Illustrative Focus+Context Approaches in Interactive Volume Visualization

Stefan Bruckner[1], M. Eduard Gröller[1], Klaus Mueller[2], Bernhard Preim[3], and Deborah Silver[4]

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

2 Center for Visual Computing, Computer Science, Stony Brook University
mueller@cs.sunysb.edu

3 Department of Simulation and Graphics, Otto-von-Guericke University of Magdeburg
preim@isg.cs.uni-magdeburg.de

4 Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey
silver@ece.rutgers.edu

Abstract

Illustrative techniques are a new and exciting direction in visualization research. Traditional techniques which have been used by scientific illustrators for centuries are re-examined under the light of modern computer technology. In this paper, we discuss the use of the focus+context concept for the illustrative visualization of volumetric data. We give an overview of the state-of-the-art and discuss recent approaches which employ this concept in novel ways.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling, J.3 Life and Medical Sciences

Keywords and phrases Illustrative Visualization, Volumetric Data

Digital Object Identifier 10.4230/DFU.SciViz.2010.136

1 Introduction

A considerable amount of research has been devoted to developing, improving and examining visualization techniques for scientific volume 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.

A recent trend in volume visualization is that researchers tend to use traditional illustrations as an inspiration for their work. As the domain of scientific illustration is based on centuries of experience in the depiction of complex volumetric structures, it represents a valuable source for visualization researchers. A common technique found in many traditional illustrations (see Figure 1) is referred to as focus+context in visualization literature. As there is often not enough space available to display all information in sufficient detail, the general idea is to emphasize regions of particular interest (focus) without completely removing other information important for orientation (context). Moreover, focus+context visualizations are not only motivated by space limitations but also by human visual perception. People are capable of simultaneously perceiving both local detail and global context [46]. Focus+context methods make it possible to show more detailed or targeted information and at the same

Scientific Visualization: Advanced Concepts.
Editor: Hans Hagen; pp. 136–162
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Center for Informatics (Germany)

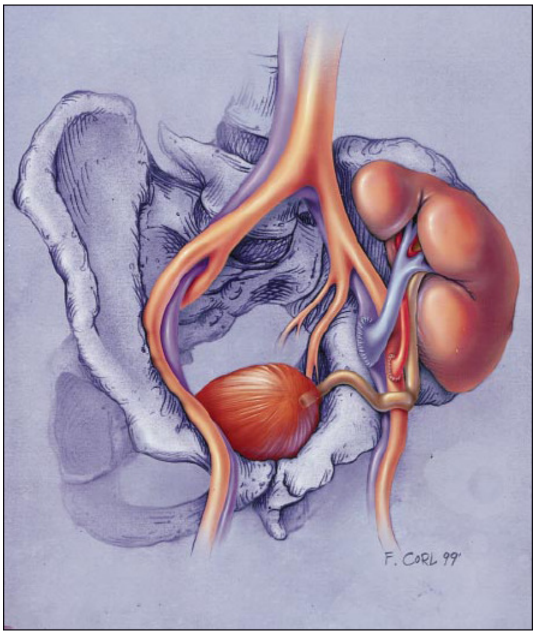


Figure 1 Medical illustration using focus+context – the image shows the position of a kidney after transplantation into the pelvic region. Bones are displayed in a stylized way while focus is emphasized by a more realistic rendering style (image courtesy of Frank M. Corl [17]).

time give users a sense of where in the data the zoomed-in, more detailed, or pointed out information is.

In connection with the advanced interaction possible in computer-based visualization the focus+context concept offers additional advantages. As the focus can be modified interactively, it serves as a means to explore complex data. This paper reviews recent approaches which employ different kinds of focus+context techniques for improved illustrative visualization of volumetric data. Section 2 gives an overview of previous work in the area. In Section 3 we discuss a framework for distortion-based focus+context volume visualization. It employs traditional (i.e. physically plausible) as well as arbitrary distortions for highlighting structures in volumetric data. Section 4 presents deformation techniques which allow a user to manipulate the data for improved comprehension in Section 4. Next, in Section 5, we focus on concepts used in the design of a volume-based illustration system which aims to produce visualizations with the aesthetic appeal of traditional illustrations. In Section 6 we discuss how illustrative focus+context visualization can be employed in medical applications for surgery planning. Finally, the paper is concluded in Section 7.

While distinct approaches, the individual parts of this paper have many similarities. They all employ the concept of focus+context to prevent information overload and allow the user to concentrate on certain structures of interest. Traditional scientific illustration is used as a source of inspiration and adapted to the additional degrees of freedom offered by computer-based visualization. Finally, all approaches use the capabilities of graphics hardware to allow interactive navigation and interaction.

2 Related Work

Focus+context approaches have been heavily employed in information visualization. Many approaches are based on spatial distortion, for instance fish-eye views [21], hyperbolic trees [31] or the document lens [44], and others [29] (see [34] for a more detailed overview). Viewpoint-dependent distortion of 3D data [8, 9] highlights regions of interest by dedicating more space to them. Other methods, for example tool glass and magic lenses [3], allow the display of additional data dimensions on demand. Cue methods [30] enhance the visualization by assigning visual cues to certain objects so that they are more prominent to the viewer without hiding the context.

While the area of non-photorealistic rendering [25, 49] is more concerned with imitating artistic styles in an automated way, illustrative visualization goes one step further and tries to apply these techniques selectively to enhance visual comprehension. Illustrative visualization can be seen as a fusion of the focus+context concept and non-photorealistic rendering. The visual abstraction is realized at two basic levels: stylized depiction (low-level abstraction) deals with how objects should be presented, while smart visibility (high-level abstraction) is concerned with what should be visible and recognizable.

The inherent complexity of volumetric data has lead to a considerable amount of research in illustrative techniques for volume visualization. Most approaches tend to combine both levels of abstractions and do not have an explicit steering mechanism. Levoy [35] 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 [43] 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. [37] developed an interactive volume illustration system that simulates traditional stipple drawing. Csébfalvi et al. [20] 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 [38] present a hardware-accelerated approach for high-quality non-photorealistic rendering of volume data. They also suggest the use of lighting transfer functions [39] for object enhancement. The concept of two-level volume rendering, proposed by Hauser et al. [26], 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. Zhou et al. [60] use a focal-region-based rendering approach which depicts context data using a different rendering technique. They also propose the use of distance to emphasize and de-emphasize different regions [59]. The distance from a focal point is used to directly modulate the opacity at each sample position. Tapenbeck et al. [51] employ distance-based transfer function based on the distance to an object (rather than a focal point). Levoy and Whitaker [36] perform adaptive resolution volume rendering based on gaze direction. Cignoni et al. [13] provide the MagicSphere metaphor to visualize 3D data with the MultiRes filter. Wei et al. [58] apply fisheye views to particle track volume data using nonlinear magnification functions. LaMar et al. [33] integrate a 3D magnification lens with a hardware-texture based volume renderer. Cohen and Brodlie [15] magnify features by generating a new volume using inverse distortion functions.

Some recent approaches explicitly distinguish between low- and high-level abstraction. Viola et al. [54, 55] map an importance function which specifies the relevance of different structures within the volume data to appropriate levels-of-spareness which control object appearance. Different importance compositing strategies control object visibility. Svakhine et al. [50] employ illustration motifs to control the appearance of objects at varying degrees of complexity.

3 Distortion-based Focus Enhancement Using Volumetric Lenses

Distortion-based focus enhancement with lenses has a long history, with the first reported uses going all the way back to the Greeks, Arabs, and Romans. Greek philosopher Aristophanes (448 BC-380 BC) already knew that glass could be used as a magnifying lens, but it was not until roughly 150 AD that Ptolemy discovered the basic rules of light diffraction and wrote extensively on this subject. Back then, magnifying glasses were mainly used as a reading aid by the literate class. For example, Roman tragedian Seneca (4 BC-AD 65) is said to have read "all the books in Rome" by looking through a glass globe of water. A thousand years later, far-sighted monks employed segments of glass spheres which they laid against reading material to magnify the letters, and the development of these "reading stones" was based on the theories of the Arabic mathematician Alhazen (roughly 1000 AD). This basic invention was then later refined by Venetian glass blowers, who constructed lenses that could be held in a frame in front of the eye instead of directly on the reading material. While these first spectacles were intended for use by one eye only, the idea to frame two ground glasses with wood or horn into a single binocular unit was introduced in the 13th century. Eventually, in 1268, Roger Bacon made the first known scientific commentary on lenses for vision correction, and he is generally credited with the invention of the magnifying glass, and perhaps with the foundation of the field of optics as a whole. Since then these basic lens optics have experienced a great revolution, on a vast order of scales, ranging from the magnification of individual biological cells to the scanning of the far-out cosmos. In many applications, and definitely in the case of spectacles, preserving context is a strong necessity, in order to maintain and provide ease of navigation. Preserving context usually means that the resolution of the visual information presented is highest in the foveal center (the focus), and then falls off towards the periphery in some smooth fashion, without performing any clipping within the viewing area. This is usually the case in lens-based physical optics. In more recent years, the laws of lens optics have also found application in the virtual world, using computers to implement these general concepts. In the beginning, the emphasis was on realistic simulations of the physical laws, and a great number of computer graphics papers were written to that effect [47]. However, while computer graphics was mainly concerned with the realistic simulation of optical effects in possibly very complex scenarios, the field of information and data visualization has been more in line with the original intentions of the ancient "reading stones" concept, that is, the magnification of objects of interest for better perception of their detail. It was quickly discovered that virtual (computerized) lenses are no longer constrained by the physical laws of optics, allowing the liberal use of these concepts in creative ways.

