INTERACTIVE VOLUME VISUALIZATION AND EDITING METHODS FOR SURGICAL APPLICATIONS

by Can Kirmizibayrak

B.S. in Electrical-Electronics Engineering, May 2003,

Bogazici University, Turkey

M.S. in Telecommunications and Computers, May 2005,

The George Washington University

A Dissertation Submitted to

the Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

Dissertation directed by

James K. Hahn
Professor of Engineering and Applied Science

Abstract

Volumetric imaging modalities are increasingly being used in surgical applications. However, visualizing the 3D information contained in these datasets effectively on a 2D display is a challenging problem. Most important issues to overcome are the occlusion of different anatomical features of each other and the difficulty of visualizing the depth information of a given pixel in a 2D image. Moreover, the resulting image is aimed to be used to guide actions performed on the patient, therefore the mental registration of real and virtual spaces has to be considered when designing visualization and interaction approaches.

This work proposes an interactive focus + context visualization method that uses the Magic Lens interaction scheme to select important parts of volumetric datasets while displaying the rest of the dataset to provide context for the mental registration process. The Magic Lens paradigm is extended to handle arbitrarily shaped selection volumes, enabling interactive volume editing to visualize different anatomical structures from multiple datasets in a single coherent view. Capabilities of modern graphics hardware are used to achieve real time frame rates. The implementation of these methods introduces novel technical contributions to view and to select arbitrarily shaped sub-volumes in real-time using polygon-assisted raycasting using meshes as proxies to store selection information. These approaches enable sub-voxel accuracy in selecting and rendering volumetric regions using significantly smaller storage space compared to using lookup volume textures for selection. Proposed methods are applied to a gesture-based interaction interface, and user studies are undertaken to evaluate the effectiveness and intuitiveness of this interface in volume rotation and target localization tasks.

Table of Contents

List of Figures

List of Tables

List of Acronyms

F+C	Focus + Context
CT	Computerized Tomography
MRI	Magnetic Resonance Imaging
GPU	Graphics Processing Unit
2/3D	Two/Three Dimensional
NPR	Non-Photorealistic Rendering
HCI	Human-Computer Interaction
NUI	Natural User Interface
GBI	Gesture-based interface
DOF	Degrees-of-Freedom
CAS	Computer Aided Surgery
IGS	Image Guided Surgery
CFD	Computational Fluid Dynamics
K2HR	Kinect Two-Hand Rotation Interface (Experiment I)
TMR	Traditional Mouse Rotation Interface (Experiment I)
KS	Kinect Slice Interface (Experiment II)
MS	Mouse Slice Interface (Experiment II)
ML	Magic Lens Interface (Experiment II)

Chapter 1 - INTRODUCTION

Advances in medical imaging technology have increased the use of a variety of imaging modalities in medicine. In addition to traditional two-dimensional (2D) modalities (such as radiography or ultrasound imaging), three-dimensional (3D) datasets are also widely used. Examples to 3D datasets include magnetic resonance imaging (MRI) and its variants (e.g. functional-MRI, MR-diffusion tensor imaging), computerized tomography (CT), nuclear imaging (e.g. positron emission tomography (PET), single photon emission computed tomography (SPECT)). The 3D data contained by these modalities in effect enables the physicians to be able 'look inside' the patient by providing information of internal patient anatomy that normally cannot be seen. However, this benefit comes with some drawbacks: the information presented in these datasets varies depending on the properties of the imaging technology used and may be difficult to interpret. Therefore, presenting a 3D dataset to maximize the information content is a challenging problem.

An important application domain where 3D medical datasets are used is surgical interventions. A successful surgery aims to minimize the damage to the patient while producing the desired outcome in the minimum amount of time possible. For this, knowing detailed information about the patient's anatomy before and during the surgery is very beneficial. A surgical example for this is tumor resection, where a crucial trade-off exists between making sure no malignant cells are left behind and removing too much tissue, which would damage the surrounding anatomical structures. Furthermore, these tasks have to be completed as quickly as possible to minimize the time under anesthesia as well as because of scheduling (and therefore financial) reasons. Again, in the resection example, knowing tumor location and boundaries would increase the accuracy of the incision as well as the removal, improving the success rate and recovery time [1]. 3D medical datasets provide additional information to surgeons to improve the success rate and accuracy of various surgical tasks [2]. Consequently, visualization of these datasets to maximize their benefits has become an important research area.

This dissertation presents a novel interactive visualization approach for surgical applications. The focus of the work is on several aspects of medical visualization: developing novel visualization methods for surgical planning and intra-operative guidance, developing and evaluating interaction methods for volumetric data exploration and manipulation, and implementation of these methods in real-time using the features of modern graphics hardware.

1.1. Motivation

Surgical interventions require successful completion of many complex cognitive and physical tasks. A surgeon must take into account many variables before and during the surgery to make correct decisions: the anatomical structure of the patient from pre- and intra-operative observations, external information from various medical imaging modalities and his domain knowledge and expertise about the current procedure. Combining all these information sources to make an informed decision is an extremely challenging task. The motivation for medical visualization is to make this process easier and more accurate. The main purpose of such visualization approaches is to present the user the most informative 'picture' from all available information sources while filtering out the irrelevant parts. However, deciding what parts of the information is important is not a trivial task. We propose an interactive volume visualization system to help surgeons explore and understand volumetric medical datasets, and aid them in applying this information to surgical applications to improve their success rate and accuracy.

Volume visualization is an important subset of medical visualization because of the ever-increasing availability of medical volumetric datasets and recent advances in graphics hardware, which made implementation of high quality visualizations possible on affordable consumer systems. Interaction is also a significant and emerging research field. The widespread availability of end-user interaction systems (examples of which include multi-touch cellphone screens to interactive gaming systems such as Nintendo's Wii) has sparked an interest in interactive approaches in many domains. This trend can be seen from the increase in number of accepted/submitted papers to the premier conference in the field (ACM Conference on Human Factors in Computing Systems (CHI)) which respectively has gone from 75/468 in 2005 to 277/1130 in 2009 [3]. This in turn has enabled applications of these novel approaches to medicine; however, because of the extensive validation and verification methods necessary for deploying new medical approaches, the migration rate of new interactive techniques to surgical domain has been low. The approaches presented in this dissertation are designed to improve on current surgical practices and to improve applicability to real situations.

1.2. Problem Statement

The benefits of volumetric image datasets can be summarized as providing physician information that they normally would not be able to see without surgical intervention. In some cases, the information contained in the images is not visible to the naked eye at all, for instance, without using functional MRI (fMRI) imaging a neurosurgeon would not be able to see distribution of blood flow in various areas of the brain even with exploratory surgery. However, since these datasets can include a multitude of data values, analysis and interpretation of 3D imaging data is a very challenging problem. Volume visualization techniques aim to present these datasets in an informative manner to maximize their benefits. A number of problems can arise because of the nature of the datasets and the display technologies used to view them.

The first challenge with the use of 3D medical datasets is the fact that these datasets will eventually have to be displayed on a 2D monitor. Even though stereo or 3D displays are available, these technologies are still not mature enough for them to be ubiquitously used without disrupting the current workflow. Therefore, in almost all medical visualization applications a 3D dataset will have to be displayed on a 2D screen, resulting in some loss of information during this projection. Two main problems that need to be overcome are occlusion and depth ambiguity. Occlusion is the result of different parts of the dataset blocking each other. This is a consequence of the very advantages medical imaging technologies provide: they allow ‘seeing through’ things; however, since this is different information than what our brains are used to interpret, it is more difficult to understand [4]. Depth ambiguity is similar in the same context: removing additional visual cues such as the disparity between the images of two eyes or modifying the visibility order of objects in a scene makes it more difficult to differentiate the depth value of a pixel in a rendered 3D image. When multiple modalities are present for the same patient, these problems are amplified because the amount of information to be visualized increases.

Another challenge this dissertation aims to tackle is the fact that surgical planning and image guided surgery systems are designed to help actions performed in the *real world*. Throughout this work, the term *real world* is used to represent the space the physician performs his actions, or in practical terms can be thought of as equivalent to the patient for almost all surgical applications. The *virtual world*, on the other hand, represents data that is acquired from the *real world* (either before or during the procedure), and is being visualized. In other words, the information displayed to the surgeon has a connection to the patient, and one of main objectives of image guided surgery systems is to make this connection (or cognitive mapping) more intuitive and easier. This process is called mental registration. One of the vital parts of this research is the interactivity schemes that help the surgeons establish this connection, both by providing an interaction interface that can be used in the real world and by providing real-time frame rates so that the feedback from the interaction is received immediately. Currently, either traditional interfaces such as the mouse or trackers are used in surgical interaction tasks. These technologies have a number of problems for them to be used in a surgical setting: sterilization, cost, calibration and accuracy, disruption of workflow to name a few. Replacing these technologies with a natural user interface using gestures can alleviate many of these problems. This dissertation proposes and evaluates such a gesture-based interaction method to be used in common surgical visualization tasks such rotation, slice and sub-volume selection and volume editing.

