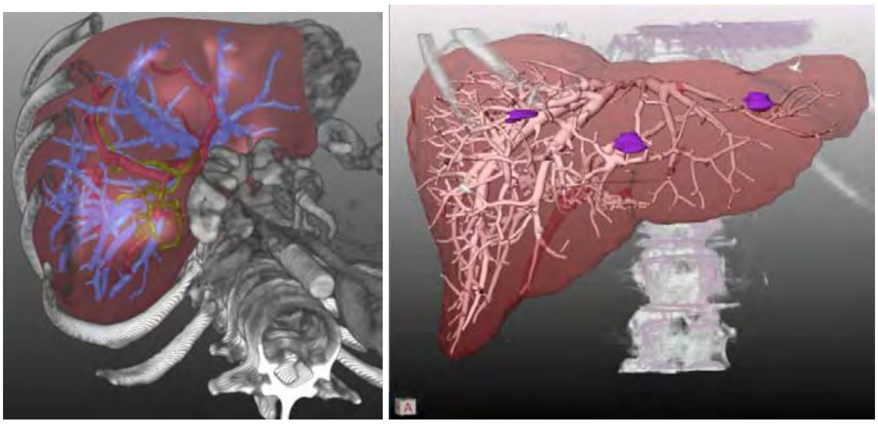


Fig. 4: Hybrid combination of the direct volume rendering of bony structures and surface rendering of intrahepatic structures (liver, vascular systems, tumors in the right view). The bony structures serve as anatomical context for the intrahepatic structures. Radiological data kindly provided by Medical School Hannover (Prof. Galanski).

Segmented objects might be selectively displayed or hidden to customize the visualization to the questions related to the therapeutic process at hand. However, the mere visualization of important structures is often insufficient. The lack of surrounding tissue, in particular of skeletal structures makes it very difficult to understand the current view. Therefore, medical doctors appreciate the visual integration of bony structures (usually displayed by means of volume rendering) and structures which are in the focus (see Fig. 4).

8 Model-based visualizations

Medical visualizations for clinical use have two fulfil two requirements:

1. they should strictly adhere to the underlying radiological data and
2. they should be easy to interpret.

Let us discuss these requirements with respect to vascular structures. These structures are often very thin, for example in the periphery of vascular trees. Due to the limited spatial resolution of CT and MR scanners these thin structures are often represented by only one or two voxels per slice. A straightforward visualization that fulfils the first requirement usually results in noisy visualizations due to the limited spatial resolution of radiological data. For example, it is very difficult, to assess the topology of a vascular tree based on such visualizations. This, however, is often crucial for surgery planning where risks of a procedure have to judged. As an example of this risk analysis it is important to recognize which branches of a vascular tree would be affected if a certain branch is cut. To support those questions, more abstract visualizations serve better. Therefore, several attempts have been made to reconstruct vascular structures based on a prior vessel skeletonization [5, 17]. In the skeletonization process, the medial axis inside the vessel as well as the minimum and maximum diameter for each skeleton voxel is derived (see [19] for details). This information can be used to visualize vascular trees by means of graphics primitives fitted along the skeleton path. As an example, we used truncated concatenated cones as the underlying primitives [7]. Special care was taken to provide smooth transitions of surface normals at branchings. Fig. 4 and Fig. 5 show examples of this vessel visualization method. This method has been developed and refined in fruitful discussions with radiologists and obviously is a good comprise between the desire to have precise *and* easy-to-interpret visualizations.

The degree of correspondence to the radiological data is adjustable by the number of cones that are generated. Cones can be generated between two subsequent voxels (highest level of correspondence) or only between branchings of the vascular tree (lowest level of correspondence). Also, the vessel diameter can adhere strictly to the values measured in the skeletonization process or be smoothed in order to reduce the effect of rounding errors (see Fig. 6). In summary, the degree of realism is adjustable with such model-based visualizations.

This method is based on the assumption that the cross-section of vascular structures is circular. This assumption, of course, is a simplification. It is not appropriate when vascular diseases should be diagnosed. However, for many therapy planning tasks it is crucial to understand the spatial relation between pathologic structures and adjacent vascular structures. In such cases these model-based visualizations are very helpful.[1] Similar model-based visualizations are used in the VOXELMAN (recall [9-11]) where B-spline shapes are fitted to selected points which represent nerves in some slices. These structures are so small that they appear in some slices and disappear in others.

[1] Some details of the described visualization are inspired by medical doctors. The visualization method is regularly used for liver surgery planning.

Using the anatomical knowledge that the nerves are connected and using knowledge about their principal shape appropriate visualizations can be generated with the reconstruction process described in [14].

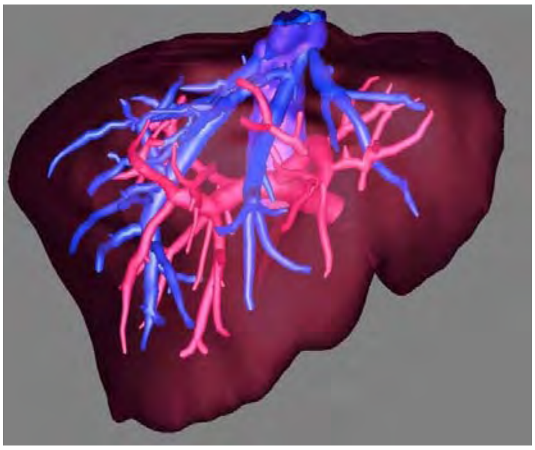


Fig. 5: High quality visualization of the intrahepatic vascular anatomy (hepatic and portal veins). Radiological data kindly provided by Medical School Hannover (Prof. Galanski).

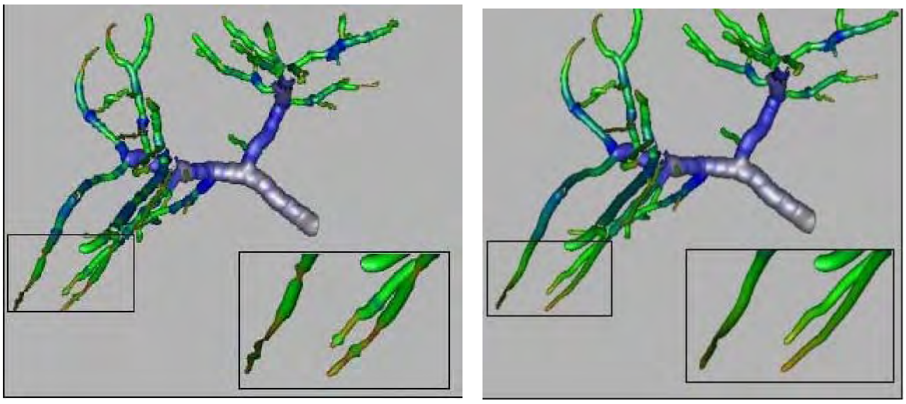


Fig. 6: Model-based vessel visualizations with different settings for the smoothing of the vessel diameter. The vessel diameter is color-coded. In the left image, no smoothing is applied which results in obvious discontinuities in the periphery. These discontinuities are due to the limited spatial resolution of the underlying data.

9 Emphasis in medical 3d visualizations

In therapy planning applications as well as in software for medical education it is often necessary to highlight an anatomic or pathologic structure. As an example, the user of such a system selects an object in a list via its name and the system should provide feedback emphasizing this object. Another obvious reason for the need of 3d emphasis techniques is due to the integration of 2d and 3d views (recall Sect. 4). After an object is selected in a 2d view it should be highlighted in the 3d view to support the connection between the two visualizations.

The emphasis of objects in 3d visualizations is difficult by its very nature. From the current viewing position objects might be too small to be recognizable or they might be occluded by other objects. In medical visualizations these problems are prevalent: objects are often concave, are at least partly occluded by others. Therefore simple emphasis techniques such as the use of a special colour for highlighted objects or blinking do not work well. Emphasis techniques therefore should ensure the visibility of the involved objects (see [18]). In principle, two strategies are possible to achieve visibility:

- the camera position might be changed to make the desired object visible or
- visualization parameters of occluding objects might be adapted to allow to look through them.

The first strategy, in general, cannot be recommended. A radical change of the camera position, not initiated by the user, is often not comfortable because the user has to interpret this change and perhaps dislikes the chosen perspective. Unnatural viewing directions may result. The change of visualization parameters, the second strategy, can be implemented by using transparency: occluding objects are rendered semitransparently to reveal objects behind. This is a viable approach, however showing apparent drawbacks. If several objects are in front of the object to be emphasized all occluding objects must be strongly transparent with the result that they are almost unrecognizable (see Fig. 7). As an alternative, fast silhouette generation algorithms might be used to enhance the visualization of the transparently rendered objects. For the sake of brevity, emphasis techniques could only be touched here. There is a variety of emphasis techniques suitable for medical visualization, including those based on the shadow generation for selected objects (see [15] for a review on such techniques).

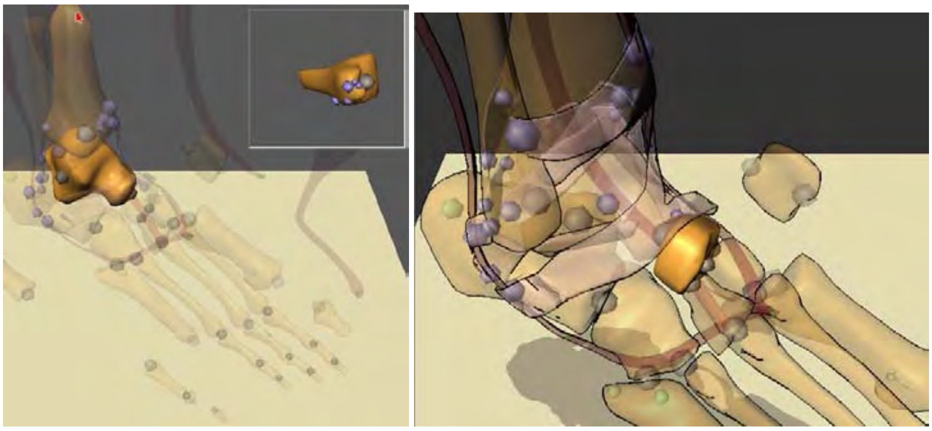


Fig. 7: Visualizations of bony structures and some muscles of the human foot as part of an educational system (the spheres represent docking positions to compose objects). A particular bone is emphasized using transparency of other objects (left). In the right view the shape of the transparent objects is emphasized with line drawing of their silhouette.

10 Concluding remarks

Smart medical visualizations integrate 2d and 3d visualizations with a bi-directional link in order to synchronize changes between both. 3d visualizations of the relevant anatomic and pathologic structures are integrated with the anatomic context which provides a reference. Appropriate visualization parameters depend on the category of the respective object; vascular structures should be rendered with other parameters than organs or pathologic lesions per default. A comprehensible visualization of thin elongated branching structures requires other parameters as large compact structures. Smart medical visualizations include predefined parameters for different categories. In particular for vascular structures a model-based reconstruction is useful for the visualization. Finally, smart medical visualizations are highly interactive. They support dynamic changes which are initiated with the pointing device on an image view.