The approach summarized in this section of the paper (see [57] for more detail) seeks to generalize distortion functions and to make them interactive via implementation on graphics hardware. The latter allows their use within an engaging volume exploration tool, where a real-time response to user actions is a must – a property that is also expected, and in fact taken for granted, in physical lenses. As was mentioned above, in addition to physics-based lens optics, software lenses also allow the derivation and implementation of functionalities that do not have counterparts in physical optics, or at least are hard to fabricate. We provide volumetric lenses in both categories. In the former, we devise a set of lenses that tune their

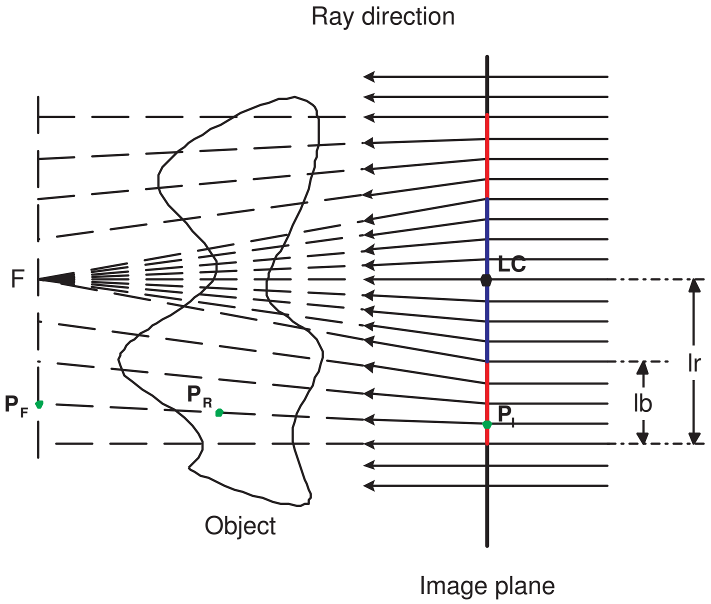


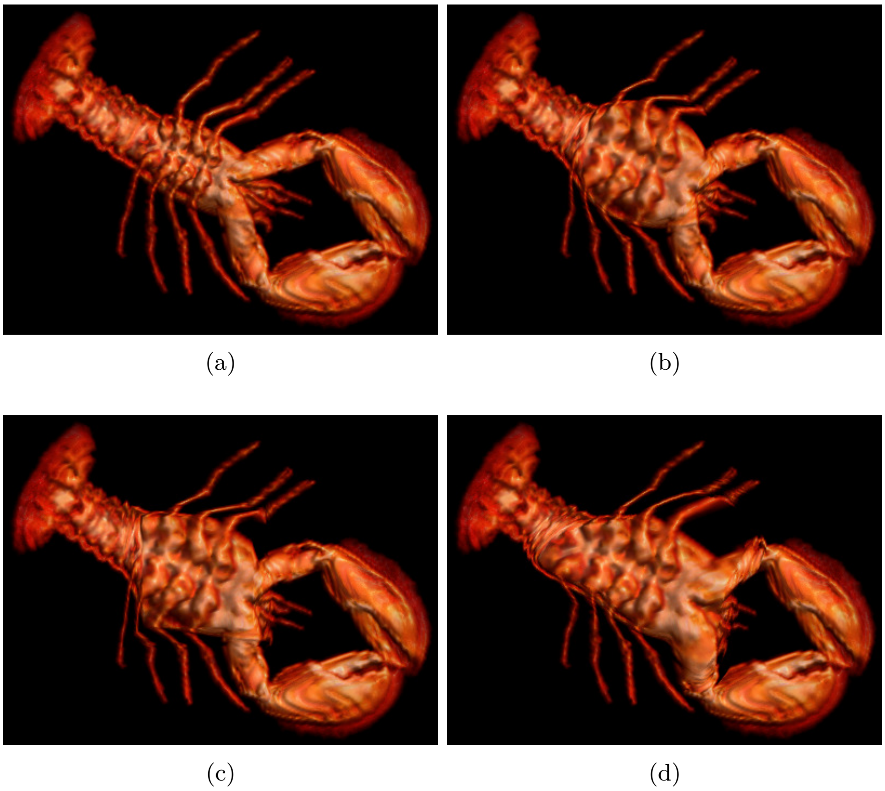
Figure 2 Illustration of the general lens mechanism.

geometry to the underlying feature semantics, while in the latter we allow the user to design the distortion function with a free-hand tool. In the following sections, we will describe the key components of our framework, and we conclude with pointers to possible extensions of our framework.

3.1 Virtual Lens Optics in a Volumetric Environment

Our lens aims to provide smooth transitions between focal and peripheral regions, with no clipping. Further, it aims to keep the lens effects local. Thus, the additional screen area dedicated to the focal (magnified) regions must be taken away from the peripheral (minified) regions. In sampling theory terms, this means that the focal volume regions are oversampled, while the peripheral volume regions are undersampled. The latter requires proper anti-aliasing during image generation. Figure 2 illustrates these concepts, using a raycasting rendering paradigm. Here, the blue line segment on the image plane represents the magnification part of the lens, LC is its center point and F is the virtual focal point. When orthogonal incident rays hit the image plane, in the area of the focal region, the ray directions are modified and go through F. Therefore, a ray cone is formed between the lens and F, and the object regions within this cone are rendered in an enlarged area on the image plane. The peripheral regions (to the left and right of the focal region) are represented by the red line segments on the image plane with width lb and are rendered at reduced resolution, while image regions outside $+-lr$ (the radius of the lens) appear at normal magnification. Thus, the paths of the rays traversing the peripheral region form the smooth transition between normal and focal region.

This general framework enables the design of lenses with arbitrary shapes. Figure 3a shows the original volume rendering obtained with no lens, and Figure 3b-d are renderings obtained using a circular, square, and arbitrary-shaped lens, respectively.


(a) (b)

(c) (d)

Figure 3 Magnifier volume renderings with (a) No lens, (b) Circular lens, (c) Square lens, (d) Arbitrary-shaped lens.

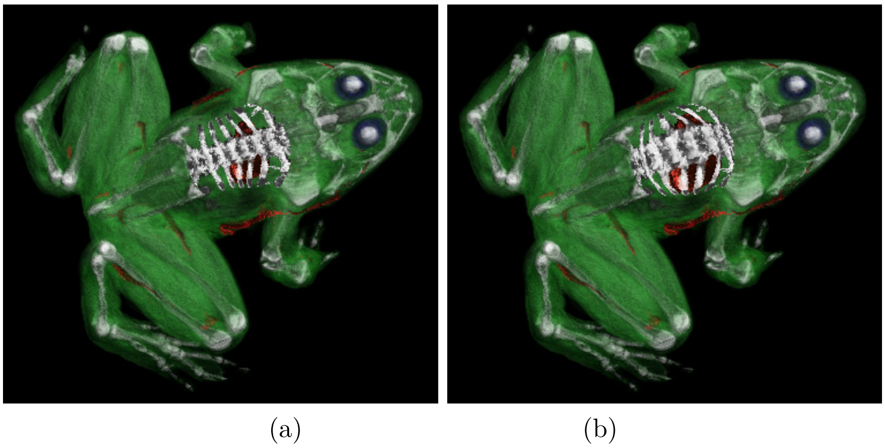

(a) (b)

Figure 4 Magnifier volume renderings for the bone feature in a segmented frog dataset. (a) and (b) are renderings without and with magnification under a circular lens.

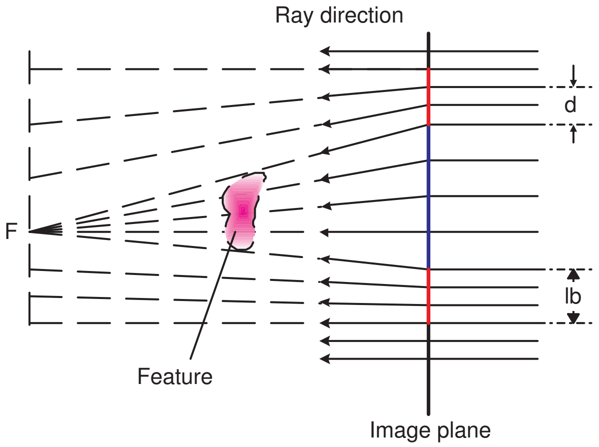


Figure 5 Feature-based lens illustration.