1.3. Proposed Solution

The premise that motivates our solution is that parts of a volumetric dataset can be more 'interesting' than the rest; however, it is challenging to have this distinction automatically. Moreover, even the 'unimportant' parts can add to the overall understanding of the dataset by providing the relationship to the overall data structure. The proposed solution is a real time interactive visualization method that uses different rendering styles for localized volumes. By displaying the user-selected important parts of the dataset differently, the physician's understanding of these regions can be improved. The surroundings of these user-selected sub-volumes are shown (possibly in less detail) to provide the relationship to the rest of the dataset and to the real world. The interface is designed as a volumetric lens/brush, whose location is controlled by the user via interaction in real or virtual space. By using this sub-volume like a brush, the user can interactively select arbitrarily shaped regions on the volume and corresponding rendering styles. The designed sub-volume selection process can be used as a volume manipulation tool that can be used for surgical planning or during surgeries. The proposed manipulations of the visualization (e.g. sub-volume selection, changing the viewpoint) can be done with a gesture-based interaction interface, eliminating the need for tactile interaction methods that can cause problems with sterilization and disruption of the workflow in the operating room setting.

1.4. Original Contribution

The main contribution of this dissertation is the introduction of an interactive visualization framework that enables arbitrarily shaped sub-volume selection and visualization with real time frame rates. A novel polygon assisted raycasting based selection process is used to select multiple regions from multiple co-registered datasets, which can be displayed with transfer functions to highlight desired anatomical structures to form a single cohesive and informative view. The interaction can be done with a gesture-based interface, which has been evaluated via user studies. The results of these studies are analyzed to provide insight for strengths and weakness of application of gesture-based interfaces to medical visualization tasks.

The methods presented are, like most volume visualization tasks, computationally expensive. From a technical point of view, the real-time implementation of these methods makes several contributions, including using GPU and polygon-assisted methods for achieving sub-voxel accuracy in sub-volume display. The implementation methods presented achieve focus+context visualization with same frame rates as traditional raycasting. Moreover, selection and rendering of multiple volume datasets using multiple transfer functions assigned to user-selected sub-volumes can be performed in real-time. The storage of selection information using polygonal meshes requires significantly smaller space compared to storage using volumetric data structures.

1.5. Document Organization

This document is organized as follows: in Chapter 2, the previous work in medical visualization will be reviewed. Chapter 3 will start by explaining the visualization approach taken, and will continue with information about proposed visualization concepts, technical details about their implementation and user studies conducted. Chapter 4 will discuss our application domain, image guided surgery and surgical planning, and how the techniques presented can solve the problems in these areas. In Chapter 5, the experimental setup and results will be introduced. The conclusion and future work will be presented in Chapter 6.

Chapter 2 - PREVIOUS WORK

Our work is inspired by numerous previous research efforts in data and medical visualization. In this section, these methods will be overviewed; with emphasis given to their advantages, shortcomings and how our approach aims to improve these methods. We will start by an overview of research efforts on medical visualization, with an emphasis given to volume visualization. We will continue with the current visualization methods used in clinical practice, followed by a discussion about the human factors issues in medical visualization applications.

2.1. Visualization for Medical Applications

There have been many research efforts to visualize medical datasets. In its essence, medical data visualization can be viewed as an information visualization problem. Information visualization researchers have long studied visualization strategies to present complex datasets effectively. In most complex datasets, the main problem is the abundance of information. The user usually has to explore the dataset and examine subsets of the contained information. Earlier examples of such datasets include maps and graphs. For instance on a map, displaying every road in any given country would require a resolution that is higher than most displays can produce and human eye can differentiate. One obvious solution to this problem is only displaying the parts that are interesting to the user and filtering out the rest. However, this approach has two main drawbacks: First, figuring out what is important in a dataset is one of the main reasons why a dataset is visualized, therefore it is usually difficult to know what parts of the dataset is important beforehand. Second, the less important parts of the dataset can be valuable to help user understand the relationship of the interesting regions to the rest of the dataset. Coming back to the map example: seeing the road currently traveled on may be the part that contains the most important information on a map, but showing for example important landmarks or highway exits in surrounding areas can help a person figure out his current location with respect to the overall route. These problems led to development of paradigms that can display detailed information along with its surroundings to provide context, which has been known as focus+context visualization. This paradigm provides the inspiration for the methods developed in this dissertation. This section will present an overview about focus+context visualization, along with the Magic Lens™ interaction scheme, which is an interaction metaphor that complements focus+context. Additional information will be given about other medical visualization approaches with comparisons with our method.

2.1.1. Focus + Context Visualization

In information visualization, focus+context visualization is a well-established paradigm that aims to display an overview (context) and detail information (focus) together; the focus region being what the user is interested in, while the context is presented "to help make sense of how the important information relates to the entire data structure" [5]. These two types of presentation are combined "in a single (dynamic) display, much as in human vision" [6]. This type of approach is very fitting to the problems in volume visualization: as displaying the context in less detail (or with more transparency) would alleviate some of the occlusion problems, while still giving the physician some information to perform the cognitive mapping between real and virtual spaces.

Early applications of focus+context visualization were usually aimed at 2D datasets such as graphs and maps. The problem to be overcome was the insufficient screen space and resolution to display all available information. Spatial distortion was one of the solutions to this problem, which gives the focus region more screen space and shrinks the context region. Lamping et al. [7] applied this idea to graph visualization using hyperbolic image space distortion (Figure 2.1, [7]). Later, as volume rendering became commonplace, distortion techniques were applied to volumetric datasets. Cohen and Brodlie [8] used inverse mapping the traversed rays in raycasting to the original dataset according to various distortion functions. Recently, Wang et al. [9] proposed an improved implementation of a similar idea that also aims to preserve features in the dataset (Figure 2.2, [9]) The problem with distortion-based approaches is that the size/depth relationships are modified, which may lead to confusing and inconsistent results in diagnostic and surgical applications.

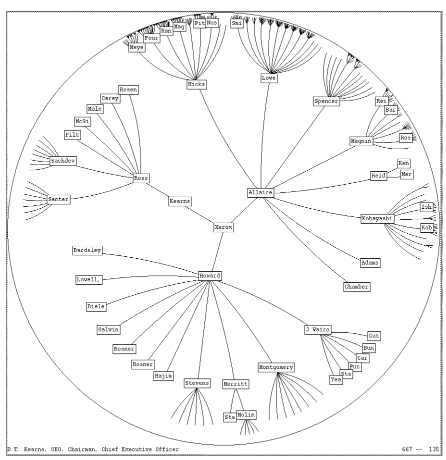


Figure 2.1. A distortion based focus+context visualization approach applied on graphs [7].

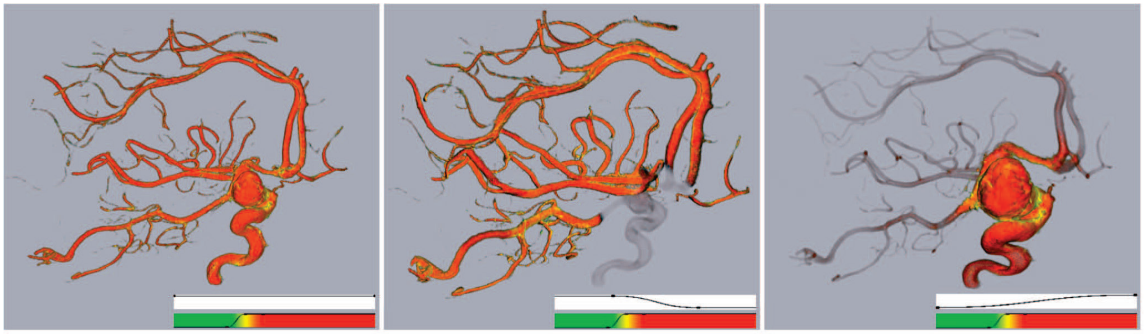


Figure 2.2 Automatic selection of distortion based on transfer function [9].