Future Work. So far, non realistic renderings [22] have not been integrated in medical visualizations used in the clinical routine. With their ability to emphasize shape and structures, these algorithms are promising for such applications. Another wide area for future work is the incorporation of uncertainty visualization techniques. The image acquisition and image analysis steps which are carried out prior to visualization may produce imprecise or unreliable results. At least some of these "errors" might be estimated. For example, tumor segmentation is often imprecise because the lesions have similar intensity values to the surrounding tissue and apparently no gradient at their borders. Special visualization techniques should be developed to encode this uncertainty (see for example [13]). Finally, the development of emphasis techniques dedicated to the complex shapes in medical visualizations is an area where further research is needed. In particular, the appropriate technique should be automatically selected by the visualization system (taking into account object visibility, size and shape).

Acknowledgements. We want to thank our collaborators at MeVis: Dr. Holger Bourquain, Horst Hahn, Milo Hindennach, Arne Littmann, Felix Ritter, Andrea Schenk, and Wolf Spindler. In particular, Horst Hahn carefully commented on the paper. The paper is also based on many discussions with our clinical partners: we want to thank in particular Prof. Hubertus Feussner (University Munich), Prof. Michael Galanski (Medical School Hannover), Prof. Karl Oldhafer (General Hospital Celle) and Dr. Gero Strauß (University Leipzig).

References

[1] HK Çakmak, U Kühnapfel (2000). "Animation and Simulation Techniques for VR-Training Systems in Endoscopic Surgery", *Eurographics Workshop on Animation and Simulation* (EGCAS '2000), Interlaken/Switzerland, pp. 173-185

[2] J Ehrhardt, H. Handels, T Malina, B Strathmann, W Plötz, SJ Pöppl (2001). "Atlas based Segmentation of Bone Structures to Support the Virtual Planning of Hip Operations", *International Journal of Medical Informatics*, Vol. 64, pp. 439-447

[3] TT Elvins (1992). "A Survey of Algorithms for Volume Visualization", *Computer Graphics*, Vol. 26 (3), pp. 194-201

[4] PC Everett, EB Seldin, M Troulis, LB Kaban, R Kikinis (2000). "A 3-D System for Planning and Simulating Minimally-Invasive Distraction Osteogenesis of the Facial Skeleton", *Proc. of MICCAI*, Springer, LNCS Vol. 1935, pp. 1029-1039

[5] G Gerig, T Koller, G Székely, C Brechbühler, O Kübler (1993). “Symbolic Description of 3d structures applied to cerebral vessel tree obtained from MR angiography volume data”, *Proc. of Information Processing in Medical Imaging*, Springer, LNCS, Vol. 687: 94-111

[6] DT Gering, A Nabavi, R Kikinis et al. (1999). “An Integrated Visualization System for Surgical Planning and Guidance using Image Fusion and Interventional Imaging”, *Proc. of MICCAI*, Springer, LNCS Vol. 1679, pp. 809-819

[7] HK Hahn, B Preim, D Selle, HO Peitgen (2001). “Visualization and Interaction Techniques for the Exploration of Vascular Structures”, *IEEE Visualization* (San Diego, CA, Oct.), pp. 395-402

[8] H Handels, J Ehrhardt, B Strathmann, W Plötz, SJ Pöppl (2001). “An Orthopaedic Atlas for the 3D Operation Planning and the Virtual Construction of Endoprostheses”, *Computer Assisted Radiology and Surgery* (CARS 2001, Berlin), Elsevier, pp. 312-317

[9] KH Höhne, B Pflesser, A Pommert, M Riemer, T Schiemann, R Schubert, U Tiede (1995). “A new representation of knowledge concerning human anatomy and function”, *Nat. Med.*, Vol. 1 (6), pp. 506-511

[10] KH Höhne, B Pflesser, A Pommert et al. (2000). *VOXEL-MAN 3D Navigator: Inner Organs. Regional, Systemic and Radiological Anatomy*, Springer-Verlag Electronic Media, Heidelberg

[11] KH Höhne, A Petersik, B Pflesser et al. (2001). *VOXEL-MAN 3D Navigator: Brain and Skull. Regional, Functional and Radiological Anatomy*, Springer-Verlag Electronic Media, Heidelberg

[12] U Kühnapfel, HK Çakmak, H Maass (2000). “Endoscopic Surgery Training using Virtual Reality and deformable Tissue Simulation”, *Computers & Graphics*, Vol. 24, pp. 671-682, Elsevier

[13] AC Pang, CM Wittenbrink, SK Lodha (1997). “Approaches to Uncertainty Visualization”, *The Visual Computer*, Vol. 13 (8), pp. 370-390

[14] A Pommert, KH Höhne, B Pflesser et al. (2003). “Creating a high-resolution spatial/symbolic model of the inner organs based on the Visible Human”. In: *Yearbook of Medical Informatics 2003: Quality of Health Care: The Role of Informatics*, Schattauer, Stuttgart, pp. 530-537

[15] B Preim, F Ritter (2002). “Techniken zur Hervorhebung von Objekten in medizinischen 3d-Visualisierungen“, *Proc. of Simulation and Visualization 2002*, SCS, pp. 187-200

[16] B Preim, M Hindennach, W Spindler, A Schenk, A Littmann, HO Peitgen (2003). “Visualisierungs- und Interaktionstechniken für die Planung lokaler Therapien“, *Proc. of Simulation and Visualization 2003*, SCS, pp. 237-248

[17] A Puig, D Tost and I Navazo (1997). “An Interactive Cerebral Blood Vessel Exploration System”, *IEEE Visualization* (Phoenix, Arizona, October), pp. 433-436

[18] T Rist, E André (1992). “Incorporating Graphics Design and Realization into the Multimodal Presentation System WIP”, *Advanced Visual Interfaces (Proceedings of AVI '92, Rome, Italy)*, World Scientific Press, Singapore, pp. 193-207

[19] D Selle, B Preim, A Schenk, HO Peitgen (2002). “Analysis of Vasculature for Liver Surgery Planning “, *IEEE Transactions on Medical Imaging*, Vol. 21 (11), pp. 1344-1357

[20] B Shneiderman (1997). *Designing the User Interface*, Addison Wesley

[21] R Spence (2001). *Information Visualization*, Addison Wesley

[22] T Strothotte, S Schlechtweg (2002). *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*, Morgan Kaufmann, San Francisco

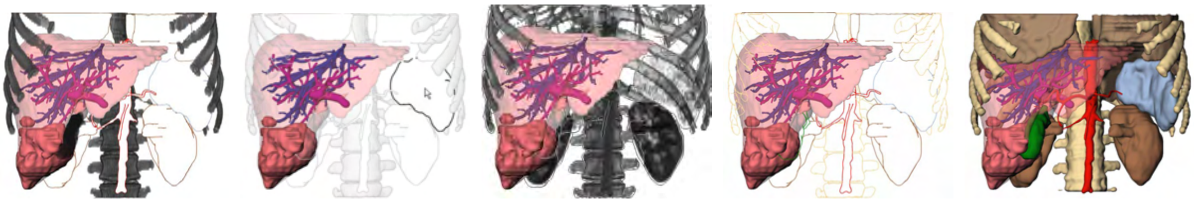
EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005)
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning

Christian Tietjen[1] Tobias Isenberg[1,2] Bernhard Preim[1]

[1] Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg, Germany
{tietjen|preim}@isg.cs.uni-magdeburg.de

[2] Department of Computer Science
University of Calgary, Canada
isenberg@cpsc.ucalgary.ca



Abstract

We introduce a flexible combination of volume, surface, and line rendering. We employ object-based edge detection because this allows a flexible parametrization of the generated lines. Our techniques were developed mainly for medical applications using segmented patient-individual volume datasets. In addition, we present an evaluation of the generated visualizations with 8 medical professionals and 25 laypersons. Integration of lines in conventional rendering turned out to be appropriate.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

Keywords: Medical visualization, non-photorealistic rendering, hybrid styles, line rendering.

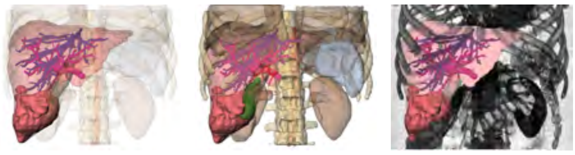
1. Introduction

Direct and indirect (isosurface) volume rendering of CT or MRI datasets dominate in application areas such as diagnosis and surgery planning. Direct volume rendering (DVR) is used for a fast overview, e. g., if complex fractures are involved or when structures should be displayed that do not exhibit (sharp) surfaces. However, using DVR it is often difficult to emphasize objects or their parts. Advanced DVR methods relying on multidimensional transfer functions are better suited for emphasis but they exhibit a huge parameter space that is usually not appropriate for clinical applications.

In contrast to DVR, surface rendering transforms a part of the volume data into a polygonal representation. This is accomplished either as threshold-based isosurface rendering or as surface rendering of segmentation results. The flexibility to adjust surface visualizations is reduced to emphasis with color and transparency. In particular, transparency easily introduces visual clutter (Figure 1). The hybrid combination of surface and volume visualization is often useful: Surface visualization is employed to show anatomic structures which have been segmented in advance, whereas DVR is employed to present anatomic context such as skeletal structures.

Inspired by traditional depictions, e. g., in medical atlases non-photorealistic rendering (NPR) techniques [SS02] emerged to produce comprehensible renditions. NPR techniques range from those line drawings inspired from traditional artwork (such as silhouettes and hatching) to methods derived from DVR (such as tone shading). So far, mostly individual techniques were used—either traditional (e. g., DVR or surface rendering) or NPR methods. In this paper, we describe the integration of line drawings with surface and volume visualization to create more comprehensible renditions. In particular, we discuss the problems that arise when these are used together in an OPEN INVENTOR based scene graph architecture [Wer94].

We rely on segmented data where the relevant anatomic



(a) Too much transparency. (b) Too little transparency. (c) Badly differentiable.

Figure 1: *3D visualization with surface rendering and DVR: context structures (bones, kidneys, gallbladder, lung, and aorta) are not differentiable or hiding the focus. The focused liver includes a tumor, the portal vein, and the hepatic vein.*

structures are delineated. This makes it possible to generate renderings of particular objects. Applications of such hybrid visualizations lie in therapy planning and educational systems. In addition, hybrid visualizations are intended to explain a planned intervention to patients. For these applications, it is essential to employ patient-individual datasets that have a reduced resolution and a higher noise level than artificial datasets such as the VISIBLE HUMAN [SASW96].

In order to confirm the effectiveness of the created visualizations, we carried out a user study. In this study, we presented the renditions to different groups that represent the different possible potential users of the visualizations: medical professionals and medical laypersons.

The remainder of this paper is organized as follows. Section 2 describes related work. Afterwards, we discuss the combination of the three mentioned rendering methods in Section 3. Section 4 then presents the user study we carried out to compare the usefulness of the different visualizations. Finally, in Section 5 we conclude the paper.

2. Related Work

The main source of inspiration for our work comes from medical atlases (e. g., [Rog92]) where line primitives are often used to depict form and to distinguish objects from each other. This is motivated by research in perception. For example, WARE discusses that silhouette lines are essential for figure-ground segregation, for the fast recognition of objects, and for the identification of object structure [War04].

The development of non-photorealistic rendering styles [SS02] to generate images similar to those in medical atlases started approximately in 1990 [ST90]. It is interesting to note that NPR has very similar goals as scientific visualization, namely to convey information effectively and to emphasize features [NSW02]. Thus, it is not surprising that the visualization community adopts techniques from NPR. For the first time this has been done in a system for radiation treatment planning [LFP*90].