A segmentation of the dataset enables magnification with semantics. Figure 4 was generated by allowing the rays under the lens to penetrate the frog's skin, magnifying the skeleton underneath.

3.2 Feature-centric Lens

The feature-centric lens provides users a means to highlight portions of interest in volume objects by dedicating more screen area to them. This also promotes a more accurate and differentiated understanding of these features, since fine detail is enlarged. In this lens, the shape of its magnification portion (and that of the surrounding transition region) is defined dynamically by the shape of the features (represented by available segmentation information) in the dataset (see Figure 5 for an illustration). Here, whether an incident ray changes direction depends on the distribution of the feature. Thus the direction of each ray needs to be determined dynamically. Transition regions are also used here to retain the spatial context of the features. For each ray orthogonally incident upon the image plane, the new direction is computed as follows. Assuming all rays changed directions to the focal point F,

- if a ray passes through the feature, then its new direction is pointing to F.
- if the ray does not pass through the feature but is inside the transition region on the image plane, the distance d (see Figure 5) from its entry point to the boundary of the feature-projected area is calculated. This distance is used to compute the new direction.
- otherwise, the ray continues along its original direction.

The transition region is determined by a boundary of certain width around the feature. For this we first project the feature onto the image plane and then fill all interior points with a constant value. This region will be magnified using the over-sampling scheme defined before. The transition region field can then be marked (on the image plane) using a distance transform and the values be used to determine the direction of the rays. Alternatively, one may also determine the ray direction vector by finding the distance of the starting position to the closest neighbor in the projected feature region. We have used a GPU-accelerated search circle approach for this, with the transition region width lb being the circle's maximal radius. Figure 6 shows some rendering results for a color volume dataset, in which a user-selected

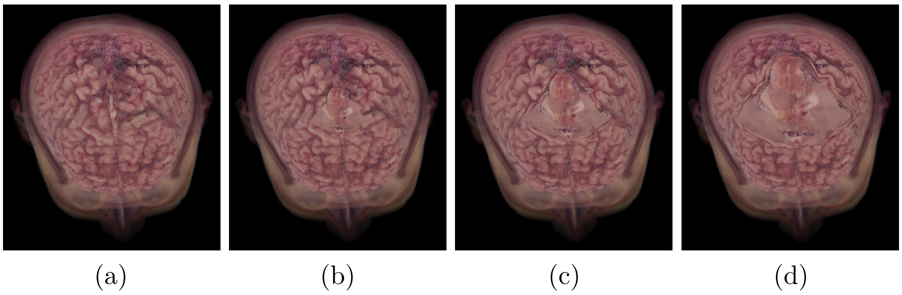


Figure 6 Feature-based lens volume renderings for a segmented human brain color volume dataset. (a) without specifying any feature of interest, (b) with a feature of interest, which is not magnified and appears too small to be seen clearly. From (c) to (d) the magnification factor increases.

feature is magnified and the other objects near that feature are compressed. Figure 6a shows the skin of the brain. Figure 6b shows an interior structure of the brain, without rendering other features which occlude this structure, while the magnified structures are shown in Figure 6c and d.

3.3 Free-form Lens Optics

Our framework also allows the design of arbitrary lens functions, using a free-form drawing tool. Figure 7 (top) shows an example. Here, the vertical axis (the height of the curve) indicates the (instantaneous) sampling rate, which is the reciprocal of the local sample distance on the image plane. The horizontal axis indicates the distance from the lens center. The higher the sampling rate, the greater is the number of rays per unit area and the magnification. Since magnification in one screen area (sampling rate > 0) must be balanced with minification in another (sampling rate < 0), the total curve integrals above and below the x-axis must approximately match. This is also illustrated in Figure 7 (bottom), where we observe that the rays shot into the object are denser in the center region of the lens and become coarser towards the boundary. Various sampling functions can be adopted to define various volumetric lenses and to achieve different volume rendering results.

Figure 8 shows some results object with the free-form lens, comparing it with the results obtained with no lens and a standard magnification lens, respectively. The toes of the foot are shown rendered with different magnification effects. The difference between Figure 8b and 8c is mainly caused by the different magnification factor distributions on the lenses. For the standard lens, the magnification factors for points projecting into the magnification region and having the same distance to the image plane are the same. Therefore, objects at the same depth are magnified uniformly. However, for the lens with cubic sampling function, the factor is the highest on the lens center and decreases gradually towards the lens boundary. Objects with projections closer to the lens center are magnified with higher magnification factors. Along any ray, the factor remains the same for different depth values.

3.4 Extensions

There are several extensions that fit well into the presented framework. For example, it would be relatively straightforward to extend the current mipmap-based zooming capabilities to more sophisticated multi-resolution data, where the data appearing under magnification comes from a different data source. This could either be a modality acquiring data at a

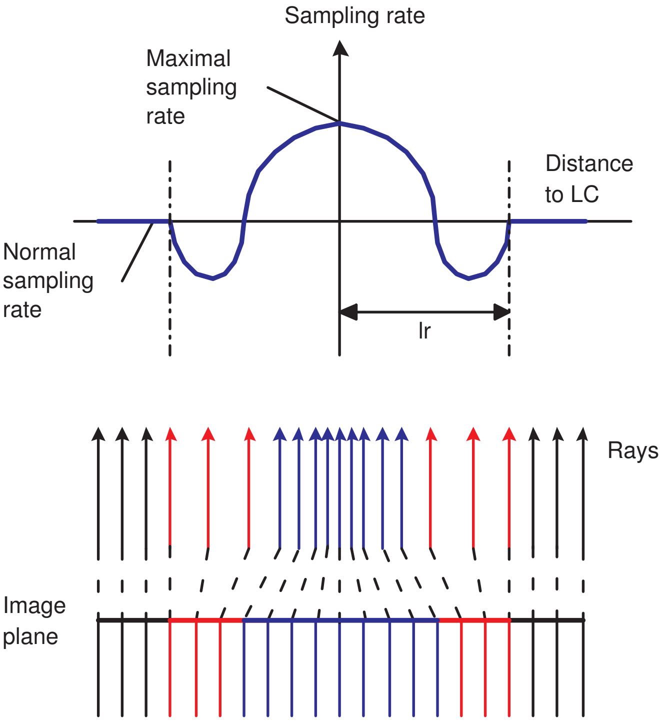


Figure 7 Sampling-rate-based lens illustration.

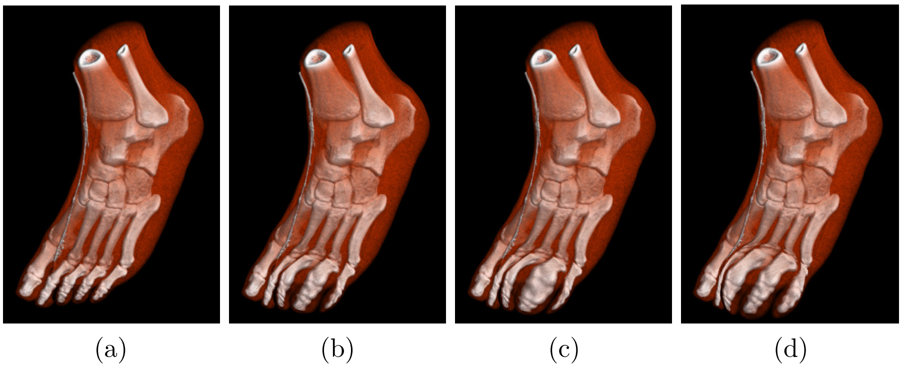


Figure 8 Comparing volume renderings with (a) no lens, (b) normal magnification lenses (c) cubic lens optics sampling function (maximal sampling rate/normal sampling rate = 3), and (d) an arbitrary lens optics sampling function.

resolution appropriate for the current local magnification rate, or a texture synthesis process that generates these data from high-resolution data swatches on the fly [56]. Finally, the lens may be generalized to provide Superman-vision capabilities – a magic lens to see the underlying uncertainties associated with the data, another channel in a multi-modal dataset, a segmentation result, or some semantic annotations that go with the data. It may fuse these views together, controlled by the user.

4 Interactive Manipulation of Volumetric Objects

The purpose of visualization is to "gain insight by using our visual machinery" [1] and "Underlying the concept of visualization is the idea that an observer can build a mental model, the visual attributes of which represent data attributes in a definable manner" [45]. The tools we currently have in 3D data visualization to help building the mental model include real-time rendering, rotation, slicing, transfer functions, segmentations and many novel focus+context renderings. However, the effective exploration of volumetric data is still a challenging task, especially for complex volumetric datasets with convoluted structures. With the prevalence of 3D imaging in all fields, such as for physical therapy, psychology, security and screening, archeology, e-commerce, etc. it is important to explore methodologies which can enhance our comprehension of the underlying structure.