Another way of improving the saliency of the focus region is manual or automatic removal of unimportant regions. For multivariate datasets, SimVis framework [10-13] provides a multi-view interface in which the users can ‘brush’ (or select) values, say on a scatterplot or histogram, to emphasize or attenuate certain features of a dataset while a corresponding linked-view shows a 3D overview of the resulting visualization. For automatic or semi-automatic techniques in medical visualization, even though they are not usually considered focus+context techniques, transfer functions provide a similar effect. Numerous transfer functions using different information domains such as voxel values, feature size or textures have been proposed [14-18]. Shortcomings for transfer functions are the tedious editing that might be necessary to achieve desired results and the fact that the effects are global. This global application of a transfer function may result in loss of information in context regions. To avoid this, Cheng-Kai et al. [19] proposed a sketch-based interface where the user can guide the segmentation process by providing rough sketches of the regions. After the automatic segmentation, different transfer functions are applied to corresponding regions. This approach is dependent on the segmentation algorithm used, thus the performance might vary with the application, and the dataset used. Viola et al. proposed an ‘importance’ driven feature enhancement scheme [20] that reduces occlusion for the regions that are deemed important, but the approach requires a method that can effectively pre-segment the dataset. Similarly, Rautek et al. [21] have used high-level semantic rules to create focus+context effects (e.g. if distance to vessels is low then color style is yellow). In their follow-up work, this approach has been extended to handle user interactions such as defining a plane or moving the mouse cursor to a desired area to affect the rendering [22]. Multiple semantic rules can be combined by fuzzy logic to create illustrative effects. These methods are effective by introducing intuitive higher-level rules that can be understood easily by the novice users. However, effective low-level implementations of these rules are necessary for the success of the approach. Moreover, the rules are interactive but fixed to simple geometric constraints (e.g. distance to point, plane (Figure 2.3, [22])).

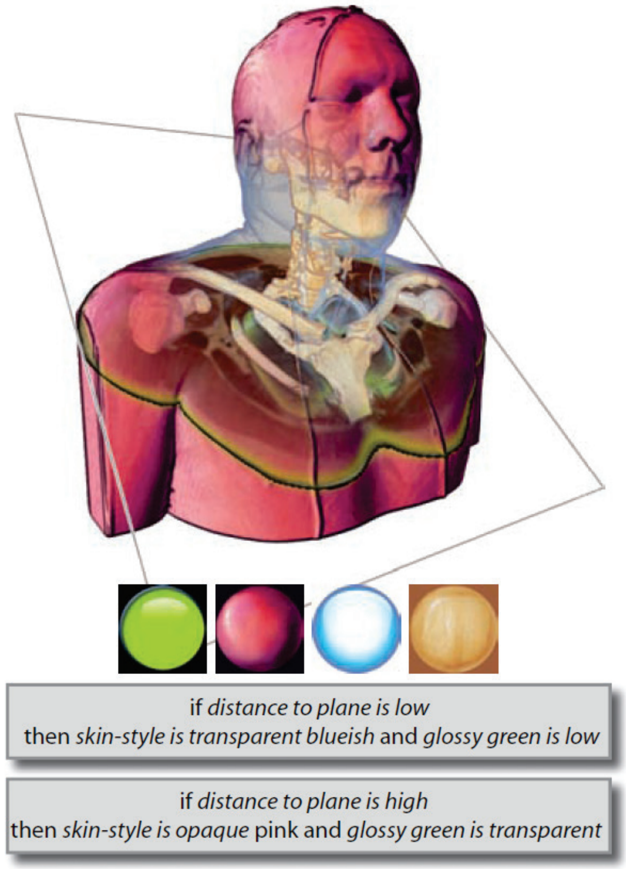


Figure 2.3. Focus+context visualization based on semantic rules using predefined styles [22].

Most of the focus+context applications in literature are designed to work with polygonal datasets (or isosurfaces), and usually assume that focus and context regions are already defined (e.g. with segmentation). In most medical applications this is not necessarily available, therefore interactive selection of focus and context regions is necessary.

2.1.2. Magic Lens Visualization

Since automatically deciding which parts of the dataset are important is challenging, giving the user control over the focus region location can be beneficial. This was the main motivation behind the Magic Lens approaches, which was proposed concurrently by Bier et al. [23] and Perlin and Fox [24] in 1993 (and is trademarked by the Xerox Corporation). Magic Lenses were first described as "interfaces to modify the visual appearance of objects, enhance data of interest or suppress distracting information" [23]. Similar to an optical lens, the user 'holds' a Magic Lens to a region in the dataset, which changes the rendering style of that selected region. This approach is closely related to the focus+context paradigm; the nuance is the control given to the user to change the location of a certain spatial region (the focus/lens region). Moreover, Magic Lenses can be used to create any desired effect in the lens region, ranging from spatial distortion (similar to an optical lens) to changing the rendering style (e.g. adding/removing data points or visual cues, changing lighting parameters etc.), making them a versatile visualization tool. Earlier applications include graphs, maps [25] and geospatial datasets [26].

The Magic Lens paradigm was first applied to 3D datasets by Viega et al. [27]. Two kinds of lenses were proposed: flat lenses and volumetric lenses. Flat lens was a direct extension of the 2D Magic Lens and optical lens metaphor: in a 3D scene all objects behind the flat 3D lens (called the lens frustum) is rendered with the lens effects. Volumetric lenses are sub-volumes defined in 3D space, which affect the rendering style inside them. One important difference between flat and volumetric lenses is that flat lenses have their far clipping planes defined by viewing frustum, while volumetric lenses have to have this defined explicitly. Viega et al. also remarks that defining relationships between multiple lenses can become cumbersome in volumetric lenses. This is partly caused by the hardware limitations at the time as well their rendering algorithm, which necessitated multiple rendering passes. Recently, Best et al. and Borst et al. [28-30]

provided methods to compose and render multiple volumetric lenses, with the user providing the logical expressions to determine the order in which the lenses should affect the rendering. Their methods are developed for polygonal 3D datasets and might be challenging to transfer to volume rendering because of performance considerations.

Different research efforts developed methods to apply Magic Lenses to volumetric datasets with real time performance. Borst et al. [29] provides a good comparison of several approaches in terms of number of rendering passes required, ability for multiple lenses, lens geometry allowed and performance (see Table 2-I). Ropinski and Hinrichs [31] introduced a rendering method that required multiple passes to display 3D Magic Lenses with convex shapes. Weiskopf et al. [32] introduced several methods for volumetric clipping, some of which can be applied to Magic Lens rendering. However, their volume probing based method required an additional volume to store lens information, which can introduce artifacts due to limited resolution of the volume and cause storage problems due to exponentially increasing sizes of the volume datasets. Depth based comparison methods perform better and can give sub-voxel accuracy. However, their proposed approaches are tailored to texture-based volume rendering, which produces lower quality images compared to raycasting. Layered depth images (LDI) have been applied for volumetric tests for polygonal objects [33], but creation of the LDI requires a preprocessing step and may be prone to aliasing artifacts if the LDI does not have high enough resolution. Moreover, this approach requires a separate volume texture to store the LDI, which may grow rapidly when resolution is increased to enable high-quality selections. The Magic Volume Lens [34] approach introduces spatial distortion inspired by optical lenses to volume raycasting (Figure 2.4). As discussed in focus+context visualization, spatial distortion can be undesirable for medical applications.

Table 2-I. Comparison of various Magic Lens rendering techniques [29] with our results.

	Multiple Lenses?	Composable Lenses?	Passes for one 3D Lens	Passes for n-3D Lenses	3D Lens Geometry	Lens Rendering Performance
3D Magic Lenses [27]	No	No	7	-	Hexahedron	Not reported
Looking Glass [35]	No	No	7	-	Hexahedron	Not reported
Depth-Peeled Lens [31]	No	No	3	-	Convex Polyhedron	One Lens at 40% performance
Magic Volume Lens [34]	No	No	-	-	Arbitrary 2d shapes on image plane	Small added cost for lens in ray-based volumetric renderings
Shader Depth Peeling [36]	Yes	No	2	n+1 (no composition)	Boxes (and Boolean Combination)	One lens at 92% performance
Composable 3D Lenses [28, 29]	Yes	No	2	n+m+1, (m intersection regions	CSG with Boolean Combinations	One lens best case 80%
Octreemizer Lenses [37]	Yes	Yes	2	Unbound	Convex polyhedra	Various results depending on intersection
Our Approach	***Yes***	***Yes***	***1***	***1***	***Convex polyhedra***	***Small added cost for preprocessing (usually negligible)***

Figure 2.4. Spatial distortion with raycasting inspired by optical lenses using the Magic Volume Lens [34].

Some focus+context visualization systems present conceptual similarities to the Magic Lens interface. The ClearView system is a focus+context hotspot visualization approach that preserves contextual information in the focus region. The importance of the features is calculated by several metrics such as normal, curvature, distance or view-distance. By keeping the contextual visual cues, occlusion problems are mitigated while keeping parts of the valuable visible information (Figure 2.5(a), [38]). Two possible methods are implemented: surface and volume based. Surface based method requires isosurface values to determine the focus and context layers. Therefore, this works best when the dataset is pre-segmented or contains clearly distinguishable layers. For the volume rendering approach, the shortcoming is that the focus region is defined on the image space based on a distance to a point on the surface; therefore, only a spherical region is possible for the focus sub-volume. Svakhine et al. uses illustrative enhancements and non-photorealistic rendering methods such as boundary and silhouette enhancement, tone shading and illumination effects to create illustrative focus+context effects (Figure 2.5(b), ([39])). Similar to ClearView, this approach only allows spherical sub-volumes to be defined as focus regions, the locations of which are selected in the image space.