The term *non-photorealistic volume rendering*—or as RHEINGANS and EBERT suggest *volume illustration*—refers to visualization techniques that are similar to surface-based NPR but are applied to volume data [RE01]. In the context of medical volume data we prefer the term volume illustration because there is no photorealistic rendering option for a clinical CT dataset.

One class of volume illustration techniques operates on the original data without any segmentation information. The gradient direction is estimated based on differences between adjacent voxels. Silhouette lines are generated where the gradient magnitude is large and the cross product of camera direction and the gradient direction is close to 0. [CMH*01] presented expressive visualizations where the large gradient between air and skin is employed for silhouette generation. [LM02] introduced an efficient implementation of volume illustration that exploits commodity graphics hardware. However, for many regions in the human body contrasts in CT and MRI data is considerably lower than at the air-skin boundary and silhouettes cannot be identified easily.

[SE04] calculate silhouettes from volumetric data using implicit trivariate tensor product B-spline functions that approximate the data. A subdivision method is used to calculate the silhouettes that have a superior look compared to voxel-based silhouette extraction schemes. Convincing results were achieved recently by KINDLMANN et al. who employed curvature information to guide the placement of silhouette lines [KWTM03]. Transfer functions (TF) could be specified with the first and second main curvature as the two-dimensional domain for TF specification. They also generate contours and can control their thickness in image-space by considering curvature information.

Hatchings based on volume information allow to produce smoother hatching lines compared to purely surface-based methods [DCLK03]. [NSW02] introduce concepts for hatching volume data based on curvature information which is extracted near user-selected seed points. Hatching is modified by transparency values and lighting conditions. They argue not to combine DVR and line drawing in order to allow for flexible stylization.

For clinical applications it is more appropriate to rely on segmentation information and to deliberately assign rendering styles to anatomic structures or categories such as nerves or muscles. [HBH03] combined volume illustration techniques with surface rendering and DVR in a two-pass approach. This is achieved by applying local TFs to different objects using graphics hardware shader.

A recent approach presented by YUAN and CHEN [YC04] combines DVR with surface shading and also adds silhouettes and feature lines. However, their approach is based on image-space silhouette and feature line detection and is, therefore, limited to image-processing stylization. Our approach, in contrast, uses object-space stroke extraction that allows more freedom to parameterize the rendered lines.

[VKG04] also employed segmentation information in order to specify the appearance of individual objects. In addition, they integrated illustration techniques such as cut-away views. The general concept of this work is to derive the importance of an object (e. g., from input by the user) and to project the most important objects onto the screen. All rendering styles are provided. However, they are integrated in a DVR which limits the flexibility of line stylization.

3. Combining Silhouettes, Surface Shading and DVR

In this section we describe a rendering process that allows to combine the three visualization techniques named above to create hybrid renditions. We integrate these techniques using an OPEN INVENTOR scene graph architecture which allows us to reuse nodes that affect visualization parameters at different positions in the same scene graph. This not only generates a more consistent visualization but also is more flexible in terms of stylizing individual objects.

3.1. Initial Considerations

The different rendering styles are not equally well suited to visualize objects. Thus, they will be used to depict different structures according to their relevance for the visualization (see Figure 7). With respect to relevance, in the following we will refer to three types of structures or objects:

Focus objects (FO): objects in the center of interest are emphasized in a particular way.
Near focus objects (NFO): important objects for the understanding of the functional interrelation or spatial location. Their visualization depends on the particular problem.
Context objects (CO): all other objects.

Line stylization—when carefully applied—may be used to control whether objects are conceived as FO or CO. One type of stylization that we include is color since this is widely used to emphasize objects or structures. In addition, we will depict hidden and visible lines in a different manner in order to convey the spatial relationships.

For the efficient extraction of lines and strokes from 3D meshes, a Winged Edge data structure is employed to store local connectivity information. This data structure has to be created as a pre-computation step for the individual objects. In order to render and stylize these objects, each object's mesh data is subsequently used to extract, e. g., the lines, and stylize them. Therefore, the use of stylization pipelines that successively modify the line style data is required. The stylization process is divided into several small steps represented by nodes in the scene graph. This concept also allows the reuse of certain line stylization pipelines because the respective nodes may be linked into the scene graph at several positions. This ensures a coherent appearance of the objects that use the same pipeline.

In order to use DVR, we integrate this rendering technique into one scene graph. This is achieved by specialized DVR node coupled with a TF node. However, since DVR always renders the whole volume into the z-buffer, it is not possible to add surface shading afterwards. Thus, the sequence in which the modules for the individual rendering techniques are added to the rendering pipeline will be important.

3.2. Hybrid Rendering

For surface shading the OPEN INVENTOR architecture relies on normal z-buffer rendering. Thus, no special order of nodes is required for a correct rendering of the scene. Due to the nature of the remaining two techniques, the resulting scene graph may get fairly complex. Therefore, we start by discussing the extension of the scene graph for surface shading to include DVR. Then, we will show how line rendering can be added and explain the required individual modifications. Our rendering process is based upon the scene graph architecture of OPEN INVENTOR. In addition, we use the OPENNPAR system that extends OPEN INVENTOR and adds line extraction and line stylization capabilities [HIR*03]. For easy scene graph manipulations as well as DVR, we employ the MEVISLAB system [HLP03].

Direct volume rendering. DVR may be combined with surface rendering by adding the respective nodes to the scene graph. However, the DVR node fills the z-buffer for the entire volume that is rendered regardless of the TF as explained above. Therefore, it has to be added to the scene graph after the surface shading has been completed, i. e., as the last node in the scene graph traversal. Otherwise, the surface objects would not be rendered because all would fail the z-buffer test. Hence, after the DVR, the z-buffer contains no more sensible depth information. Furthermore, DVR may previously be modified according to segmentation results as shown in Figure 2. The bit mask of segmented areas is used for the purpose of displaying or hiding user-selected objects.

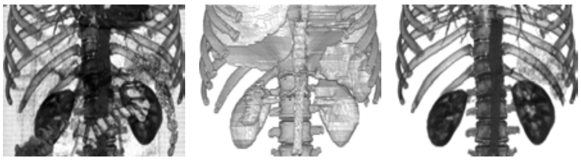

(a) Original DVR. (b) Bitmask from segmentation. (c) Masked DVR.

Figure 2: *The original volume has been segmented in advance. Thus, it may be restricted to bones, gallbladder, aorta, kidneys, milt and lung. The gallbladder is not visible because of the chosen transfer function.*

Silhouette rendering. The object-space line rendering approach that we employ comprises the following steps: geometry data preparation, line extraction and stroke generation, hidden line removal, and stroke stylization (see Figure 3).

To facilitate the following stages, first a geometry data structure is created that provides adjacency information. Then, after extracting the significant edges (silhouettes and feature lines), these edges are assembled into strokes and are processed in a stroke pipeline for stylization. As an essential step in this pipeline hidden lines have to be removed. A fast and simple method to solve this problem is illustrated in Figures 4(a) to (c). First, the objects are rendered into the z-buffer (while the frame buffer remains unchanged). In a second step, all extracted lines are scan-converted individually and stepwise classified as hidden or visible using the previously generated z-buffer data by locally comparing z-depths [IHS02]. Then, stylization may be applied such as changing the stroke's width, saturation, color, and texture. Further methods for line stylization such as cut away views or lighted regions may also be applied. In fact, the edge extraction and stroke generation is independent from the final stroke rendering because the data in the stroke pipeline is not altered by any of the other rendering steps. We will use this fact later on for combining all three rendering techniques.

If surface rendering is used in addition to the line graphics, the surface objects have to be rendered into the final image prior to the lines. Silhouettes being located exactly at discontinuities of the z-buffer of the surface objects. One side of each generated and stylized line would otherwise be overwritten by the surface object since they are typically more than one pixel wide. This is also the reason why there has to be an explicit hidden line removal (HLR) for the computed strokes before starting the final rendering process. In addition, the z-buffer generated for HLR would interfere with a correct rendering of the surface objects.

However, this approach is only applicable for opaque objects because transparent objects usually do not change the z-buffer. Thus, lines that lie behind a transparent object would not be removed. In order to prevent distant lines to be rendered on top of closer transparent model parts, the z-buffer rendering must be carried out for the transparent objects as well. Unfortunately, this conflicts with the regular shading technique for transparent objects.

Combination of rendering styles. According to the discussion above, the line extraction and visibility classification has to occur before rendering surface objects. Also, the DVR has to be performed after all surface objects have been drawn. However, rendering surfaces on top of the stylized lines would potentially overwrite parts of the rendered lines as explained above. Fortunately, we can make use of the line extraction, classification, and storage being independent from the process of rendering the line into the frame-buffer as suggested above. Therefore, we use the following procedure for generating the hybrid rendition:

1. generate the z-buffer for surface and line objects (including transparent objects),
2. extract lines from internal mesh representation,
3. determine line visibility according to the z-buffer,
4. clear the z-buffer,
5. render surface objects using z-buffering,
6. render stylized lines with writing z-buffer data but without doing the z-buffer test, and
7. render volume using z-buffering.

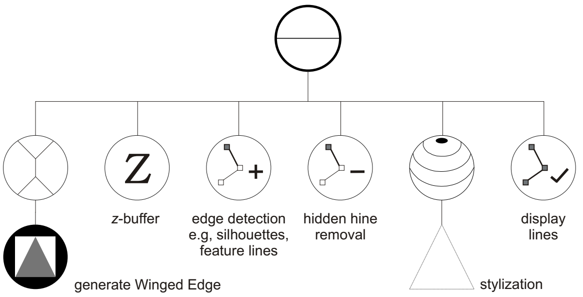


Figure 3: *Scene graph for silhouette rendering.*

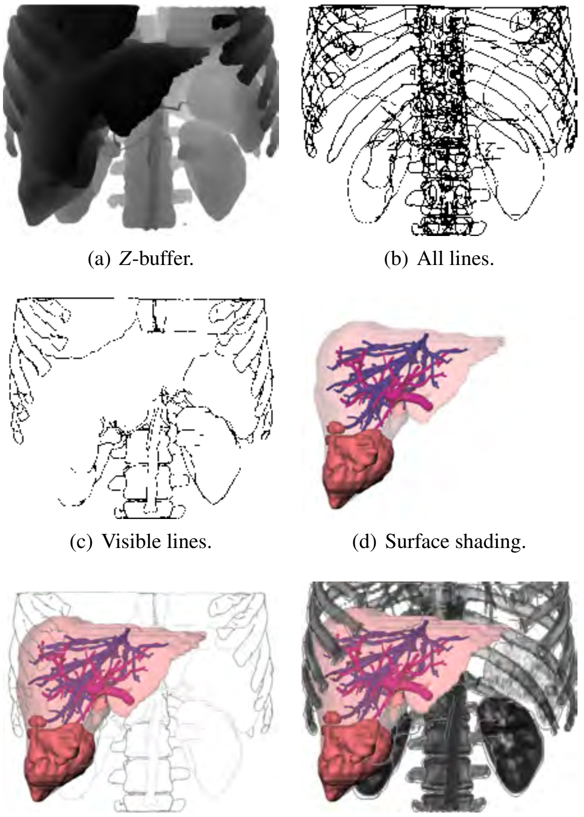

(a) *Z*-buffer. (b) All lines.
(c) Visible lines. (d) Surface shading.
(e) Lines & shading. (f) The resulting hybrid image.