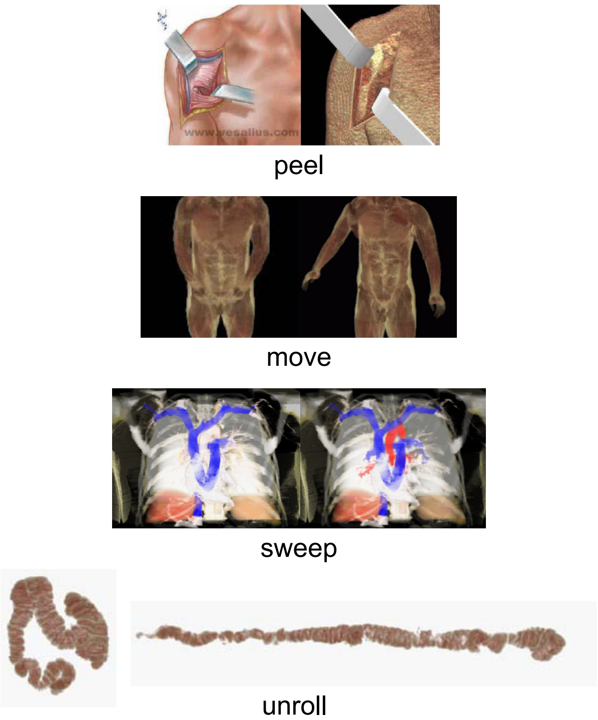
Interestingly, the use of physical representations in rapid prototyping (layered manufacturing) has capitalized on this fact. Claims include: 2D screen displays do not always provide an intuitive representation of 3D geometry; unusual or deformed geometry may be hard to comprehend on-screen; the integration of different modalities is hard to visualize; and the planning of complex 3D manipulation from 2D images can be difficult [10]. As opposed to virtual prototypes (display), "physical prototypes bring in a completely new interactive modality – the sense of handling an object" [2]. It can be much easier to learn about complex 3D shapes by holding the actual objects.

In this section, we describe a more *active* approach to visualization which allows the user to manipulate the data. The purpose is to allow the viewer to explore the data for comprehension not necessarily to simulate reality. Techniques or operations on the data that exemplify this approach include bend, move/re-pose, peel, pull, sweep, roll/unroll, cut, retract, and split. All of these descriptions are verbs, symbolizing an action on the dataset. Some examples of this type of visualization are shown in Figure 9. This type of visualization is common in surgical education and simulation, medical illustration and other types of illustration. Below we briefly describe some of the different types of manipulations which are useful, and discuss how to achieve these effects on 3D datasets.

4.1 Spatial Transfer Functions

In medical illustration, one commonly sees peel-away effects simulating surgical cuts and other types of surgical procedures. Therefore, what is desired is a deformation-like procedure which can model cuts or splits in a volume. Surgical simulation packages can handle cuts, but they are usually specialized to one model. Other physically based deformation methodologies exist (see [11] for a review) and most focus on deforming the surface without modeling cuts. For visualization and illustration purposes it is not clear that the manipulations must be physically based. Most related is the work of [40] where surgical-like operators were defined on volumetric datasets.

A *spatial transfer function* [12, 27] defines a geometrical transformation of the scalar values of volumetric models to allow different effects such as splitting, squeezing and sweeping. The



peel

move

sweep

unroll

Figure 9 Different manipulations on a volume: peel, move, seep, unroll. Images are from [28, 48, 23, 19].

spatial transfer function is able to handle discontinuities in the rendering method which other more direct data manipulation methodologies have trouble with. The surgical cut shown in Figure 9 is rendered with a spatial transfer function. The volumetric model with the spatial transfer function is rendered by first computing the transformation on the bounding volume, and mapping back to find the actual volumetric values. These can then be composited using a standard ray casting approach. Details can be found in [28].

4.2 Curve-Skeleton Decomposition

To achieve the bend, move/re-pose, and sweep, a *proxy geometry* can be used. The proxy geometry allows the user to easily specify a transformation on the data. One such proxy geometry which is useful is a curve-skeleton. A curve-skeleton is a 1D line-like representation of the object (sometimes referred to as a centerline). A skeleton also is a natural decomposition of many objects and provides a simple way to specify a path (e.g., for virtual navigation). A skeleton of a volumetric object is a useful shape abstraction that captures the essential topology of an object. It also has a cognitive basis in shape comprehension. Motion is also traditionally specified in computer graphics using a skeleton. (Therefore, all of the motion capture available to computer graphics can be available for volumes as well.) The skeleton can be extracted by using a variety of methods (see [18] for more information). Once the skeleton has been obtained, the joints can be chosen [22]. The joints define the areas where bending can occur. Each skeletal segment defines a bounding cuboid about a logical segment of the volume, as seen in Figure 10. The width of the cuboid can be determined from a distance field computed on the volume. The subdivision of the cuboids can be determined

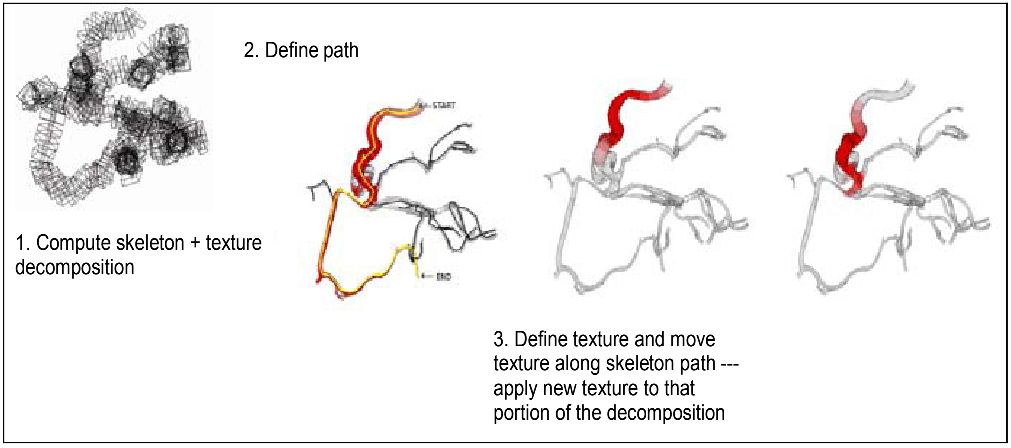
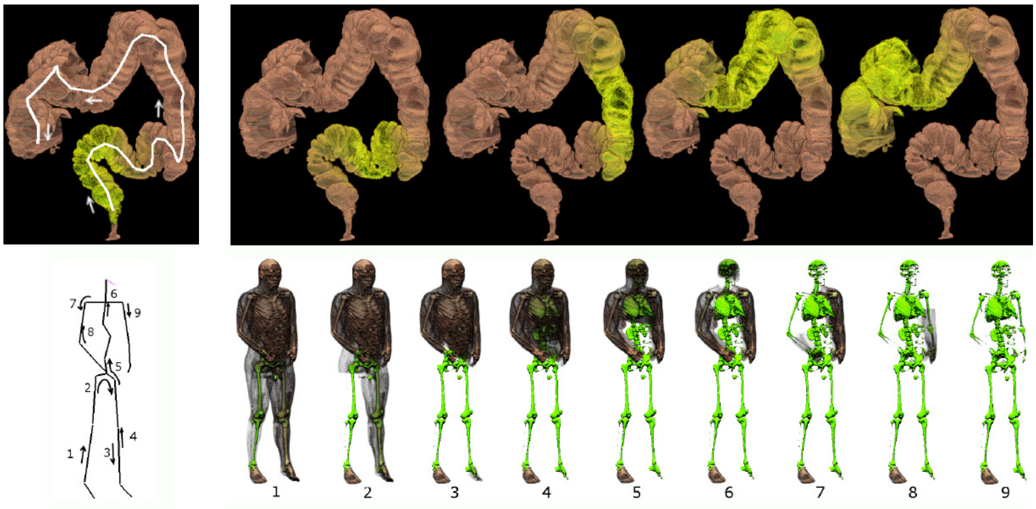

2. Define path
START
END
1. Compute skeleton + texture decomposition
3. Define texture and move texture along skeleton path --- apply new texture to that portion of the decomposition

Figure 10 Block based decomposition of a volume along a skeleton path. The skeleton is divided into small cube-like regions. Each region can be sliced and then rendered. By specifying small regions, different colors, transfer functions, or general transformations can be applied independently. An effect simulating blood flow is shown by moving a texture along a specified path.

based upon the requirements of the visualization (described below). This geometry, though a coarse approximation to the actual shape, suffices for reconstructing most shapes and provides a simple geometry for fast rendering. The boxes are also good for fuzzy bounded volumes, which may not have a definite boundary. An example of the cuboid structure about a central axis is shown for the aneurysm dataset in Figure 10.