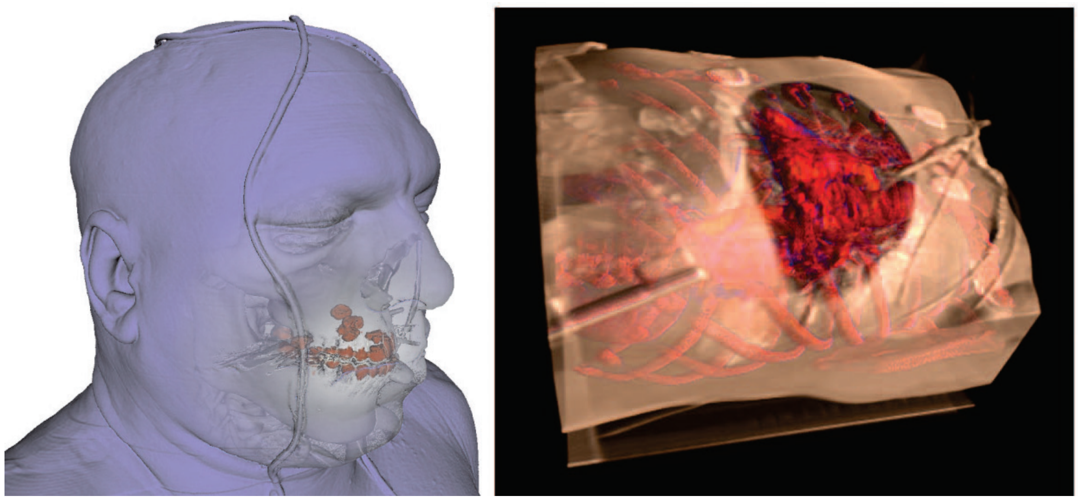


Figure 2.5. Examples of ClearView's context-preserving [38] and Svakhine et al.'s illustrative rendering [39].

2.1.3. Other Medical Volume Visualization and Manipulation Methods

Focus+context and Magic Lens visualizations are just a subset of many medical visualization techniques that have been proposed. In this section, the most relevant of these methods to our work will be discussed. This list is not exhaustive as there are a substantial number of research efforts for medical volume visualization, but is rather meant to provide an overview of the progress and state-of-the-art.

One of the greatest inspirations for medical visualizations is illustrations. Having a history going back to Leonardo da Vinci, medical illustrations have been used for centuries to teach and understand anatomy. The biggest problem with illustration is the need for the illustrator: a high quality illustration requires expertise and devoted time. To overcome this obstacle, researchers developed interactive methods to automate the illustration process.

One important tool an illustrator has to convey information is the drawing method selected. Similarly, different rendering methods were proposed to enhance certain features of volume datasets. Rheingans and Ebert [40] introduced non-photorealistic rendering effects such as boundary enhancement, silhouettes, tone shading, sketch lines or distance color blending. Svakhine et al. also used NPR techniques for both focus+context effects [39] and results that mimic traditional medical illustrations [41]. Bruckner and Groller [42] used halos in volume rendering to enhance depth perception. Lu et al. [43] applied stippling techniques to volume rendering to enhance features with high gradient to create more expressive and informative images. van Pelt et al. [44] used GPU-based particles to create effects that resemble traditional line drawings. Chen et al. [45] proposed a method that aims to create an illustrative visualization of muscle fibers from diffusion tensor images.

Another illustration technique that has been inspiring for visualization is spatial displacement and deformation. Techniques such as cut-aways, peel aways and exploded views have been applied to volume visualization. Correa et al. introduced multiple illustrative deformation tools such as peelers, retractors and pliers to reveal otherwise occluded anatomical features [46]. These operators resemble both illustrative deformations and surgical tools, resulting in intuitive visualization effects. One shortcoming of this approach is that features can only defined by isosurfaces or pre-segmented datasets, which can be inaccurate or time consuming to calculate. Bruckner and Groller [47] proposed exploded views to reveal hidden structures inside the dataset, which requires pre-segmented (as background and selection) datasets and allows basic geometric shapes to define the exploded view parts. The explosion is controlled by force functions that try to push the background (overview) objects away from the selection (detail) objects and try to keep the selection object un-occluded. Same authors previously introduced VolumeShop [42], which is an interactive visualization system that allows cut-outs, call-outs, ghosting effects and annotations. The user can 'paint' the selection sub-volumes by a click-and-select interface, where a spherical Gaussian brush is drawn into the selection volume. This approach is extended by Burger et al. [48] with an efficient GPU implementation that speeds up the temporally expensive process of writing to a volume texture. This volume texture is used to contain color information (i.e. volume coloring) or to remove structures from the volume rendering (i.e. volume editing). Even though impressive rendering performance is achieved, the technique has some drawbacks. Firstly, the brush used is limited to a spherical shape and high performance is predicated upon a small brush size. Secondly, the coloring method works on isosurfaces, which contain less information than the volume dataset itself. Third, the selection resolution is limited by the volume dataset resolution, which might lead to visual artifacts. The authors propose higher resolution selection volumes to increase the resolution in sub-volumes; however, this is only a temporary workaround and would not be feasible if the selection volume is closer to the size of a high-resolution volume dataset.

2.2. Visualization in Current Clinical Practice

Aside from the research efforts that were described in the previous sections, medical visualization approaches have already found their way into clinical applications. From image acquisition and diagnostic applications to surgical planning and guidance systems, numerous systems are being used in a variety of medical settings. However, the visualization approach used for most of these applications has not dramatically changed. The de facto standard for medical visualization is still two-dimensional: orthogonal sliced based views (Figure 2.6 [49]). Usually, the slices are located in fixed and known axes (namely, sagittal, coronal and axial) to ensure consistency. These visualizations can be supplemented with an additional 3D view to give the user contextual information about where the slices are located. In image guided surgery systems, the locations of these slices can be controlled by tracked surgical tools: i.e. the surgeon sees the slices taken at the precise location determined by a stylus or a surgical tool.

Figure 2.6. Orthogonal slice-based visualization of BrainLab surgical navigation software [49].

Intuitively, this can be seen as not taking full advantage of the 3D information: when the user sees three 2D slices on the screen, he is unaware of the rest of the dataset. The user can construct a mental 3D model by interacting and changing the location of these slices, which is a tough cognitive task prone to errors [50, 51] and can be more challenging to certain groups of people [51-53]. However, this seemingly simplistic visualization approach avoids some pitfalls of volume visualization. Occlusion is avoided because the datasets are shown on the screen in a spatially distinct manner. The size/depth ambiguities caused by perspective projection is also not present. Nevertheless, seeing and understanding 3D information from 2D cross-sections is a challenging task and requires extensive training to be performed effectively. Moreover, the mental registration process between pre-operative datasets and the patient becomes more difficult when slice-based visualizations are used. A volume visualization approach that alleviates the aforementioned challenges can improve the success of surgical interventions.

2.3. Human-Computer Interaction in Medicine

The Magic Lens is an inherently interactive paradigm, and like all interactive applications can benefit from intuitive and effective interaction schemes. The human-computer interaction (HCI) community has provided insightful research to analyze many interaction tasks in the past few decades. In general, HCI research requires thorough analysis of cognitive processes necessary for performing complex tasks, supplemented by empirical studies. Even though medical applications and volume visualization have been subjects of studies in this context (most relevant of which will be summarized in this section), it is hard to say that our understanding about cognitive processes necessary for complex medical tasks is complete. Specifically, the mental registration of real and virtual spaces, which is an important component of image guided surgery, is an important component of this dissertation that requires further research.

The first group of studies analyzes the problems associated with displaying 3D information on 2D screens. Teyseyre and Campo [54] provide an overview of 3D software visualization, concluding usability (i.e. emphasis on human factors), collaboration, integration (i.e. moving research projects into deployed systems) and display technologies to be the areas where improvements are most necessary. There have been studies comparing 2D and 3D visualizations, but these were usually domain and task specific (e.g. for air control [55] or telerobotic positioning [56]), therefore the results might not necessarily translate to medical tasks. Tory et al. [57-59] compared the effectiveness of 2D, 3D and combined 2D/3D visualization methods, concluding a combined method (ExoVis, Figure 2.7(a), [57]) outperform strict 2D and 3D displays for precise orientation and positioning tasks. Velez et al. [60] find that spatial ability may have an impact on performance of an individual's understanding of a 3D visualization. Keehner et al. [61] and Khooshabeh and Hegarty [62] have similar conclusions for the cognitive task of inferring cross-sections from 3D visualizations. It should be noted that this (although being conceptually similar) is the exact opposite problem of the general medical practice of inferring 3D structure from multiple views, a fact acknowledged by the authors. Another conclusion of these studies is that interactivity might not always be useful for some people with low spatial ability, which is an important reminder about importance of human factors when developing visualization methods. The authors hypothesize that additional interactivity might not be useful when it reduces the information conveyed in the visualization, or can make understanding explicit visualizations cognitively more costly than internal visualizations (e.g. mental registration, inferring cross-sections) in some cases. In other words, an interactive system is only useful when the user employs the interactivity to extract more information from the visualization, which may not be the case if the interaction method is poorly designed or cognitively challenging. These facts were instrumental in designing the visualization approaches presented in this work, and motivated the use of a natural user interface approach.