Figure 4: *Sequence for combining all rendering styles.*

The z-buffer of the surface objects and the line objects is rendered first (Figure 4(a)). The z-buffer is generated for all objects regardless whether they are transparent or opaque. Thereafter, the lines are generated (Figure 4(b)) and HLR is performed using the z-buffer information (Figure 4(c)). Because the line data is stored separately, it is not affected by

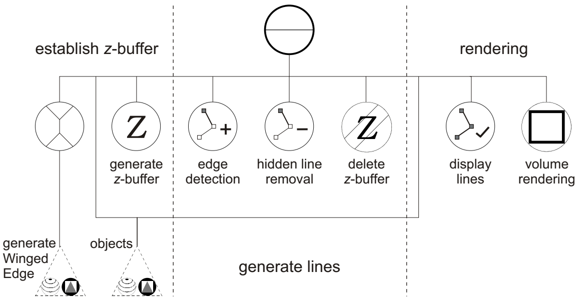


Figure 5: *Scene graph for combining all rendering styles.*

surface and volume rendering. Thus, it can be determined this early in the process. After the line extraction, the generated z-buffer is not needed anymore because it also contains transparent and line objects and becomes deleted.

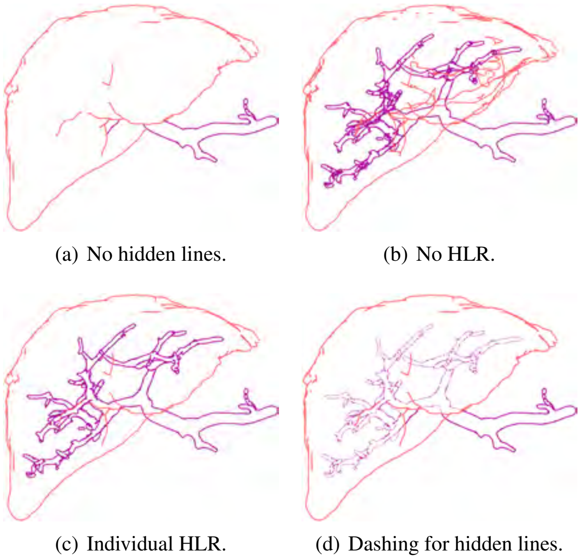
Now, the surface rendering is initiated (Figure 4(d)). Since we included transparent objects in the initial z-buffer rendering, there will be no lines that will mistakenly be drawn on top of them. Due to the separate storage of the stroke data, the lines can be displayed with correct depth information. For this purpose, the line rendering is performed without z-buffer test but with writing z-buffer data (Figure 4(e)). DVR is carried out as the last step and after the lines because now the line data is present in the z-buffer as well (Figure 4(f)). Figure 5 shows the scene graph for the entire process.

Removing self-occluding lines. In some cases it might be useful to show line drawings behind other line drawings. However, at the same time the hidden lines of both the front and back objects have to remain hidden so that the rendition does not get confusing. Unfortunately, the process discussed so far does not allow this. If both objects are rendered into the z-buffer simultaneously, it is not possible to distinguish between lines that are self-occluding and those that are hidden by a different object.

This problem can be solved by rendering the z-buffer for each object separately such that the self-occluding lines can be removed individually for each object (see Figure 6). The disadvantage of this procedure is that after each individual HLR it is necessary to clear the z-buffer. However, the produced effect (see Figure 6(d)) illustrates the spatial relationships and improves the understanding of the geometry.

4. Evaluation

In order to analyze whether line rendering is a meaningful extension to the existing visualization techniques, we carried out a user study. In addition to that, we have analyzed the answers to extract parameters for useful hybrid renditions combining DVR, surface shading, and line rendering. However, the goal for this task was not to find the 'best' visualization but several appropriate ones for the specific dataset since the quality of visualizations strongly depends on the domain.



(a) No hidden lines.

(b) No HLR.

(c) Individual HLR.

(d) Dashing for hidden lines.

Figure 6: *Removing self-occluding lines. In (a) the rendering of hidden lines is prevented due to a collective z-buffer. Individual HLR (c) solves this problem. Additional dashing and thinner hidden lines (d) produce an even stronger effect.*

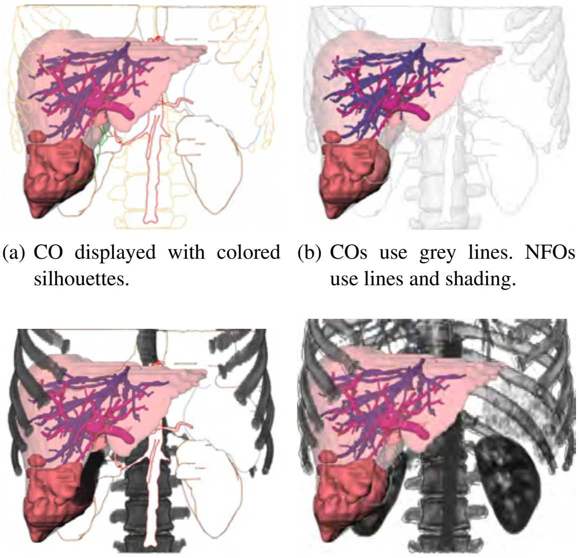
In cases where structures have to be displayed as COs, traditional illustrations typically use silhouettes and feature lines. We hypothesize that line renderings are preferred for the joint visualization of COs, NFOs and FOs. We also hypothesize, that a stylization with colors is most satisfying.

4.1. Evaluated Application Domains

The visualizations shown in this paper were generated for an educational system, the LIVERSURGERYTRAINER [BMOP04]. In this training system, users are asked, for example, to decide whether the patient could be safely operated and specify a resection line. For this purpose, comprehensible renderings of the intrahepatic anatomy (vascular structures, liver tumors) are necessary. In order to provide a realistic setting, the system employs clinical datasets. The relevant anatomic and pathologic structures are segmented to permit selective visualizations.

Context visualization. There are many possibilities for the visualization of COs. The shape of the structure to be visualized will finally prove the most appropriate variant (Figure 7). DVR usually looks unnatural and is difficult to interpret. On the other hand, DVR shows the original data and is able to show non-segmented structures.

Simplifying complex visualizations. In complex scenarios, only few viewing directions allow to appraise how all objects are situated to each other. By providing interaction means to view the scenario from all directions, viewers watch it also from other directions. By employing silhouettes we are able



(a) CO displayed with colored silhouettes.

(b) COs use grey lines. NFOs use lines and shading.

(c) NFOs use DVR while COs are shown using colored lines on a white background.

(d) COs and NFOs are shown using DVR. NFOs also have lines added.

Figure 7: *A number of possibilities to visualize COs, NFOs, and FOs differently. The FO is the liver in all visualizations. NFO are the bones and the gallbladder. Figure 7(c) and 7(d) were not part of the evaluation.*

to achieve a plainer visualization for suboptimal directions that is easier to understand.

Figure 8 illustrates this application. A liver is divided into several vascular territories. To eliminate a liver tumor, in most cases all involved territories have to be removed. For planning purposes all vascular territories, the tumor, and the portal vein have to be visualized. Typically, visualizations as in the right of Figure 8 are used. They can be simplified by rendering the affected or healthy vascular territories using colored silhouettes.

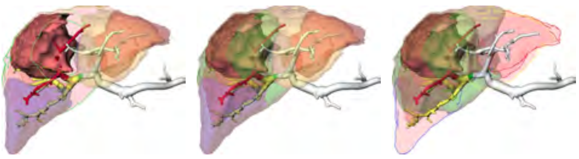
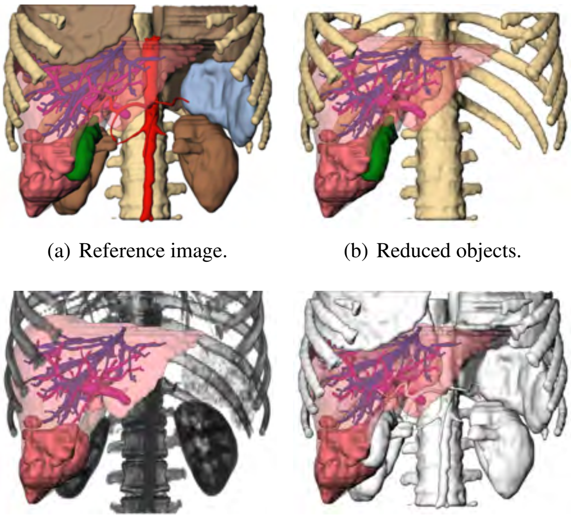


Figure 8: *Different coronal visualizations of the affected vascular territories. Left to right: affected vascular territories displayed via silhouettes, all segments transparent, and healthy territories via silhouettes.*

4.2. Study Subjects

The presented visualization process was developed for medical doctors who are familiar with medical atlases. Therefore we asked them to compare and evaluate the quality of the visualizations. In addition, the visualizations were shown to medical laypersons. This second survey has practical relevance because patients usually receive a pre-surgery consultation. Comprehensible 3D visualizations may be employed in such a consultation.



(a) Reference image.

(b) Reduced objects.

(c) Masked DVR.

(d) b/w shading.

Figure 9: *Selected variants without lines. Other visualizations shown are transparent versions of Figure (a) and (b).*

4.3. Methodology

Based on a CT dataset, we generated a representative repertory of possible visualizations. In computer-assisted surgery, surface visualizations and DVR are common (see Figure 9). These visualizations were compared with hybrid renditions using silhouettes. Considering the pros and cons of each visualization for specific tasks such as using more or less context to show the FO, the subjects were asked for their preference. To obtain meaningful answers, we had to narrow down the specific application domain and chose liver surgery. To get as many as possible answers, the questionnaires were sent in printed form. The chosen viewing direction is the 'view of the surgeon', meaning the perspective of the surgeon on the patient during the surgery.

4.4. Questionnaire Assembly

Our questionnaire is guided by principles described in [Shn97]. Every page of the questionnaire uses the same pattern. On top, two visualizations of the same anatomic structures are shown. Below these images multiple choice questions are placed. First, one of the two presented pictures had to be chosen. This question asked for the personal preference without closer examination whether the image is particulary

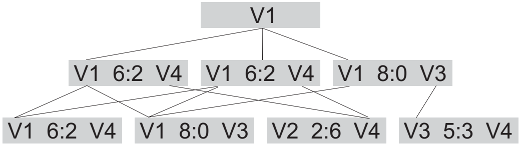


Figure 10: *Extraction of the decision tree for the medical doctors after critical examination. V1 refers to Figure 7(a), V2 to Figure 9(b), V3 to Figure 4(e), and V4 to Figure 9(a). From these Figure 7(a) was voted to be best.*

suited to solve certain problems. Subsequently, specific questions were asked to encourage the respondent to comment on the usefulness of the visualizations. In these questions a specific problem had to be assessed on a five value scale (for example ranging from 'very clearly arranged' to 'very complex'). Finally, the subjects were asked to specify the variant they would prefer for surgery education.

To classify the answers, we asked for some personal data: age, gender, education level, and personal skills with PCs as well as with 3D applications. The laypersons were introduced into the necessary medical knowledge by an additional preamble.

4.5. Analysis

For every pair of pictures on each page of the questionnaire the number of votes was counted. Due to the assembly, it could be determined which one of two images seems to be more suitable. The reference image (see Figure 9(a)) was compared with all other images. The remaining images were compared with only a few others to get a cross validation for the final results (shown in Figure 10). All in all 9 images for context visualization on 11 pages and 3 images for simplifying visualizations on one page are compared. Comparing all images with each other would have resulted in too many comparisons. By comparing the winners of each pair, we determined the most optical appealing and the most appropriate visualization for a given task since in medical training environments both aspects are important.