The skeleton/cuboids supports both kinematic manipulation and sweeping. One can grab the skeleton and move it as shown in two images from Figure 9. In the first, the hands of the visible man are moved away from the torso to allow better viewing of the torso area. In the second, a volumetric colon is unrolled (its skeleton is straightened). The manipulation can be performed by explicitly computing a new volume in the reposed position [23] or interactively using a technique which only renders the manipulated volume [48]. For interactive manipulation, the cuboids are transformed about the joint, and each cuboid is sliced along the viewport. The slices are mapped back to the original texture to determine the appropriate color value. The texture-mapped polygons are then composited back-to-front which volume renders the deformed volume. Interpolation is done between two end planes of the adjoining bounding boxes which essentially covers the joint area with a stretched volume. The advantage of using this approach is that it requires a minimum of geometric processing and is therefore very fast. If the volume is rotated, the underlying geometry has to be re-sliced and composited. Since each cuboid is sliced independently, it is necessary to sort the sliced polygons along the view direction from back to front. This is achieved in a two-pass rendering algorithm with the aid of a data structure that indexes polygons with respect to their depth coordinate.

The skeleton decomposition also supports selective rendering [48] and sweeps [19]. Selective rendering is where a portion of the dataset is rendered with a different transfer function. This allows one to highlight different parts of a volume for more effective focus+context viewing. By adding motion to the selective rendering, we can create a *swept* representation where a dataset is traversed along a specified path. This is commonly used in medical illustration to produce animations, highlight features or enhance the rendering of a dataset. Techniques for navigation of datasets are used in virtual colonoscopy, where the userís viewpoint is traversed



(a) (b)

Figure 11 Swept based volume rendering. The effect simulates flow through a volume and enables focus+context views. (a) shows the skeleton path (b) is the traversal. See [19] for more examples and a movie.

along a path. Such techniques can be described as inside-out visualizations of the dataset. Here, we are doing an outside-in visualization, where the exploration is enabled by moving a transfer function, in addition to independent control of the userís viewpoint. It is naturally a focus+context technique, as it focuses the viewers gaze on the area being swept while still showing the entire object. It is similar to using a highlighter pen to emphasize parts of an object. An example of this technique used in medical illustration to show blood flow can be seen in [5].

4.3 Dataset Traversal

Traversing (explicitly or mentally) a complex datasets seems to be an essential part of understanding 3D shape and a skeleton or sweep structure can aid in this process. When rendering volumes as 3D textures, transfer functions are usually applied as a lookup color table. This table defines the color and transparency associated with each density value of the volume. The skeletal-based block decomposition allows us to apply a different transfer function to each cuboid of the decomposition along a path at a particular time. A traversal path is selected, and each segment of the path is highlighted at a different time. For rendering, each cuboid of the decomposition is rendered as textured slices. When a slice is being rendered, the proper transfer function is found (i.e., if that slice is part of the segment being highlighted). Since all of the slices are composited together, any number of highlights can be used and the combinations are solved on-the-fly. This creates a feeling of "sweeping out the structure", similar to lighting the structure from within. Two examples of swept visualization can be seen in Figure 11. In the first, parts of a colon are highlighted, and in the second the visible man dataset is traversed while changing the rendering parameters (from bone to skin). Figure 10 shows a swept volume simulating blood flow. More details of the algorithm are given in [19].

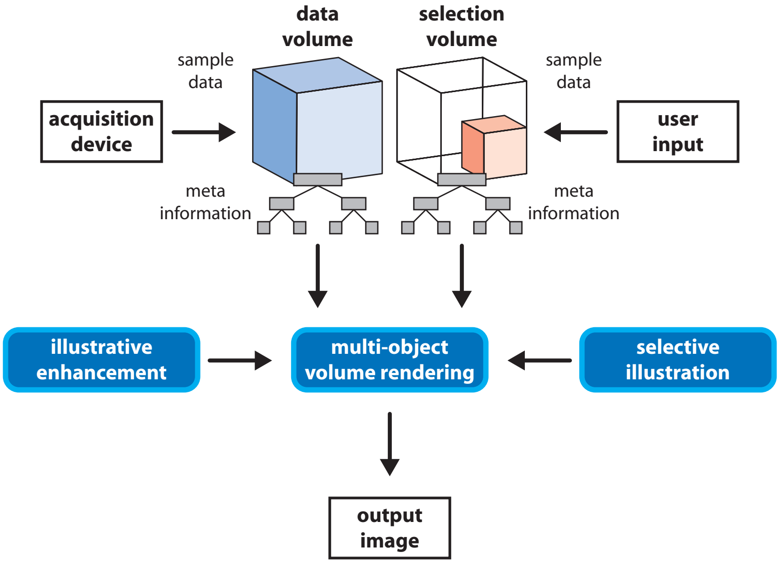


Figure 12 Conceptual overview of our direct volume illustration environment.

5 Interactive Design of Illustrations from Volume Data

In this section we discuss several concepts used in the design of VolumeShop [7], a system for interactive generation of illustrations directly from volume data. 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 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.

The architecture of VolumeShop 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. 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 the system is given is Figure 12.

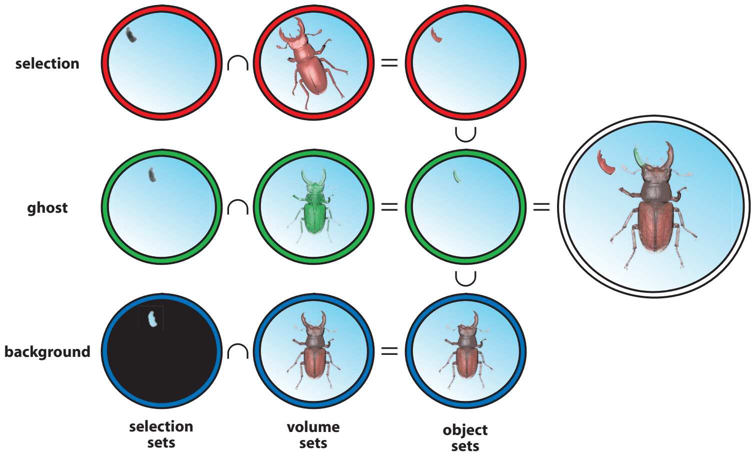


Figure 13 Overview of the basic multi-object combination process for background, ghost, and selection: the intersection between selection sets and volume sets results in object sets which are then combined.

5.1 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.

VolumeShop's 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 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 normally exist, if at all, only implicitly. The approach uses fuzzy set arithmetic to derive the selection, ghost, and background objects (objects sets) as an intersection of the selection volumes (selection sets) and the opacity transfer function specified for the data volumes (volume sets), as illustrated in Figure 13.

Additionally, the user is supplied with 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. Two dimensional intersection transfer functions are employed for this purpose. An intersection transfer function specifies the color and opacity at a resample location based on the scalar volume of the volumetric objects present at that location. Per definition background and ghost never intersect. The selection, however, can intersect either the background, the ghost, or both. 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. VolumeShop provides 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). Furthermore, there are several presets where the opacity is computed from the one-dimensional opacity transfer functions by one of the compositing operators derived by Porter and Duff [42]. The color can be specified arbitrarily. Additionally, the user can paint on the two-dimensional function

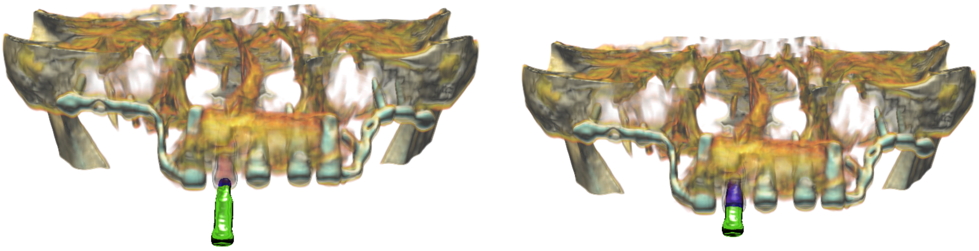


Figure 14 Using intersection transfer functions to illustrate implant placement in the maxilla. As the selection (green) is moved into the ghost (faint red), the intersection transfer function causes it to be displayed in blue.

using a gaussian brush to highlight specific scalar ranges. Figure 14 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.

5.2 Illustrative Enhancement

Illustration is closely related to non-photorealistic rendering methods, many of which attempt to mimic artistic styles and techniques. VolumeShop uses 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. In order to integrate many of these models, VolumeShop uses a two-dimensional lighting transfer function for shading objects. 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.

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. This approach allows different illustrative shading models to be 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. We can thus create a lighting transfer function where we set ambient, diffuse and specular components to zero where $\hat{N} \cdot \hat{L} \approx 2(\hat{N} \cdot \hat{H})$. Other methods, such as cartoon shading [14] or metal shading [24] can be realized in a straight-forward manner and combined with effects like contour enhancement. Figure 15 shows an image rendered using four different lighting transfer functions.