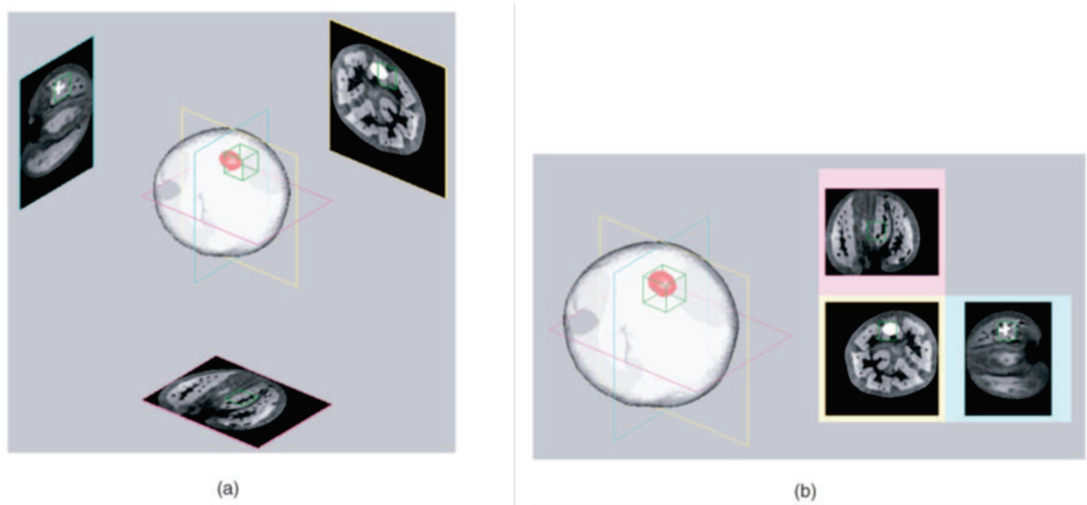


Figure 2.7. (a) ExoVis and (b) Orientation icon visualizations [57].

From the hardware perspective, surgical applications might present additional challenges (such as sterilization) which complicate the use of traditional input devices such as keyboards and mice. Yaniv and Cleary [63] present an overview of various interaction and display technologies used in image-guided surgery and conclude that the majority of the systems use standard computer monitors for display and four-quadrant (three orthogonal slices and 3D overview) based views for visualization. For interaction, tracking is an important technology. Tracking systems find the position and orientations of tools and anatomical structures in real-time, allowing quick feedback to user's actions to be shown in visualization. The dominating tracking technologies used are optical and electromagnetic tracking (examples are shown in Figure 2.8) because of their flexibility to variety of applications. Both of these types of systems are in general costly, and have a number of advantages and disadvantages. Optical systems use multiple cameras to triangulate positions of markers using pre-calibrated known locations of the cameras. For passive marker systems, markers are cheap, disposable and easy to sterilize. The biggest drawback is the line-of-sight requirement, which might limit the range of actions of the surgeon and makes them inapplicable to minimally invasive surgeries with flexible endoscopes (e.g. colonoscopy). Electromagnetic systems use a transmitter that emits known electromagnetic patterns, and receiver(s) to calculate the field strength to calculate current sensor location and orientation with respect to the transmitter, but might be prone to interference.

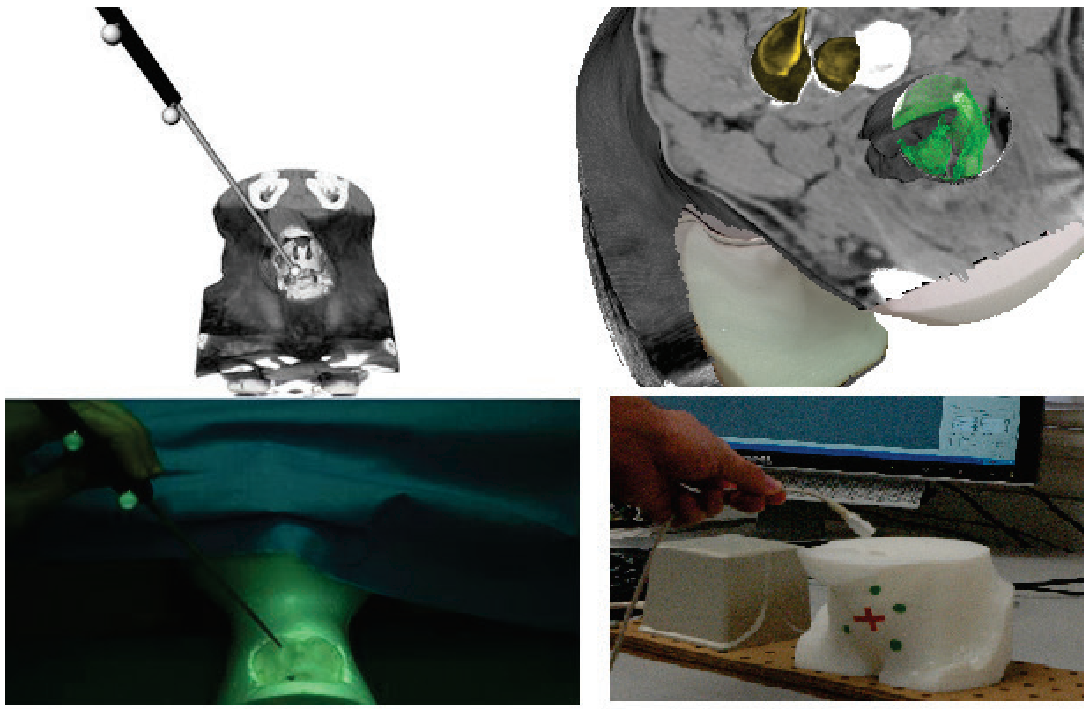


Figure 2.8. Example uses of optical and electromagnetic tracking.

2.3.1. Gesture-Based Interaction

HCI researchers aim to design interaction systems that does not bother the user or disrupt the current workflow. This idea led to an increasing effort on designing 'natural' user interfaces (NUI). These interfaces are aimed to perform interactions that are similar to everyday actions the users normally perform. Examples of these include using gestures or voice commands to interact with systems. The recent introduction of Microsoft Kinect with its affordable price and widespread availability has sparked a surge in the interest on such systems with a variety of application areas [64]. Medical and especially surgical visualization is a suitable application domain for gesture-based interaction systems. As mentioned earlier, traditional interfaces such as the mouse or trackers might have sterilization problems when used in an operating room. Using these systems might have other implications that can disrupt the surgical workflow, some of which have been recently described by Johnson et al. [65] in the context of interventional radiology. For instance, the users might have to direct other users to interact with the system due to sterilization and asepsis concerns, resulting in increase in task completion time. Another possible effect of using traditional interaction interfaces such as the mouse can be the loss of attention and focus, because the surgeon will most likely have to move to be able to reach the mouse. Therefore, touchless interaction methods can be useful in performing interactions in the operating room, eliminating these problems.

Gestures have previously been used in HCI research. Hauptmann [66] performed experiments with users performing actions, with analysis showing that people prefer to use both gestures and speech for the graphics interaction and intuitively use multiple hands and multiple fingers in all three dimensions. Bowman et al. [67] present a detailed overview of 3D interaction techniques, including gesture based interfaces. Few recent research efforts used depth cameras to extract user hand locations to enable interaction [68, 69], including systems designed for medical applications such as Gestix [70]. These systems focused more on the extraction of user hand locations rather than analyzing the human factor considerations. As concluded by Johnson et al. [65], design and evaluation of intuitive touchless interfaces may be very beneficial for various surgical visualization tasks, which is one of the contributions of this dissertation.

2.4. Summary

This section gave an overview about the problems associated with medical visualization. Various previous proposed solutions to the problems were presented, along with their shortcomings. This dissertation introduces methods that are inspired by some of these concepts (especially focus+context and Magic Lens visualization). We believe by applying these concepts to surgical applications, supplemented by novel visualization and interaction techniques implemented in real-time by the aid of GPU programming, these problems can be alleviated.

Chapter 3 - METHODS

3.1. Visualization Approach

The proposed visualization approach fits into focus+context paradigm. The main motivation for selection of this paradigm is the complexity of volumetric medical datasets and its similarity to way human beings see and interpret information. The human retina includes a region called *fovea,* which includes a high density of photoreceptors enabling more information to be captured. This anatomical structure affects the way humans see the world, the brain fixates the eyes on the regions it deems important [71]. For complex scenes, we tend to scan through the salient features to collect information about what we see. Similarly, focus+context visualization assigns more visual information to important (focus) parts of the dataset, while displaying the surrounding areas in less detail to provide context. While this analogy makes focus+context visualization intuitive to understand, some inherent properties of medical datasets and volume visualization, as discussed in the introduction, complicate its application to the medical domain. For effective application of the focus+context paradigm to medical datasets, there are three main problems to overcome:

I. **Importance.** It is a difficult task to define important regions of an unknown dataset automatically.

II. **Occlusion.** Since medical datasets are acquired in a different manner than our visual system (i.e. can penetrate opaque tissue), presenting this information effectively requires solving occlusion and depth ambiguity problems.