4.6. Interpretation

We received 33 filled out questionnaires in total. Eight sheets were returned by surgeons (2 female and 6 male) and 25 by medical laypersons (11 female and 14 male). The average age of the medical professionals is 42.8 years (ranging from 29 to 50) and they have medium skills with PCs and 3D applications. The average medical layperson is 25.3 years old (ranging from 23 to 27) and has very good skills with PCs and 3D applications.

Because only eight sheets were received by surgeons, it was difficult to extract significant results. In general, medical doctors settled upon less context information but noted that basic information about all COs has to be available all the time. In addition, the application of silhouettes seems to be appropriate for surgery planning. The visualization of the affected vascular territories using silhouettes was regarded as appropriate by six out of eight medical professionals. The two images that were favored by five of eight surgeons are shown in Figure 7(a) and 7(b). Our observations indicate a tendency, that surgeons prefer little context information. For this reason, the Figures 9(c) and 9(d) are also good candidates.

It was not possible to draw a distinct conclusion for the laypersons. This may be due to the novelty of the presented visualizations for the respondents. I. e., the reference image (Figure 9(a)) was compared to the DVR, the transparent, and the hybrid visualization. For the DVR and the transparent surface shading, no significant tendency was registered. Among all images which include silhouettes, 75% favored Figure 7(a). 83% favored Figure 7(a) compared to the transparent visualization. Almost all laypersons favored the silhouette image with additional surface shading.

Our results indicate that silhouettes are an appropriate extension of the existing visualization possibilities. In a direct comparison between the transparent surface and the hybrid visualization, the silhouettes are regarded as superior. The exclusive application of silhouettes without any further information about shape or color of the objects was regarded as unfavorable by all respondents. However, with additional information such as colored silhouettes or highly transparent surfaces, the silhouettes were rated to be very good.

5. Conclusion & Future Work

In this paper we described a scene graph based combination of silhouette, surface, and volume rendering. It shows that using a scene graph architecture facilitates a flexible and parameterizable integration of these existing rendering components. Stroke extraction and stroke rendering are decoupled from each other to allow for the display of correctly determined visible lines with stylization applied. Existing techniques such as using one or several semi-transparent layers of shaded objects may be used as well. One limitation is that we cannot render semi-transparent shading with the VR shining through. This would require to render the volume first which is not possible with the proposed pipeline. The combination of silhouette rendering with the traditional rendering styles for medical visualization has great potential. The integration of silhouettes allows to convey complex spatial relations more clearly which is essential for therapy planning and medical education. We collected feedback from medical professionals and laypersons indicating that hybrid renderings including silhouettes are appropriate to convey spatial information based on clinical data.

A great deal of work still needs to be done. Illustration techniques such as hatching and stippling should be integrated in a similar object-based manner. The interaction to adjust hybrid medical visualizations should be redesigned to reduce the interaction effort. In [KTH*05], applications of hybrid renderings for a special area of surgery planning are described.

Acknowledgments

Special thanks go to Angela Brennecke and Sebastian Mirschel. The liver data sets were provided by Prof. Dr. Karl J. Oldhafer of the Allgemeines Krankenhaus Celle, Germany. Prof. Oldhafer substantially supported the user study. Thanks also to Wolf Spindler for providing the volume renderer. The high-quality vessel visualizations in this paper are created by [OP05]. This work was carried out in the framework of a project supported by the Deutsche Forschungsgemeinschaft (DFG) (Priority Programme 1124, PR 660/3-1).

References

[BMOP04] BADE R., MIRSCHEL S., OLDHAFER K. J., PREIM B.: Ein fallbasiertes Lernsystem für die Behandlung von Lebertumoren. In *Bildverarbeitung für die Medizin* (Heidelberg, 2004), Springer Verlag, pp. 438–442.

[CMH*01] CSÉBFALVI B., MROZ L., HAUSER H., KÖNIG A., GRÖLLER E.: Fast Visualization of Object Contours by Non-Photorealistic Volume Rendering. *Computer Graphics Forum 21*, 3 (Sept. 2001), 452–460.

[DCLK03] DONG F., CLAPWORTHY G. J., LIN H., KROKOS M. A.: Nonphotorealistic Rendering of Medical Volume Data. *IEEE Computer Graphics & Applications 23*, 4 (July/Aug. 2003), 44–52.

[HBH03] HADWIGER M., BERGER C., HAUSER H.: High-Quality Two-Level Volume Rendering of Segmented Data Sets on Consumer Graphics Hardware. In *Proc. of IEEE Visualization* (2003), IEEE, pp. 301–308.

[HIR*03] HALPER N., ISENBERG T., RITTER F., FREUDENBERG B., MERUVIA O., SCHLECHTWEG S., STROTHOTTE T.: OpenNPAR: A System for Developing, Programming, and Designing Non-Photorealistic Animation and Rendering. In *Proc. of Pacific Graphics* (Los Alamitos, 2003), IEEE, pp. 424–428.

[HLP03] HAHN H. K., LINK F., PEITGEN H.-O.: Concepts for Rapid Application Prototyping in Medical Image Analysis and Visualization. In *Simulation und Visualisierung* (2003), SCS, pp. 283–298.

[IHS02] ISENBERG T., HALPER N., STROTHOTTE T.: Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes. *Computer Graphics Forum 21*, 3 (Sept. 2002), 249–258.

[KTH*05] KRÜGER A., TIETJEN C., HINTZE J., PREIM B., HERTEL I., STRAUSS G.: Interactive Visualization for Neck-Dissection Planning. In *Proc. of EuroVis* (2005), Eurographics Association, pp. 295–302.

[KWTM03] KINDLMANN G. L., WHITAKER R., TASDIZEN T., MÖLLER T.: Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proc. of IEEE Visualization* (2003), pp. 513–520.

[LFP*90] LEVOY M., FUCHS H., PIZER S. M., ROSENMAN J., CHANEY E. L., SHEROUSE G. W., INTERRANTE V., KIEL J.: Volume Rendering in Radiation Treatment Planning. In *Visualization in Biomedical Computing* (Los Alamitos, 1990), IEEE, pp. 4–10.

[LM02] LUM E. B., MA K.-L.: Hardware-Accelerated Parallel Non-Photorealistic Volume Rendering. In *Proc. NPAR* (2002), ACM Press, pp. 67–74.

[NSW02] NAGY Z., SCHNEIDER J., WESTERMANN R.: Interactive Volume Illustration. In *Proc. of Vision, Modelling and Visualization* (2002), pp. 497–504.

[OP05] OELTZE S., PREIM B.: Visualization of Vasculature With Convolution Surfaces: Method, Validation and Evaluation. *IEEE Transactions on Medical Imaging 24*, 4 (Apr. 2005), 540–548.

[RE01] RHEINGANS P., EBERT D.: Volume Illustration: Nonphotorealistic Rendering of Volume Models. *IEEE Transactions on Visualization and Computer Graphics 7*, 3 (July–Sept. 2001), 253–264.

[Rog92] ROGERS A. W.: *Textbook of Anatomy*. Churchill Livingstone, 1992.

[SASW96] SPITZER S., ACKERMAN M., SCHERZINGER A., WHITLOCK D.: The Visible Human Male: A Technical Report. *Journal of the American Medical Informatics Association 3*, 2 (Mar./Apr. 1996), 118–130.

[SE04] SCHEIN S., ELBER G.: Adaptive Extraction and Visualization of Silhouette Curves from Volumetric Datasets. *The Visual Computer 20*, 4 (June 2004), 243–252.

[Shn97] SHNEIDERMAN B.: *Designing the User Interface*, 3rd ed. Pearson Addison Wesley, 1997.

[SS02] STROTHOTTE T., SCHLECHTWEG S.: *Non-Photorealistic Computer Graphics. Modelling, Animation and Rendering*. Morgan Kaufmann, 2002.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible Rendering of 3-D Shapes. In *Proc. of SIGGRAPH* (1990), ACM Press, pp. 197–206.

[VKG04] VIOLA I., KANITSAR A., GRÖLLER M. E.: Importance-Driven Volume Rendering. In *Proc. of IEEE Visualization* (Los Alamitos, 2004), IEEE, pp. 139–145.

[War04] WARE C.: *Information Visualization*, 2nd ed. Morgan Kaufmann, 2004.

[Wer94] WERNECKE J.: *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Rel. 2.* Addison-Wesley, 1994.

[YC04] YUAN X., CHEN B.: Illustrating Surfaces in Volume. In *Proc. of VisSym* (2004), Eurographics, pp. 9–16.

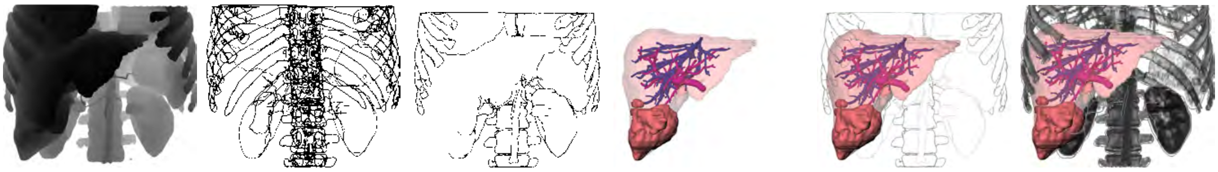


Figure 4: *Sequence for combining all rendering styles.*

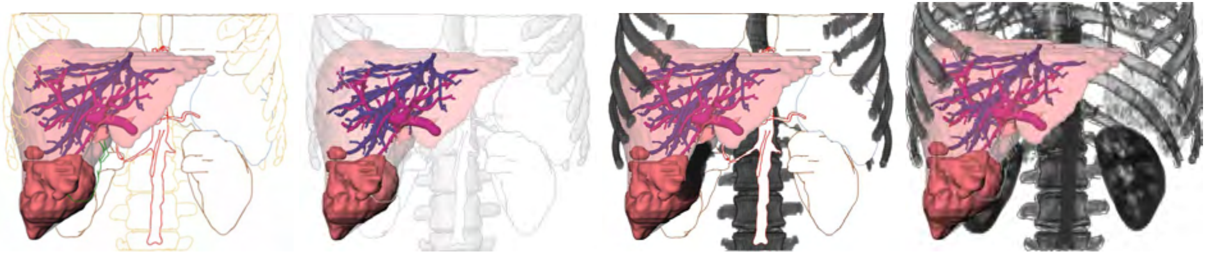


Figure 7: *A number of possibilities to visualize COs, NFOs, and FOs differently.*