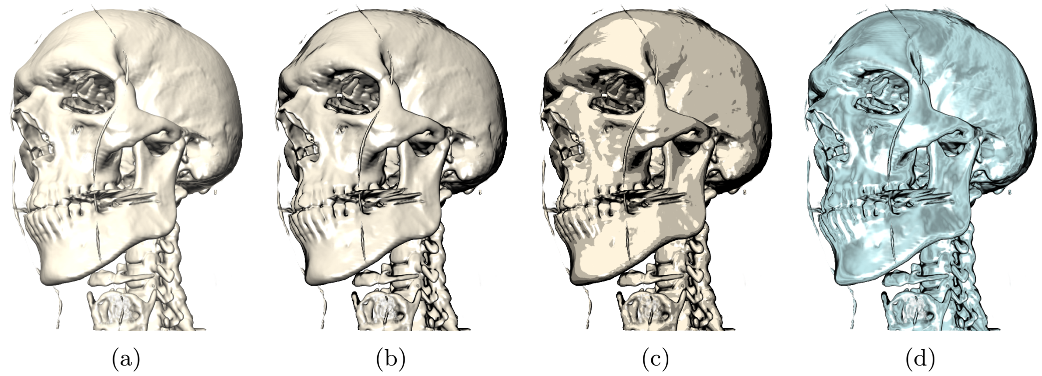


Figure 15 The same data set rendered with four different lighting transfer functions (the lighting transfer function for each image is displayed in the lower left corner - ambient, diffuse, specular, and opacity enhancement components are encoded in the red, green, blue, and alpha channel, respectively). (a) Standard Phong-Blinn lighting. (b) Phong-Blinn lighting with contour enhancement. (c) Cartoon shading with contour enhancement. (d) Metal shading with contour enhancement.

5.3 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.

5.3.1 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. [54] introduced importance-driven volume rendering, a general framework for determining which object is to be cut by using an importance function. VolumeShop's simplified three-object setup allows a 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.

Ghosting [6] 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 spatial location of the object in focus. 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 16.

5.3.2 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 spatial 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.

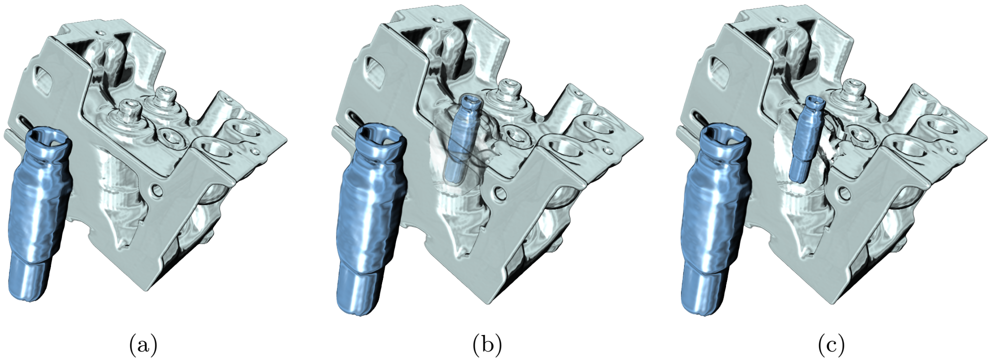


Figure 16 Different degrees of ghosting - from no ghosting (a) to full cutaway (c).

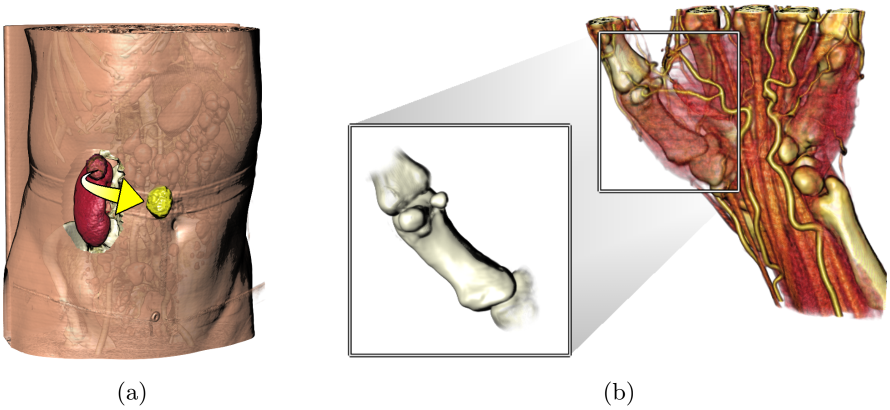


Figure 17 Using different artistic visual conventions. (a) Illustrating a tumor resection procedure using an automatically generated arrow. (b) Detailed depiction of a hand bone using a fan.

VolumeShop provides the user with different kinds of visual enhancements inspired by these conventions.

For example, 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, VolumeShop employs 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 17 (a) shows an example for the use of arrows.

Another metaphor used are "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 17 (b)).

Apart from controlling visual appearance, it is useful to provide different interaction types based on the role of an object in the illustration. For example, the user can "pin" down the current selection, i.e. its on-screen location will remain static, but it is still affected by rotations. A rotation of the viewpoint causes the same relative rotation of the object. 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.

5.4 Conclusion

The use of an optimized GPU volume rendering algorithm with multi-object capabilities (see [7] for more details) allows us to provide a responsive interface for interactive generation of volume-based illustration using the techniques described in this section. VolumeShop is an open architecture and supports extensions using a plug-in mechanism. For more information (including a downloadable version) see: `http://www.cg.tuwien.ac.at/volumeshop`.

6 Illustrative Visualization for 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 the experiences with planning 20 neck dissections based on CT datasets ([32] and [53]).

6.1 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 muscles, nerves or blood vessels determine the surgical strategy. If for example the *A. carotis interna* is infiltrated, the patient is regarded as not resectable. Visualization techniques should be developed to support decisions regarding the surgical strategy.

6.2 Conventional Surgical Planning

Surgical planning is 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. 3D renderings are rarely used and many surgeons are not convinced of the additional value of 3D renderings at all. This attitude has 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 are primarily used to give an overview. 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 they did it before and can use the advanced techniques additionally. The most benefit can be achieved if 2D and 3D visualizations are carefully synchronized, e.g. with respect to the selection and emphasis of an object (Figure 18).

6.3 Advanced Surgical Planning

Advanced surgical planning requires reliable segmentation results. Segmentation of the relevant structures is a challenging task and an area of ongoing research. In our case study,

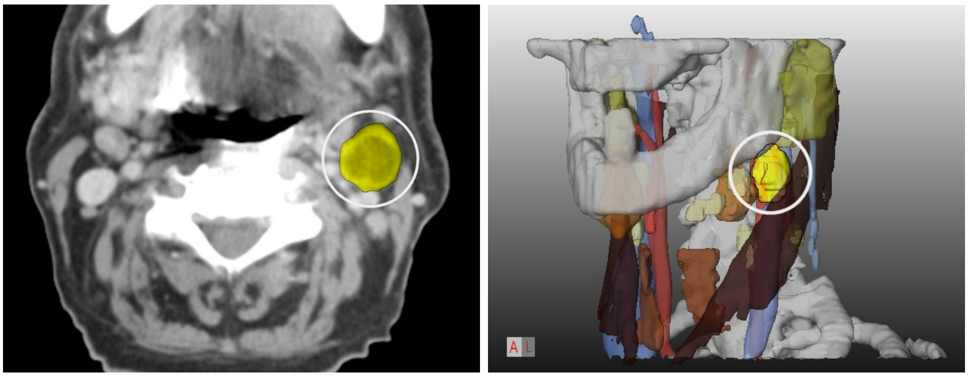


Figure 18 The lymph node emphasized in the 3D visualization is simultaneously emphasized in the original slices.

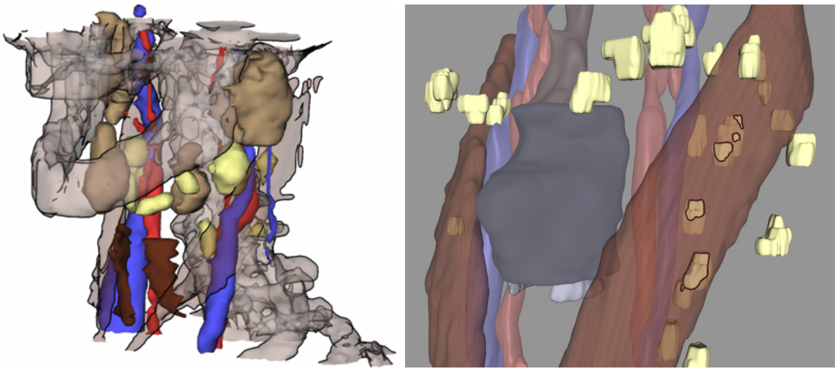


Figure 19 Left: Illustrative rendering for neck dissection planning. Silhouettes are generated for the bones which serve as anatomic context. Right: Silhouette rendering reveals lymph nodes which touch and potentially infiltrate a critical structure.

the segmentation of all relevant structures is accomplished with NeckVision, a dedicated software assistant [16].