III. **Projection.** Since 2D monitors are ubiquitous, it is necessary to display 3D information on a 2D display. This projection results in loss of information.

From a technical standpoint, the following issues have to be considered:

I. **Region Definition.** Focus and context regions have to be defined (and stored) to be rendered differently.

II. **Performance.** For interactivity, rendering these regions differently should be done in real time.

III. **Interaction.** An intuitive interface should be provided to enable effective interaction, which has to be implemented for the operating room environment.

It should be noted that these problems are not mutually exclusive. For instance, the problems created because of projection include occlusion, or interactivity can be used to solve the importance problem.

3.2. Magic Lens Visualization

Magic Lens visualization gives a versatile tool to address the problems listed above. This paradigm has a broad definition and has been applied to various problems. In this dissertation, we propose to use a volumetric Magic Lens with no spatial distortion.

A volumetric Magic Lens can be defined as a sub-volume of known shape whose location and orientation is controlled by the user. In comparison to a flat lens, which affects the visible properties of all objects behind it, a volumetric lens can be used as an interface to enhance information inside a user specified sub-volume. A comparison of these kinds of lenses is shown in 2D in Figure 3.1. This approach is suitable to surgical applications where the interaction is usually done on the surface (i.e. patient's skin). Moreover, the known size and shape of the volumetric lens can be helpful to get depth and relative size information by interactively changing the lens location. Similarly, spatial distortion was avoided because in surgical applications, relative size and shape information may be crucial and distortion can result in incorrect diagnosis or decisions. Spatial distortion can also make the mental registration process more difficult.

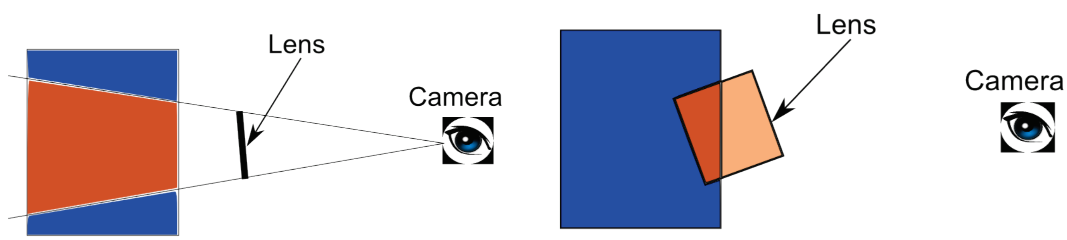


Figure 3.1. The differences between a flat lens (a) and volumetric lens (b) (illustrated in 2D) (note that the camera location does not play a role in the definition of the lens region for a volumetric lens.)

The use of volumetric lens as an interaction tool can alleviate some of aforementioned problems. The user can interactively decide the important parts of the dataset in two ways:

- By exploring the dataset (where the lens region is considered the focus and the rest of the dataset is the context, (Figure 3.3),
- By 'volume painting', where the user can mark irregular and arbitrary shaped focus regions on the dataset. The dataset is initialized as the context region

completely, the user adds the focus region by interactively adding sub-volumes by using the Magic Lens as a volumetric brush (Figure 3.6)

This approach is also helpful in avoiding occlusion problems. By assigning more transparent rendering styles to occluding regions, the structures that occlude the important parts of the datasets can be selectively removed. Using the Magic Lens approach creates more desirable results than global application of transparency, which can complicate the mental registration process or apply transparency to target structures that might need to stay opaque (Figure 3.2). By using the Magic Lens to explore the dataset, the users can create a mental 3D model of the volumetric dataset (Figure 3.3).

The problems that arise from projection of the 3D dataset to 2D are also helped by volumetric Magic Lens visualization. The most important of these problems is the visibility ordering: when a target structure that is behind an opaque surface is displayed, our brains have a difficulty interpreting the correct order of visibility (i.e. the object appears as if floating in front of the surface) [72]. The cues provided by interactively changing the Magic Lens position helps establish the correct relationships. This is beneficial especially in interactions in the real world (e.g. using a tracker), where the psychomotor feedback can aid spatial perception.

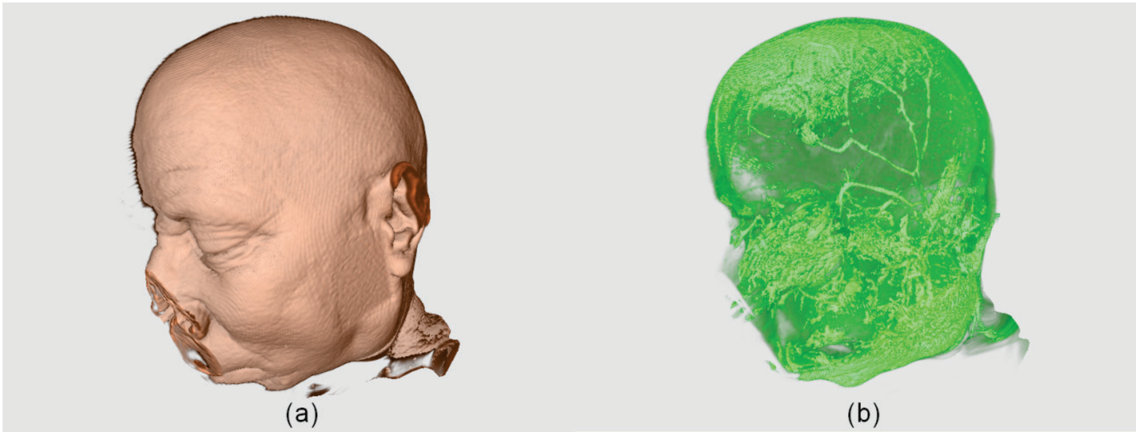


Figure 3.2. Global application of transfer functions, (a) displays skin and soft tissue, while (b) shows vascular structures. Note the loss of context in (b) because of the global application of transparency.

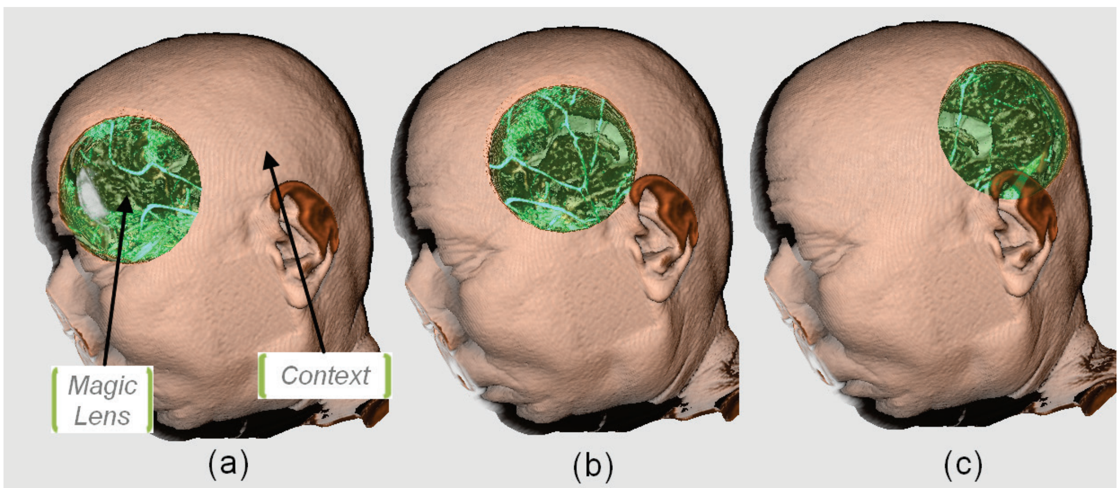


Figure 3.3. Volume exploration with Magic Lens. The lens region is moved the left-to-right (a-c) showing the vascular structures. The rest of the dataset is shown to provide context.

3.3. Real-time Magic Lens Rendering

The previous section conceptually explained advantages of using volumetric lenses. To implement this method, an efficient rendering algorithm is required to enable the benefits provided by the interaction. The most common method used in volume rendering is ray casting which produces visually pleasing results. Briefly, ray casting calculates the final color and opacity value of a pixel by shooting rays from the virtual camera position and accumulates samples along the ray direction. The real-time implementation of this computationally expensive operation has been possible with the developments of modern GPUs. Since rays can be calculated independently from each other, multi-core architecture of GPUs can be efficiently used. However, using a naïve ray casting method requires the sampling to start from the image plane, which means the samples between the image plane and the object, as well as the samples between the backfaces of the object and the far clipping plane will be empty space. To avoid this, researchers have used bounding volumes [73] and polygonal objects to act as proxies to dictate ray start and end positions [74].