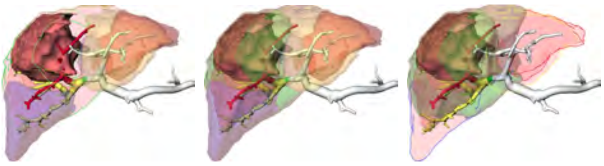


Figure 8: *Different coronal visualizations of the liver.*

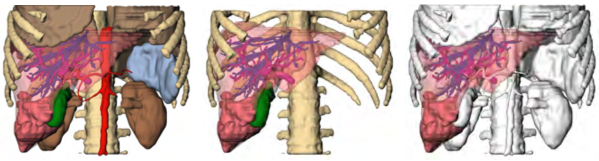


Figure 9: *Selected variants without lines.*

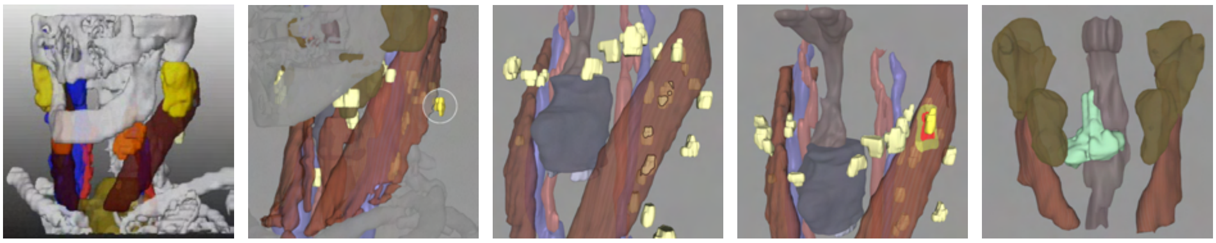
EUROGRAPHICS - IEEE VGTC Symposium on Visualization (2005)
K. W. Brodlie, D. J. Duke, K. I. Joy (Editors)

Interactive Visualization for Neck-Dissection Planning

Arno Krüger[1] Christian Tietjen[1] Jana Hintze[1] Bernhard Preim[1] Ilka Hertel[2] Gero Strauß[2]

[1] Department of Simulation and Graphics
Otto-von-Guericke-University of Magdeburg, Germany
{krueger|tietjen|hintze|preim}@isg.cs.uni-magdeburg.de

[2] ENT Department
University Hospital of Leipzig, Germany
Ilka.Hertel|strg@medizin.uni-leipzig.de



Abstract

In this paper, we present visualization techniques for neck dissection planning. These interventions are carried out to remove lymph node metastasis in the neck region. 3d visualization is intended to explore and to quantify anatomic and pathologic structures and thus support decisions concerning the surgical strategy. For this purpose we developed and combined visualization and interaction techniques such as cutaway views, silhouettes and color-coded distances. In addition, a standardized procedure for processing and visualization of the patient data is presented.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computing Methodologies]: Computer GraphicsMethodology and Techniques; I.4.m [Computing Methodologies]: Image Processing and Computer VisionMiscellaneous; J.3 [Computer Applications]: Live and Medical SciencesHealth;

Keywords: Medical visualization, neck dissection, operation planning, lymph node exploration

1. Introduction

Neck dissections are carried out for patients with malignant tumors in the head and neck region. These surgical procedures are necessary because the majority of the patients develops lymph node metastases in the neck region.

The extent of the intervention depends on the occurrence and location of enlarged (and probably) malignant lymph nodes. In particular, the infiltration of a large muscle (*M. sternocleidomastoideus*), a nerve (*N. facialis*) or blood vessel determine the surgical strategy. If for example the *A. carotis interna* is infiltrated, the patient is regarded as not resectable. The identification and the quantitative analysis of lymph nodes with respect to size and shape is crucial for the surgeon's decision. The image analysis and visualization techniques described in this paper support decisions regarding the resectability and the surgical strategy for neck dissections.

Visualization techniques aim at comprehensible renderings of the relevant information. This includes the visualization of the target structures and some context information necessary to illustrate the spatial relations. By means of our visualizations, we convey information concerning shape and size of lymph nodes, as well as critical distances or even infiltrations of lymph nodes into important structures. In addition to carefully parameterizing surface rendering, we explore silhouette rendering and cutaway views for parts of an object's surface. Although these visualizations are targeted at neck dissection planning they are applicable to other applications such as the evaluation of lung nodules. We also discuss interaction facilities to explore the data. In particular, we discuss the selection of lymph nodes based on their properties. Our case study report is based on 18 clinical CT-datasets which have been acquired and processed for the planning of neck dissections.

2. Image Analysis

In order to support neck dissection planning, it is crucial to segment the relevant anatomic and pathologic structures. The segmentation is a prerequisite for the selective visualization and the quantitative analysis of the patient data. For surgical planning, the extent of pathologic structures, distances to important anatomic structures and the potential infiltration are of special interest.

2.1. CT Data

We employed 18 CT-datasets which have been acquired for neck dissection planning. Eleven of these datasets contained a tumor in the head and neck region and were suspected of containing lymph node metastases as well. The quality of the datasets was diverse with respect to the signal-to-noise ratio, motion artifacts as well as the slice distance (0.7 to 3 mm), resulting from different CT scanning devices and acquisition parameter.

The data were exchanged based on a WWW upload including information concerning the diagnosis of the patient and specific requirements for computer-supported planning.We choose not to employ MRI data, although they are wide-spread for diagnosis in the neck region due to their inherent inhomogeneity and low resolution.

2.2. Requirements

In collaboration with our clinical partners, the target structures of the segmentation were identified as being most relevant for preoperative planning:

- Vascular structures (*V. jugularis, A. carotis*)
- Muscles (*M. sternocleidomastoideus*)
- Skeletal structures (*Mandible* and *Clavicle*)
- Salivary glands (*Gl. submandibularis, Gl. parotidea*)
- *Pharynx*
- *N. accessorius* (where visible),
- Primary tumor,
- Lymph nodes, with emphasis on enlarged and potentially malignant nodes.

In selected cases, the segmentation of additional structures is desirable, e.g. additional muscles or nerves.

2.3. Segmentation

Segmentation was carried out by means of the software platform MeVisLab (MeVis, Bremen, http://www.mevislab.de), a library which provides a variety of image preprocessing and segmentation methods. A live wire approach was employed for the segmentation of muscles (*M. sternocleidomastoideus, M. omohyoideus*) and the salivary glands. With this semi-automatic approach, the user selects seedpoints and the system calculates a path of minimal cost in between. This procedure is carried out in selected slices; the intermediate contours are interpolated [SPP00]. The interactive watershed transform [HP03] proved to be suitable to identify and delineate the *V. jugularis* and *A. carotis*. Intensity-based region growing was used for bone and *Pharynx* segmentation.

While the muscles and the glands could be identified in the majority of the datasets, most of the desired nerves could not be identified due to their size in relation to the image resolution. Among the vascular structures, only the *A. carotis* and *V. jugularis* could be segmented in most of the data.

Nerves are very difficult to detect in CT-images because they are very small. In datasets with a large slice distance (>3 mm), they could not be detected at all. In CT-data with low slice distance, the *N. accessorius* and *N. vagus* could be identified manually in a few slices. Due to the low slice distance the partial volume effect (averaging of signal intensities in a volume element) is less disturbing. As the approximate course of these nerves is essential for surgeons, we chose to segment the nerves partially and to employ this information for an approximate visualization (see Section 6).

Primary tumors were segmented manually as well. They exhibited low contrasts and could only be distinguished by exploiting considerable anatomic knowledge, in particular symmetry considerations. At present, also the lymph nodes are identified manually. Our ongoing research aims at an automatic detection of lymph nodes, using assumptions regarding their grey values, size and shape. The segmentation is described in more detail in [HCP*05].

3. Visualization of the Segmented Target Structures

We discarded volume rendering because it does not provide essential information for our purposes. By relying on surface visualizations, we provide all necessary information within rather small surface models which can be easily transmitted over the internet and explored using wide-spread software. In addition we use functionalities from modern graphics hardware (GPU) which is optimized for surface rendering.

3.1. Color Selection

Our color selection was guided by observations from textbooks [Net02] and later refined in discussions with the clinical partners. Transparency was primarily used to expose important structures, such as lymph nodes. This type of visualization is shown in Figure 1.

After processing three CT-datasets, we evaluated all visualization parameters. In several in-depth discussions, we modified colors and transparencies for all structures to enhance contrasts and recognizability of object borders. As a result, a final color table was developed which represents our standardization (see Table 1). Finally, all datasets have been adapted to these values.

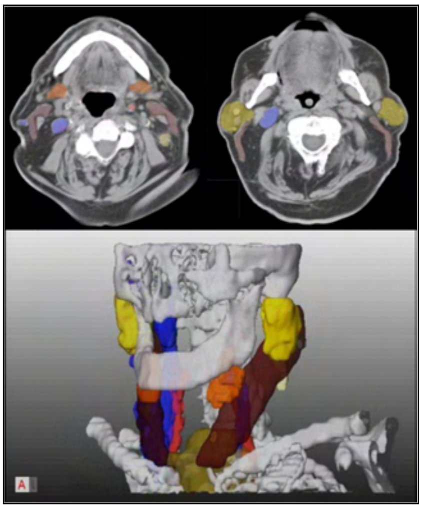


Figure 1: *2d and 3d visualizations are combined for neck dissection planning. The colors used in the 3d visualization are also used to superimpose segmentation results in 2d slice data.*

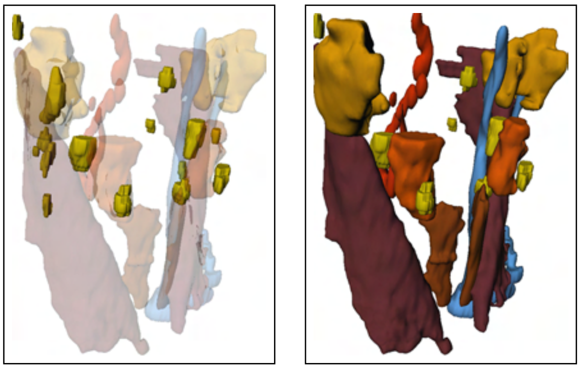
3.2. Material Effects for the Visualization

It turned out that object-based transparency specification does not allow a comprehensible visualization of complex structures. With multiple highly transparent objects, the specific location of a target object is barely visible with a high opacity on the other hand the spatial relations between emphasized objects are difficult to recognize (see Fig. 2).

The key for the solution is employing object-based opacity maps with alternating opaque and semi-transparent stripes. They are mapped to the neck muscles in roughly the same direction as real fibers. For this purpose we use the calculated envelope of each muscle and compute its bounding cylinder. By using the normals of the muscle we transfer the texture coordinates from the cylinder to each vertex of the muscle. With this technique the real texturing of neck muscles is slightly indicated in the visualization. In [DCLK03] the identification of muscle fibres for hatching is presented. This technique was found to be too complex for our purposes, since neck muscles are regarded as context information only.

Structure	Red	Green	Blue
A. carotis	240	50	50
V. jugularis	80	80	250
Muscles	100	40	20
Skeletal structures	255	255	255
Salivary glands	180	150	110
Pharynx	255	190	150
Nerves	240	185	80
Primary tumor	255	255	200
Lymph nodes	255	255	150

Table 1: *Color table for the standardized visualization of neck structures*



(a) Neck structures drawn too transparent.

(b) Neck structures opaque.

Figure 2: *With object-based transparency assignment, the location of lymph nodes cannot be depicted effectively.*

To improve the visibility of lymph nodes and tumors, all colors from other objects are reduced in saturation and lightness. Especially skeletal structures are invisible during surgery and provide only spatial orientation, e.g. *Mandible* and *Clavicle* serve as landmarks. In Figure 3, the improved color selection for lymph node emphasis is shown.

Inspired by illustrations from textbooks [Net02], we use a material with shiny impression for vascular structures. This technique enhances the recognizability of vascular structures (cf. Fig. 3 or 6).

Segmented objects from clinical datasets mostly exhibit unnatural artifacts. Therefore we smoothed all objects visually (not geometrically), by assigning a slight self-illumination (emissive color in an SoMaterial node of OpenInventor) to these objects. This effected a visual flatness of the surface because the shading does not render hard contrasts (see the bones in Fig. 3 and 5).