Illustrative Rendering. Silhouette rendering is employed for two purposes. The obvious use is to indicate the context objects, such as bones (Figure 19, left). In addition, silhouettes may be used to discriminate two classes of objects; those which exhibit a certain feature (or are more “interesting") are rendered with silhouettes enabled whereas the remaining objects are drawn without silhouettes. In neck surgery planning, many lymph nodes have to be explored by the user. In particular, lymph nodes which are enlarged and touch a critical structure are essential and thus rendered with silhouettes (Figure 19, right). Surgical users regard this a substantial help since otherwise it is not recognizable whether the lymphnode is (only) close to a critical structure or touches it. The combination of silhouette-, surface- and volume rendering is accomplished with a scenegraph-based approach [52].

Approximative Rendering of Nervs. For some structures, such as nerves, a complete segmentation is not possible. Nerves are very small 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 hard to distinguish from its surrounding. As a consequence, only in some slices a nerve could be identified at all. Nevertheless, the rough course of the nerves

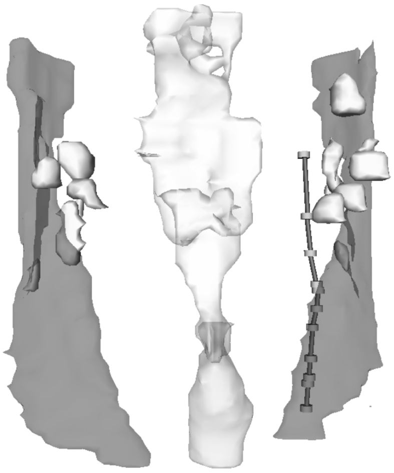


Figure 20 Approximate visualization of N. facialis for neck dissection planning. In the lower portion, the density of disks is higher which implies a more reliable visualization.

is essential for surgical planning. Anatomic experience shows that nerves proceed almost linearly and do not deviate strongly from the straight connection between positions found in some slices. Since it is important to prevent the injury of nerves, approximate visualizations should be generated where the segmented portions are emphasized and the part in between is reconstructed as linear connection. The emphasis of the segmented portions is carried out with small cylindrical disks (see Figure 20).

6.4 Enhancing Slice-based Visualizations

In the following, we describe how slice-based visualizations can be enhanced to give an overview on anatomic structures. This description is motivated by the importance of slice-based views for surgery planning and is based on [53].

For an overview of the segmented structures in a slice-based visualization, it is essential to present the relative position of structures in the current slice as well as their positions within the whole set of slices. In the following, we refer to the slice number as the z-dimension. The visualization problem that occurs here is similar to time scheduling. In this area, various techniques have been developed to visualize data entries and their temporal relations. Graphical overviews should present appointments distinguishable from each other and the temporal relations between them (see for example, the LifeLines project [41]). Translated to slice-based visualization, the interval of slices (z_{min}, z_{max}) of the segmented structures corresponds to the lengths of appointments.

Similar to temporal overviews in time scheduling, we attach a narrow frame next to the cross sectional image that represents the overall extent of slices in the volume data set. The top and bottom boundary of the frame correspond to the top and bottom slice of the dataset. Each segmented structure is displayed as a bar at the equivalent vertical position inside this frame. The vertical extent of the bar represents the interval (z_{min}, z_{max}) for each structure. We refer to this combination of bars as LiftChart and regard it as a widget which provides interactive facilities to locate structures and slices. The LiftChart widget can be used for interaction and navigation. The horizontal slice indicator is operated like a normal scrollbar and moves through the slices. If the mouse is placed over a particular bar, information about

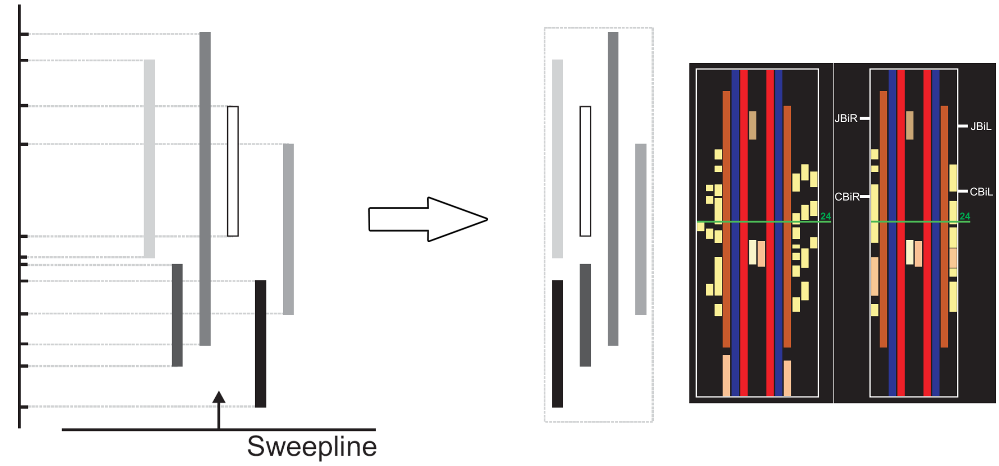


Figure 21 Left: The Sweepline moves from bottom to top and places the next available bar at the leftmost unused column. Middle: Each anatomic structure is represented by one bar. The LiftChart is divided in three parts: one part for structures on the left and on the right side each and one part for structures in the middle. Right: Lymphnodes are aggregated in one column. Additional landmarks serve as orientation aids.

the underlying anatomic structure is shown.

In order to optimize the LiftChart's use of screen space, the bars are ordered with a Sweepline algorithm. The slices are processed from bottom to top and for each anatomic structure a bar is drawn in the leftmost available column. If a structure ends, the respective column is freed again and can hold the bar of a new structure starting farther above (Figure 21, left).

The LiftChart enhances the recognition of relative positions of structures in the volume dataset. To simplify the correlation between the slice view and the LiftChart, the color and style of the bars should correspond with the color and style of the structures displayed in the slice view. The colors represent different categories: Lymph nodes (yellow), Muscles (brown), veins (blue), arteries (red) and the lung (skin-colored). The green line denotes the current slice. The currently displayed slice is depicted by a horizontal line and annotated with the slice number. To visualize not only the z-distribution of structures, but also their horizontal position, several arrangements of the bars have been explored (Figure 21). In the simplest form, each anatomic structure is represented by one bar. Since some anatomic structures have a defined side, the LiftChart may be divided in three parts: one part for structures on the left and on the right side each and one part for structures in the middle (see the left image of Figure 21). In particular, the separation in lymph nodes located at the left and right side is motivated by the surgical strategies (left and/or right-sided surgery).

It is also possible to group bars which belong to the same category of anatomical structures to minimize the horizontal extent of the widget (see Figure 21, right, where the lymph nodes are aggregated into one column). Furthermore, landmarks for orientation in the dataset may be displayed. As an example relevant for neck dissections, in the right image of Figure 21, the bifurcations of the *Vena Jugularis* (JBiL/JBiR) and *Arteria Carotis* (CBiL/CBiR) are indicated.

Slice-based Visualization of Safety Margins. Safety margins are essential for preoperative planning and intraoperative navigation. To prevent damage to structures at risk, the distances of the surgical tool to such structures have to be carefully observed during

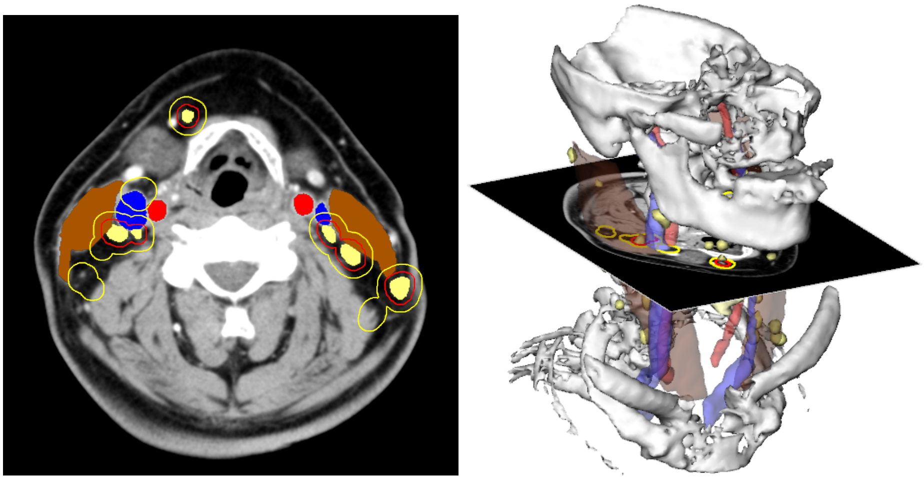


Figure 22 Depicting safety margins around pathologic lymph nodes as special halos on the current slice. Left and right are swapped in the slice view, because the viewing direction is from bottom to top. Left image: the safety margins of $2mm$ (red) and $5mm$ (yellow) are shown as the user would see them in the slice view. Right image: the position of the slice with the displayed structures is shown for comparison.

the surgery. Halos (in the original sense of the word) can convey this distance information. Therefore, for all structures at risk an Euclidean distance transform [4] is performed and the resulting distance information is overlaid on the slice image. We considered color-coding the distance information but rejected this idea, since a color map conveys too much information not relevant for the surgical strategy. Depicting important distance thresholds as halos by drawing isolines representing 2 and $5mm$ distances reduces the information to a few categories which are easy to interpret (Figure 22).