In order to render volumetric Magic Lenses, we need to determine if a given sample is inside or outside the lens region. Previous research efforts either used separate passes to do this (Borst et al. provides an overview[29]), or checked each sample during the volume rendering process [30]. Especially for volume raycasting using GPU architectures, this approach presents problems because it requires the shader to perform different actions for each sample based on the result of the in-out checks. Since modern graphics processors are designed to perform SIMD (Single-Instruction-Multiple-Data) tasks, these kinds of branching actions are detrimental to raycasting performance. This is especially true in operations that require loops such as volume raycasting, and the performance gets worse as number of branches (for instance lenses or selection regions) increase. Our approach overcomes this problem by segmenting rays for each pixel on the GPU before performing the raycasting. This in effect means dividing the rays into three segments for each lens region: in front of the lens, inside the lens and behind the lens as illustrated in Figure 3.4. This way parallelism of the computations is improved since for each fragment the same set of operations will be carried out.

To perform this task, we use depth images of the lens shape that is rendered in a pre-rendering step. The approach is inspired by polygon-assisted raycasting [74], where the depth values of polygonal objects created from volumetric datasets in a preprocessing step are used to define ray entry and exit positions for raycasting. Similarly, we can use a polygonal object to define the boundaries for the currently selected Magic Lens sub-volume by rendering the polygonal shape in a pre-rendering step. This step is computed in significantly smaller amount of time compared to raycasting because polygonal rendering is a faster operation (and in most practical cases Magic Lens shapes have simple geometries).

In addition to performance improvements, another advantage of this approach compared to the analytical approach (i.e. using functions for performing in/out tests, such as Joshi et al.'s approach [75]) is that more complex shapes can be defined easily by creating polygonal objects. Compared to volume texture lookup based Magic Lens rendering approaches the advantage of our approach is the accuracy around lens boundaries. Volume textures can have jagged-looking artifacts around lens boundaries depending on the resolution of the volume texture used (similar to volume clipping artifacts shown by Weiskopf et al. [32]). Using depth images, the Magic Lens rendering can be done with accuracy independent of the resolution of the volume dataset and these artifacts can be eliminated. Moreover, storing a separate volume texture to store the lens shape usually uses significantly larger space than a polygonal model. Finally, this approach is flexible enough to allow rendering schemes for arbitrarily shaped selection volumes defined by the user, technical details of which will be described in the following sections.

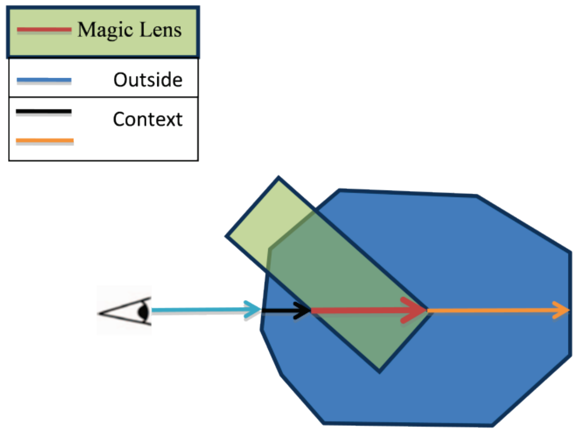


Figure 3.4. Lens rendering approach for volumetric lenses (illustrated in 2D).

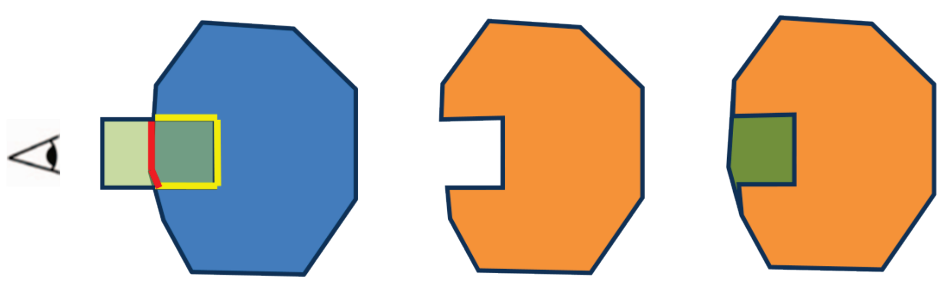
Our approach requires the construction of a polygonal mesh for each new volume. This has to be done once for each volume dataset, and the mesh can be extracted and stored prior to the visualization or during the initialization of the program using well known algorithms such as marching cubes [76]. The assumption we make in terms of lens shape is that the lens volumes will be convex, which is reasonable given that most research efforts constrain lenses to be spherical, while our approach allows any convex shape that can be described as a polygonal object.

3.4. Volume Editing/Painting

Using a volumetric Magic Lens is a versatile tool for data exploration; however, the user is limited to a predefined lens shape. In other words, the lens has no 'memory'; it only changes the display properties of the currently selected region. In many applications, the interesting parts of the dataset have irregularly shaped boundaries. Conceptually, this is similar to a segmentation problem. An important problem while applying segmentation algorithms to visualization tasks is that they are global, that is, they are applied to the whole dataset. In many medical applications, the contextual information is crucial in defining whether a region is interesting or not. For instance, the proximity of a vessel to a target structure might make it more important than another vessel, even though both structures share similar intensity values in a medical dataset. There are approaches to use additional cues such as proximity for segmentation; however, these are usually task specific. We believe an interactive system to visualize volume datasets that allows the user to select the interesting parts of the dataset while keeping the rest for contextual information would be beneficial and provide a valuable extension to the focus+context paradigm.

Conceptually, our approach extends the Magic Lens to act as a volumetric brush. While exploring the dataset with the lens, the user can opt to start marking the areas he deems interesting, or use the lens to remove structures that are irrelevant or obstruct the view of target structure. From an implementation perspective, the most obvious solution to do is to use an additional volume texture to store if (and how) a voxel should be displayed. However, as it was the case in the rendering of the Magic Lenses in the previous section, this has several drawbacks that prevent an efficient real-time implementation. Volume textures are slow to write into, require more storage (or memory) space, and have limited resolution. Increasing the resolution increases the speed and storage problems exponentially. Therefore, our approach of using polygonal meshes to assist in volume rendering can be suitable alternative for this problem.

The biggest obstacle to overcome if we want to use polygonal meshes to store the information about regions in 3D is the difficulty of changing the topological information in real-time. Our solution to this is storing and changing vertex information in the GPU. The approach is, if the user decides to use the Magic Lens as a brush, all the vertices of the proxy mesh inside the lens region are pushed back to the backface of the lens. Since this polygonal mesh defines entry and exit positions for volume rendering, the region that is occupied by the lens will be skipped when the rendering is performed, even if the volumetric lens is moved from that location. The most obvious consequence of this is using the lens as a volumetric eraser: this approach effectively removes that region from the volume rendering. A useful extension of the idea is using two proxy meshes: one mesh can be kept unchanged while the second one is modified in the manner described above. This way, when the depth values from these meshes are obtained in the preprocessing stage, we have two distinct surfaces between which a selection volume is defined (Figure 3.5).



(a) (b) (c)

Figure 3.5. Illustration of mesh deformation using depth images. Depth values of the Magic Lens from the viewpoint (highlighted in yellow) in (a) are used to move the vertices inside the lens (in red), resulting in the new proxy mesh in (b). By using two meshes together, a selection region (green) can be defined in (c).

The implementation of this approach requires the exploitation of multicore processing capabilities of the GPU. First, we format the vertex positions of the proxy meshes (which contain N vertices) and store them as an image of size $N^{1/2}xN^{1/2}$. In each frame where the volumetric brush is enabled, this 'image' is passed to a fragment shader, which checks the position of each vertex in the mesh to see if that particular vertex is inside the lens boundaries. If a vertex is inside the lens volume, its depth value gets overwritten by the depth of the backface of the current lens. Effectively, this pushes each vertex to the back of the lens along the viewing direction without changing the topology of the mesh.

The advantages of this approach are several: first, the selection is done using the polygonal mesh, which means the selection is completely independent from the resolution of the volumetric dataset. This gives us smooth selection boundaries with sub-voxel accuracy. Secondly, polygonal datasets in general are smaller in size compared to volumetric datasets, for instance, the mesh used in Figure 3.6 is 8.6 MBs while the volumetric dataset used to create it is 172 MBs. This means we can store volume selection information using significantly smaller storage space. This is particularly useful since an effective editing operation requires functionality to undo actions, which would require saving multiple copies of the selection volume if the volume-based approach is used and would be infeasible.