Further visualization techniques were investigated. The use of silhouettes for highly transparent objects considerably increases the recognizability. As a result, silhouettes are used e.g. to enhance strongly transparent objects (see Fig. 3). We do not consider hatching as a promising technique for surgery planning. It seems to be challenging to reliably derive appropriate hatching parameters from the complex geometries of segmented objects. The improved spatial understanding is probably not significant to justify additional interaction effort.

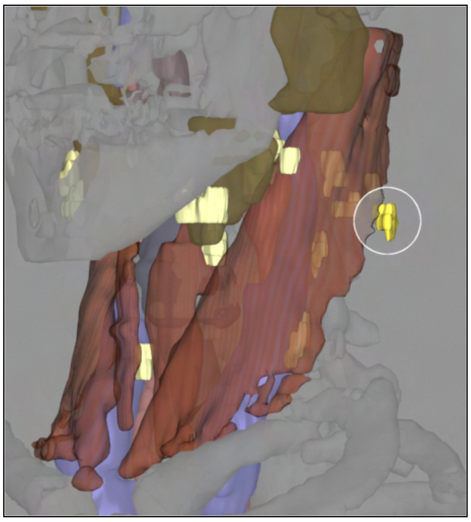


Figure 3: *Emphasized lymph node partially behind the M. sternocleidomastoideus. Note the use of a cutaway view as well as the thin silhouette line to enhance depth perception.*

In contrast, the use of cutaway views is promising. As an example, a muscle which is covering a tumor should be rendered only transparent in regions, where the tumor is behind. In Section 4.2 the use of cutaway views is presented.

3.3. Integrating Measurement in Neck Dissection Planning

Measurement tools to compute the extent of anatomic structures and the distance between structures are also provided [PTSP02]. With these tools (see Fig. 4), the extent of enlarged lymph nodes can be determined precisely. The measurements are directly included in the 3d visualization.

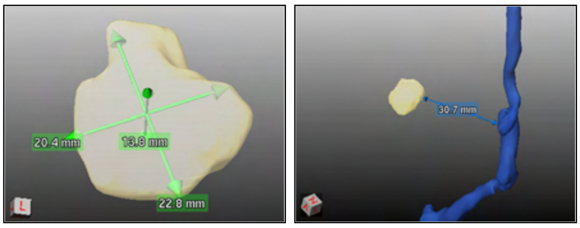
4. Interaction Techniques for Exploring Lymph Nodes

The exploration of a complex set of enlarged and therefore surgically relevant lymph nodes requires appropriate interaction techniques. The usual selection of objects via their name is not feasible. The selection of entire lymph node groups or via measurement results is more appropriate. Extent or minimal distances to risk structures are possible criteria. Two problems related to lymph nodes are essential for surgery planning: the exploration of enlarged lymph nodes (partly over 20) and the evaluation of infiltration or resectability of lymph nodes.

4.1. Sequential Visualization of Lymph Nodes

The basic interaction for the exploration of lymph nodes is the selection. We suggest a facility to step through all lymph nodes with a simple interaction. We found that the most interesting information are the quantity of enlarged lymph nodes and their potential malignity. Therefore, a simple list, ordered by the lymph node number, is not appropriate for exploration. In our planning tool we provide three different selection criteria - extent, volume or malignity from TNM-classification, discussed in Section 4.5.

As a feedback after selection, lymph nodes should become visible. It is not appropriate to render a large object, such as the *M. sternocleidomastoideus*, highly transparent, to expose a small lymph node. A better alternative is to render only a small part of the muscle transparent which can be achieved by cutaway views (see [VKG04]).



(a) Extension measurement of a tumor, automatically computed via principal axis transformation.

(b) Minimal distance between an enlarged lymph node and the *V. jugularis*.

Figure 4: *Measurement tools for automated computing the extent of structures and minimal distances.*

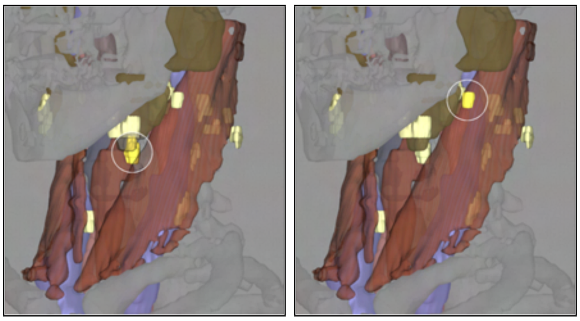
4.2. Emphasis with Cutaway Views

For the exploration of lymph nodes, we generate a cutaway view with a cylindrical cutting volumen. We calculate the convex hull from the lymph node in 3d and project it via OpenGL to the screen. From the result wie calculate the convex hull in 2d and finally the minimal enclosing circle plus a fixed margin around a lymph node. So with each step we reduce the involved points to enhance the speed. The resulting volume is cutting all structures in front of the lymph node. The cylinder is aligned orthogonal to the viewing plane and is terminated at the lymph node. OpenInventor, OpenGL 2.0 and the fragment shader functionality from modern GPUs is used to realize the cutting of the structures in realtime.

The intersecting parts of foreground objects within the bounding volume are displayed strongly transparent. However, the depth perception is limited in this region. Therefore a thin silhouette (see [IFH*03]) of the muscle is included brightly, calculated simply from the scalar product between viewing vector and the surface normals.

With these visualization parameters, foreground and background objects are correctly perceived. The emphasized area is additionally marked with a bright circle. The radius of this area is the same as the radius of the cutting cylinder. In technical illustrations in contrast, organic shapes or zigzag should be used to compute cutaway views (see [DWE03]). The lymph node is also visually enhanced by raising the saturation of color (see Figure 3 for the interactive visualization result).

As shown in Figure 5, this combination of visualization techniques is applicable also for multiple occlusions 5(a) or full visibility 5(b). Hence, the surgeon may interactively step through all lymph nodes, e.g. by pressing the tab key. The currently selected object (CSO) will always be clearly emphasized. We chose not to rotate the camera automatically to emphasize the CSO because of distracting effects.



(a) Lymph node behind the *M. sternocleidomastoideus* and the *Gl. submandibularis.*

(b) Lymph node in front of all other structures.

Figure 5: *Combining color, transparency mapping and a cutaway view to emphasize a single lymph node. In the interactive tool the stepping through all lymph nodes is facilitated.*

4.3. Visualizing Distances to Risk-Structures

For the evaluation of the resectability, distances to risk structures are crucial. We employ a distance-dependent coloring of the neck vessels and muscles which conveys the distance to the lymph nodes. With a discrete color scale (gradation: 2 and 5 mm) the resectability of this target may be evaluated.

The emphasized lymph node in Figure 6 is located in front of the *M. sternocleidomastoideus*, where the distance information is displayed. Two depending levels are encoded for the distance. For a well-defined separation of the structures in the 3d scene, the used colors were evaluated by the clinical partners and regarded as appropriate. This kind of accentuation leads to a more simple application for several objects, so the visual focus is located at the CSO. Distance information relating to multiple objects would more likely confuse the viewer.

A color-coding of the distances does not indicate an infiltration of a muscle or a vessel. We employ a line character accentuation (see [TIP05]) of the cut line which is marked above the illustration. In Figure 7, the potential infiltration of the *M. sternocleidomastoideus* is shown. According to the segmentation results, these lymph nodes reach into the muscle. In reality it is possible, that the muscle tissue is displaced, but not infiltrated. A displacement occurs considerably more often. Distance-related visualizations may be generated for each anatomic structure. Other relevant examples are the *A. carotis* and the *V. jugularis*.

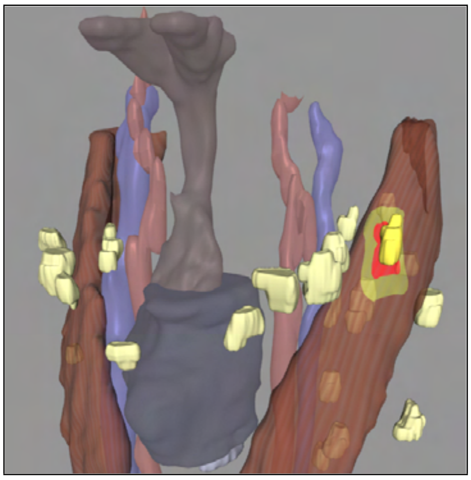


Figure 6: *Color-coded distance of a lymph node to the M. sternocleidomastoideus. The 2 mm distance is coded in red, 5 mm in yellow.*

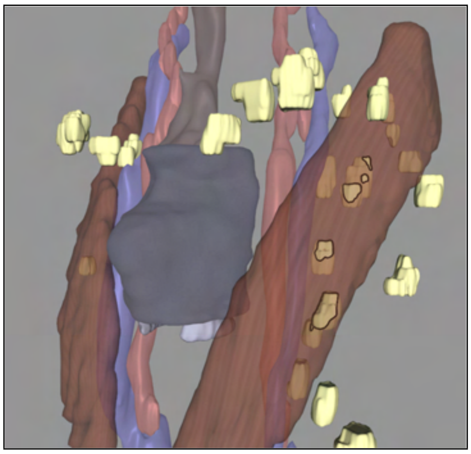


Figure 7: *Possible infiltration of the M. sternocleidomastoideus. Silhouette lines form an intersection line between muscle and lymph nodes.*

A drawback of this visualization technique is, that lymph nodes which are in front and potentially infiltrate the muscle, will not be accentuated in the current viewing position. Therefore the user should rotate the 3d scene. A color-coded view for potentially infiltrating lymph nodes is also provided for a fast overview.

4.4. Visualization of Lymph Node Size

Primarily, the size of lymph nodes is important. Medical doctors consider lymph nodes with an extent of more than 1 cm as critical (potentially malignant) and would resect them. Therefore, we generate an initial visualization which conveys the size of lymph nodes. In Figure 8, this is realized by a front view and color-coded lymph nodes. The color graduation appears via two discrete values: yellow for lymph nodes smaller than 1 cm and turquoise for extents beyond. Here, we emphasize lymph nodes larger than 1 cm minus the slice thickness, to account for possible inaccuracies from image acquisition and segmentation. The maximum extent is used to realize the color-coding. If this value exceeds the calculated threshold, the lymph node will be classified as bigger than 1 cm and is coded as enlarged. A discrete color-coding with more than two grades turns out to be inappropriate, because the surgical decision is of binary nature, too. In the real surgical procedure, all lymph nodes will be removed, that are palpably enlarged.

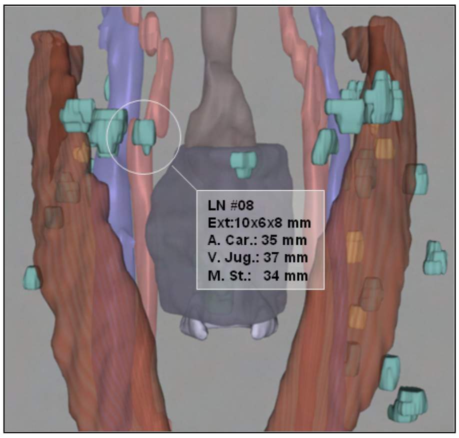


Figure 8: *Color-coded lymph node size. Two levels are used: yellow below 1cm and turquoise above. The exact values are readable via tooltip.*

This visualization is based on data known from the segmentation process. All lymph nodes and tumors are measured automatically by a principal component analysis from which the extent is derived [PTSP02]. We do not calculate the volumes of lymph nodes since high measurement inaccuracies due to very small volumes are to be expected (volumetry is uncertain when a large portion of border voxels occurs).