6.5 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 correctly delineated is probably more important than the details of their visualization. Valuable computer support for surgical planning requires high-quality image acquisition, reliable and fast image analysis techniques *and* comprehensible visualizations. The combination of 2D and 3D renderings is essential for the acceptance of computer-supported surgical planning systems.

The use of illustrative techniques for neck dissection planning is based on discussions with clinicians. Although illustrative techniques are not wide-spread in surgical planning, our research indicates, that they 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 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 select appropriate techniques and parameters. These steps need to be strongly supported since the time for surgical planning remains severely restricted.

7 Conclusion

In this paper we have presented illustrative techniques for the visualization of volumetric data. We have shown that the concept of focus+context is a particularly useful metaphor for dealing volume data due to their inherent complexity.

Lens-based distortion is a powerful framework for the exploration of volume data even if no additional information is available. When segmentation data is present, feature-based lenses can be used to enhance fine details in an intuitive way. The "hands-on" approach to volume visualization presented in Section 4 is a natural way for examining volume data. Three-dimensional interaction allows users to examine objects in a similar way as he they would do in real life. Section 5 showed that the use of advanced volume visualization techniques makes it possible to interactively generate expressive illustrations based on real data rather than hand-made geometric models. And finally, in Section 6 we have seen that medical applications such as treatment planning can greatly benefit from illustrative visualization.

Illustrative visualization has generated a considerable amount of interest in the community. While we can learn a lot by studying the world of illustration, many new challenges arise when adapting traditional techniques to computer-based visualization. The aspect of interaction seems of particular importance in this context as traditional illustration does not feature any interaction capabilities.

References

1 T. A. DeFanti B. H. McCormick and M. D. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6), 1987.

2 M. J. Bailey. The use of solid rapid prototyping in computer graphics and scientific visualization. In *ACM SIGGRAPH 1996 Course Note 37: The Use of Touch as an I/O Device for Graphics and Visualization*, 1996.

3 E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of ACM SIGGRAPH 1993*, pages 73–80, 1993.

4 G. Borgefors. Chamfering: A fast method for obtaining approximations of the euclidean distance in N dimensions. In *Procedings of the Scandinavian Conference on Image Analysis 1983*, pages 250–255, 1983.

5 M. Brierley. Arteriovenous malformations, 2000. `http://brainavm.oci.utoronto.ca/swf/intro.html`.

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

7 S. Bruckner and M. E. Gröller. VolumeShop: An interactive system for direct volume illustration. In *Proceedings of IEEE Visualization 2005*, pages 671–678, 2005.

8 M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceeding of the IEEE Symposium on Information Visualization 1996*, pages 46–53, 1996.

9 M. S. T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Extending distortion viewing from 2D to 3D. *IEEE Computer Graphics and Applications*, 17(4):42–51, 1997.

10 K. L. Chelule, T. J. Coole, and D. G. Cheshire. Fabrication of medical models from scan data via rapid prototyping techniques. In *Proceedings of Time-Compression Technologies 2000*, pages 45–50, 2000.

11 M. Chen, C. Correa, S. Islam, M. W. Jones, P.-Y. Shen, D. Silver, S. J. Walton, and P. J. Willis. Deforming and animating discretely sampled object representations. In *Proceedings of Eurographics 2005 – State of the Art Reports*, pages 113–140, 2005.

12 M. Chen, D. Silver, A. S. Winter, V. Singh, and N. Cornea. Spatial transfer functions: a unified approach to specifying deformation in volume modeling and animation. In *Proceedings of the International Workshop on Volume Graphics 2003*, pages 35–44, 2003.

13 P. Cignoni, C. Montani, , and R. Scopigno. Magicsphere: an insight tool for 3d data visualization. *Computer Graphics Forum*, 13(3):317–328, 1994.

14 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.

15 M. Cohen and K. Brodlie. Focus and context for volume visualization. In *Proceedings of Theory and Practice of Computer Graphics 2004*, pages 32–39, 2004.

16 J. Cordes, J. Dornheim, B. Preim, I. Hertel, and G. Strauß. Preoperative segmentation of neck CT datasets for the planning of neck dissections. In *Proceedings of SPIE Medical Imaging 2006*, pages 1447–1456, 2006.

17 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.

18 N. Cornea, D.Silver, and P. Min. Curve-skeleton applications. In *Proceedings of IEEE Visualization 2005*, pages 95–102, 2005.

19 C. D. Correa and D. Silver. Dataset traversal with motion-controlled transfer functions. In *Proceedings of IEEE Visualization 2005*, pages 359–366, 2005.

20 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.

21 G. W. Furnas. Generalized fisheye views. In *Procedings of ACM SIGCHI 1986*, pages 16–23, 1986.

22 N. Gagvani and D. Silver. Parameter-controlled volume thinning. *Graphical Models and Image Processing*, 61(3):149–164, 1999.

23 N. Gagvani and D. Silver. Animating volumetric models. *Graphical Models and Image Processing*, 63(6):443–458, 2001.

24 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.

25 B. Gooch and A. Gooch. *Non-Photorealistic Rendering.* AK Peters, 2001.

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

27 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.

28 S. Islam, D. Silver, and M. Chen. Volume splitting and its applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):193–203, 2006.

29 T. Keahey and E. Robertson. Techniques for non-linear magnification transformations. In *Proceedings of the IEEE Symposium on Information Visualization 1996*, pages 38–45, 1996.

30 R. Kosara, S. Miksch, and H. Hauser. Focus+context taken literally. *IEEE Computer Graphics and Applications*, 22(1):22–29, 2002.

31 M. Kreuseler, N. Lopez, and H. Schumann. A scalable framework for information visualization. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, pages 27–38, 2000.

32 A. Krüger, C. Tietjen, J. Hintze, B. Preim, I. Hertel, and G. Strauß. Interactive visualization for neck dissection planning. In *Proceedings of EuroVis 2005*, pages 295–302, 2005.

33 E. LaMar, B. Hamann, and K. I. Joy. A magnification lens for interactive volume visualization. In *Proceedings of the Pacific Conference on Computer Graphics and Applications 2001*, pages 223–232, 2001.

34 Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.

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

36 M. Levoy and R. Whitaker. Gaze-directed volume rendering. *Computer Graphics*, 24(2):217–223, 1990.

37 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.

38 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.

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

40 M. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of the IEEE Visualization 2003*, pages 401–408, 2003.

41 C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. LifeLines: Visualizing personal histories. In *Proceedings of ACM SIGCHI 1996*, pages 221–227, 1996.

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

43 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.

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

45 P. K. Robertson. A methodology for choosing data representations. *IEEE Transactions on Computer Graphics and Applications*, 11(3):56–68, 1991.

46 M. Sheelagh, T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Making distortions comprehensible. In *Proceedings of the IEEE Symposium on Visual Languages 1997*, pages 36–45, 1997.

47 P. Shirley. *Fundamentals of computer graphics.* AK Peters, 2005.

48 V. Singh, D. Silver, and N. Cornea. Real-time volume manipulation. In *Proceedings of the International Workshop on Volume Graphics 2003*, pages 45–51, 2003.

49 T. Strothotte and S. Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation.* Morgan Kaufmann, 2002.

50 N. Svakhine, D. S. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3):31–39, 2005.

51 A. Tappenbeck, B. Preim, and V. Dicken. Distance-based transfer function design: Specification methods and applications. In *Proceedings of Simulation und Visualisierung 2006*, pages 259–274, 2006.

52 C. Tietjen, T. Isenberg, and B. Preim. Combining silhouettes, surface, and volume rendering for surgery education and planning. In *Proceedings of EuroVis 2005*, pages 303–310, 2005.

53 C. Tietjen, B. Meyer, S. Schlechtweg, B. Preim, I. Hertel, and G. Strauß. Enhancing slice-based visualizations of medical volume data. In *Proceedings of EuroVis 2006*, pages 123–130, 2006.

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

55 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.

56 L. Wang, Y. Zhao, K. Mueller, and A. E. Kaufman. Generating sub-resolution detail in images and volumes using constrained texture synthesis. In *Proceedings of IEEE Visualization 2004*, pages 75–82, 2004.

57 L. Wang, Y. Zhao, K. Mueller, and A. E. Kaufman. The magic volume lens: An interactive focus+context technique for volume rendering. In *Proceedings of IEEE Visualization 2005*, pages 367–374, 2005.

58 X. Wei, A. E. Kaufman, and T. J. Hallman. Case study: visualization of particle track data. In *Proceedings of IEEE Visualization 2001*, pages 465–468, 2001.

59 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.

60 J. Zhou, M. Hinz, and K. D. Tönnies. Focal region-guided feature-based volume rendering. In *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission 2002*, pages 87–90, 2002.