By a mouse-over interaction, tooltips, respectively text boxes, fade in with the precise measurement values of extent and minimal distances to risk structures (see Fig. 8). For primary tumors the same interaction is provided.

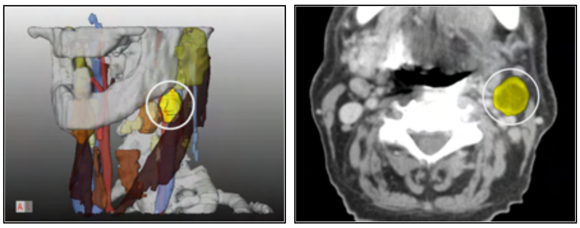
4.5. Malignity and TNM Classification

Surgeons grade the level of a tumor disease according to a fixed scheme, the TNM classification. It is constructed based on three numerical values, with possible levels for each. T stands for the tumor grade (five levels), N for the lymph node state (N0 or N1) and M for the level of distant metastases (M0 or M1). About the last one we cannot state anything with computer assistance, but the T and N values can be determined algorithmically.

The lower levels of the T-classification (1, 2 and 3) characterize mostly the size of pathological structures. Level 4 and 5 are assigned with respect to the infiltration of critical structures, such as vasculature. An adhesion with a neck arteria tends to a level 5 classification, because such cases are not resectable.

For tumor classification, the computer cannot perform an automatic estimation. The shapes vary strongly and the spatial relationships are too complex. A computer assistance for the surgeon is reasonable: According to the measurement results, the extent can be employed for a suggestion of the T-level is presented. By observing the 3d-visualization and the overlayed segmentation in the CT-slices (see Fig. 1), the surgeon can correct the initial level.

The N part of the classification can be automated. This is reasonable, because of the large number of lymph nodes in most cases. The extent of a lymph node is considered in determining conclusion about malignancy. Spherical structures are more likely malignant than longish ones and their roundness can be calculated by comparing the 3 principal directions.



(a) 3d-visualization. (b) Axial CT-slice.

Figure 9: *Representation of a lymph node with a central necrosis marked in 3d and in a 2d view.*

Another clue concerning malignancy is the inner gray value characteristic. A central necrosis has much darker values in the center than on the border. In Figure 9, such a case is shown. Detectable characteristics in gray values can be used for an automated N classification. If required, the suggested TNM is displayed on the screen border in our operation planning tool, like other patient data.

5. The Segmentation and Visualization Process

It was a major goal to produce comparable visualizations for different patient data. Besides the visualization, a procedure for the treatment of datasets is to be defined. This includes the type and resolution of the datasets, the application of segmentation methods, the type of visualization and the presentation of the results. We consider the following aspects of standardization:

- type and resolution of the processed datasets (CT, < 3 mm slice distance)
- standardized "order sheet" with specifications regarding structures, that should be segmented besides the standard in this case
- technique of segmentation for the different structures
- naming of the structures
- colors, views and types of visualization
- measurement of segmented lymph nodes, tumors and the distances to risk structures
- data exchange and result presentation on the project's web page

The segmentation and visualization is carried out as a service for the surgical partner in the framework of a research project. Patient datasets are always submitted with an "order sheet". By this, the diagnosis is stated and target structures and measurements besides the standard are listed (see Section 2).

The segmentation process is also standardized (sequence of segmentation tasks). However, parameters have to be adapted to each dataset. The result is presented as images and small animation sequences. 2d-slice views with the segmentation results (which are radiologically evaluated) transparently overlaid to the original data were created to support the verification of segmentation results and the mental integration of 3d visualizations with the underlying slice data. Selective clipping of bony structures was used to enhance the interpretation of the spatial relations. The clips, images and interactive 3d-data are available for the project partner via a secured web page.

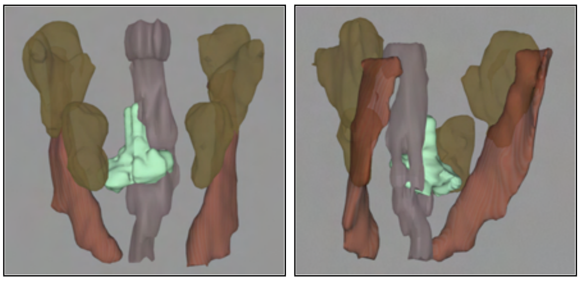
The average time for image analysis and creation of the visualizations was approximately 90 minutes. Most of the time was spent on the segmentation of lymph nodes.

6. Influence on Surgical Strategies

The visualization results were compared with the experiences of real surgical interventions. The surgeons attested a high degree of correspondence to intraoperative views. In some cases, the results of computer assisted planning were essential for the surgical decision.

For the surgeon it is necessary to evaluate distances to risk structures (see Section 4.4). The number of lymph nodes is employed to develop an understanding of the expected difficulty of the resection. These nodes are not visible in the neck area, but hidden in e.g. fatty tissue.

The above-mentioned information improves neck dissections with respect to speed and safety. In contrast, other information a priori can lead to choose another surgical strategy. If it turns out, that possibly important structures are infiltrated, the involved areas are not resectable, without previous radiation therapy. Therefore, it is important to estimate the resectability as reliably as possible and to choose the right surgical strategy preoperatively. In neck dissections, there are the following strategies: left or right sided and with different kinds of radicality e.g. with resection of muscles.



(a) Front view.

(b) Back view.

Figure 10: *A large tumor (green) is infiltrating the Pharynx. The upper tail also infiltrates the cranial base.*

Figure 10 presents a case, where the tumor had a long tail, that was not noticeable on CT-slices. Due to the infiltration of the cranial base, the intervention had to be terminated unsuccessfully. The surgeon stated, that having seen the segmentation results in advance might have led to another surgical strategy. Currently, CT-data is acquired close before surgery. There is almost no time for preparing the operation planning.

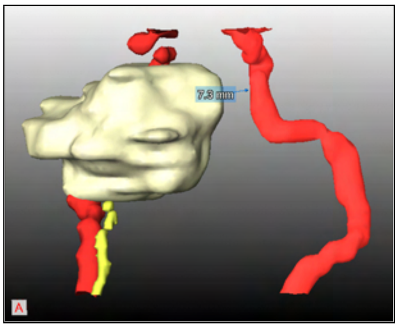


Figure 11: *N. hypoglossus (yellow) and A. jugularis (red). Notice the slice artifacts in the course of the nerve near the tumor (dark yellow).*

Information about the nerves are valuable, because they are often injured and patients carry away heavy intricacies for lifetime. Slice distance concerned in the project is 2-3 mm in average. A higher resolution is required to segment big facial nerves. Even then, only a partial segmentation is feasible, because their size and contrast is too small. This is the case even at the high resolution of the Visible Human dataset. Similar to [HPP*00], we model the course between selected points of the nerve. Instead of visually appealing B-Splines, we use simple lines with an appropriate thickness. Figure 11 shows such a segmentation result from a 1 mm slice distance data set. The visualization of the approximate course of the nerve was regarded as useful.

7. Conclusion & Future Work

We presented image analysis and visualization techniques for planning neck dissections. The focus of our work is the visualization of enlarged lymph nodes and the surrounding structures. The image analysis which requires considerable experience, is carried out as a service in the framework of a research project. As a result, surgeons are provided with standardized static visualizations and with standardized animation sequences, primarily rotations of different subsets of the relevant target structures. To explore the data themselves, they are provided with an interactive system with surface rendering and measurement facilities.

We attempted a standardized report consisting of images from standardized viewing directions. The correlation between 3d visualizations and the original 2d slices of the radiological data is crucial to assess whether the 3d visualizations are reliable. Therefore segmentation results are indicated as semitransparent overlays to the original CT-data. The potential of 3d visualizations for surgery planning cannot be fully exploited by means of standardization. Each and every case exhibits some peculiarities which require interaction techniques to explore them. In particular, the occurrence, number and size of enlarged lymph nodes differ from patient to patient. Therefore, we developed an "order sheet".

Our work is directed at a progress in planning neck dissections; more reliable preoperative decisions and more safety during the intervention are the primary goals. Our strategies to adjust 3d visualizations, to explore spatial relations, is applicable to other areas of computer assisted surgery. The use of silhouettes as well as the use of cutaway views to expose hidden pathologic structures turned out to be useful for surgery planning. Cutaway views are also useful for the exploration of round lesions (lung nodule) or small liver metastases.

Future work includes an in-depth user study to characterize the impact of 3d visualization on the surgical strategy. In this study, we will compare surgery planning based on conventional information (axial slices of CT-data) with surgery planning based on the additional information which is available after image processing. A specialized further development of the interactive planning tool, the "InterventionPlanner ENT" (ear nose throat), currently will be finished.

Acknowledgments

This work was carried out in the framework of a project supported by the Deutsche Forschungsgemeinschaft (DFG) (Priority Programme 1124, PR 660/3-1). Special thanks go to Jeanette Cordes and Ragnar Bade for good ideas and substantial support on visualization issues. Dr. Uta Preim validated image analysis results as radiologist.

References

[DCLK03] Dong F., Clapworthy G. J., Lin H., Krokos M. A.: Non-photo-realistic rendering of medical volume data. *IEEE Computer Graphics and Applications 23* (2003), 44–52.

[DWE03] Diepstraten J., Weiskopf D., Ertl T.: Interactive Cutaway Illustrations. *Computer Graphics Forum 22*, 3 (2003), 523–532.

[HCP*05] Hintze J., Cordes J., Preim B., Strauss G., Hertel I., Preim U.: Bildanalyse für die präoperative Planung von Neck dissections. In *Bildverarbeitung für die Medizin 2005* (2005), Informatik aktuell, Springer.

[HP03] Hahn H., Peitgen H.: IWT-Interactive Watershed Transform: A hierarchical method for efficient interactive and automated segmentation of multidimensional grayscale images. In *Medical Imaging 2003: Image Processing* (2003), vol. 5032, SPIE, pp. 643–653.

[HPP*00] Höhne K. H., Pflesser B., Pommert A., Riemer M., Schubert R., Schiemann T., Tiede U., Schumacher U.: A realistic model of the inner organs from the visible human data. In *MICCAI* (2000), pp. 776–785.

[IFH*03] Isenberg T., Freudenberg B., Halper N., Schlechtweg S., Strothotte T.: A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications 23*, 4 (2003), 28–37.

[Net02] Netter F. H.: *Atlas of Human Anatomy*, 3 ed. ICON Learning Systems, 2002.

[PTSP02] Preim B., Tietjen C., Spindler W., Peitgen H.-O.: Integration of Measurement Tools in Medical Visualizations. In *Proc. of IEEE Visualization* (2002), pp. 21–28.

[SPP00] Schenk A., Prause G., Peitgen H.: Efficient Semiautomatic Segmentation of 3D Objects in Medical Images. In *Medical Image Computing and Computer-assisted Intervention* (2000), vol. 1935 of *LNCS*, Springer, pp. 186–195.

[TIP05] Tietjen C., Isenberg T., Preim B.: Combining Silhouettes, Surface, and Volume Rendering for Surgery Education and Planning. In *IEEE/Eurographics Symposium on Visualization* (2005), Springer.

[VKG04] Viola I., Kanitsar A., Gröller M. E.: Importance-driven volume rendering. In *Proc. of IEEE Visualization* (2004), pp. 139–145.