There are some drawbacks to this approach. When a polygon is close to the boundary of the Magic Lens volume, some of its vertices can be inside the lens volume and some can be outside. This can cause jaggy looking artifacts along the selection boundary if a low resolution mesh is used. Secondly, each distinct selection requires a separate mesh to be stored and rendered in preprocessing. In our implementations, the first problem was alleviated with increasing the resolution of the mesh, and satisfactory results were achieved without sacrificing speed and still using considerably less space compared to storing an additional volume texture (comparative results will be introduced in Chapter 5). Third, the selection volumes are assumed to be convex and contiguous. This in practice means the selection can only be done on the surface of the dataset, which is not a significant problem in our application domain.

3.4.1. Multimodal Visualization

One of the difficult problems in visualization of medical datasets arises when multiple data sources are present, for instance CT and MRI scans of the same patient. For instance, a CT scan provides detailed information about osseous structures while the MRI captures soft tissue information better. There is a significant and growing amount of literature about finding the correspondences between different kinds of datasets (i.e. registration). However, even though the datasets are correctly registered, displaying these datasets is still a challenging problem. Displaying these datasets separately introduces another dataset for the physician to consider when performing the mental registration. The mechanisms behind mental registration of multiple data sources are not psychologically well understood, and having multiple frames of reference have been shown to be detrimental to performance [77]. The alternative of merging and displaying datasets together intensifies the problems already inherent in volume rendering and might impair the user's understanding of the datasets, especially due to increased occlusion and information overload.

Our approach of using the Magic Lens described until now focused on *how* to display user specified focus and context regions of a single dataset. For instance, the volume exploration or painting can be used to assign different transfer functions to the desired regions. The same framework can be extended to handle *what* should be displayed in the user specified regions. Using combinations of (possibly multiple) focus and context regions, the user can create an intuitive rendering by taking into account the positions of these regions relative to desired anatomical structures, while using the rest of the dataset to provide the context (Figure 3.6). In this figure, the region from the MRI dataset displays the blood vessels in the brain, while the skull, soft tissue and contrast enhanced vascular information are from a co-registered CT dataset of the same patient. The combined image contains information from all these sources shown in a cohesive view. One important advantage of our framework is this approach can be incorporated seamlessly: during the volume rendering different regions can be assigned with desired transfer functions and datasets, and can be rendered appropriately.

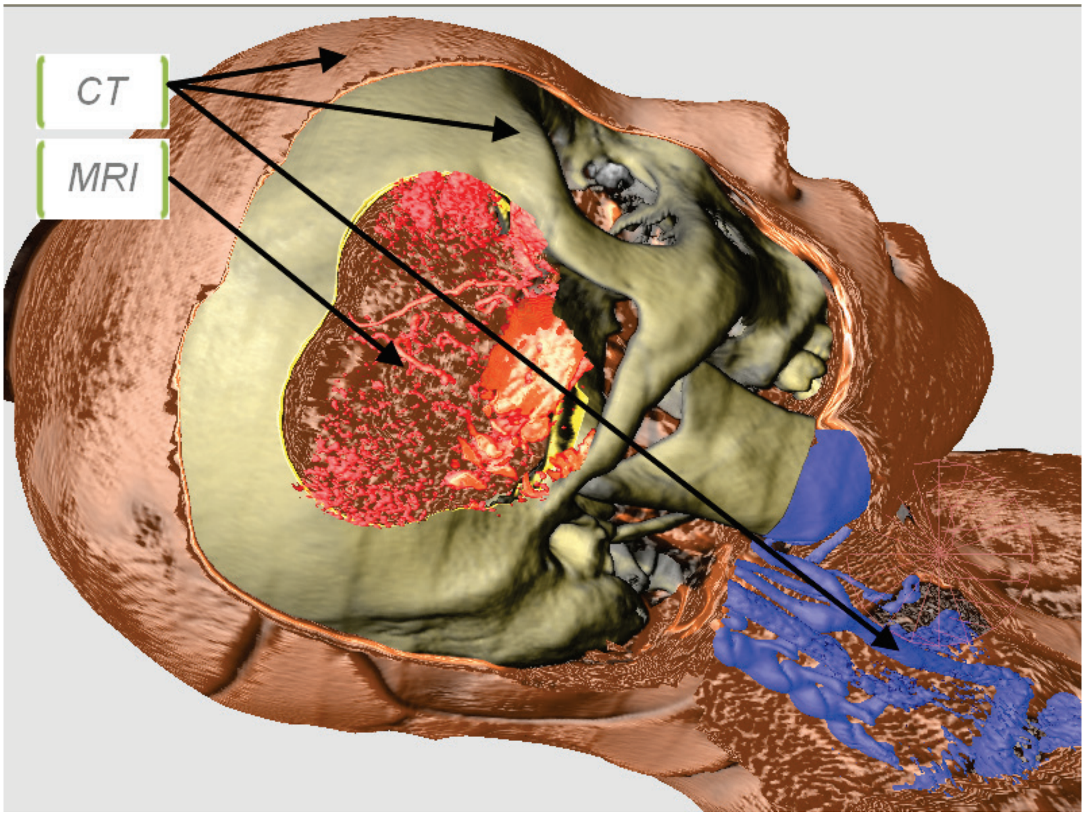


Figure 3.6. Combination of multiple modalities with volume painting.

Our modified raycasting approach for this application is performed as follows (Figure 3.7): the depth values for entry-exit positions for each region are calculated before raycasting and are used to calculate ray segment lengths. In the raycasting step, we perform the raycasting of regions in a pre-defined order. This means regions will have a predefined ordering in rendering, (e.g. region 1 can overwrite region 2, and so on). This avoids the branching behavior and performance decrease that would be caused if conditional statements were used to check each sample inside the rendering loop for using different rendering styles (e.g. if current sample is inside Region 1 render Dataset 1, if inside Region 2 render Dataset 3 etc.). Therefore, the only computational overhead introduced is the calculation of depth values for each region in the pre-rendering step (a fast operation because polygonal meshes are used), and calculation of ray lengths, which is done once for each fragment, instead of for each iteration of the raycasting loop. The dataset and transfer function used for each distinct selection region can be changed by the user during runtime. This method gives the user flexibility to select which dataset/rendering parameters will be shown in each region, while maintaining frame rates of regular raycasting because no branching behavior is employed in the shader.

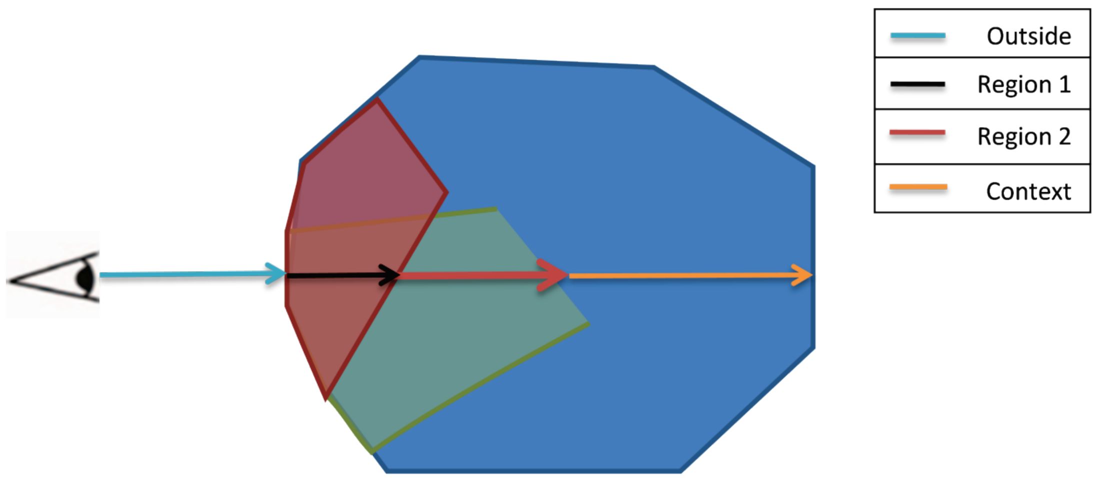


Figure 3.7. Conceptual diagram showing arbitrary shaped intersecting selection regions (illustrated in 2D). The rays are segmented using the depth images of proxy meshes and rendered in the pre-defined order (in this example Region 1 (red) is overwriting Region 2 (green).

3.5. Implementation and Evaluation of Interaction Methods

The methods presented so far assumed that the user will interact with the system to change the lens/brush location. There are many different methods to perform these kinds of interactions, which mainly fall into two categories: image space and object space. Image space refers to the user interacting with the created visualization, while object space interaction takes place in the real world, where the 3D location of a point selected by the user is used to control the visualization.

For image space interaction, the most commonly used interaction device is the mouse, which is inherently a 2D device. To determine the location and orientation (i.e. 6 DOF) of the volumetric lens, we need to deduce the remaining degrees of freedom. For the depth, the depth of the pixel pointed by the cursor in the rendered image can be used with the mouse scroll providing an offset to move the lens along its local z-axis. For the orientation, one possible method is using the surface normal of the selected point, which in effect makes the lens volume be always perpendicular (or parallel) to the object surface. A second possibility is using the viewing parameters, which can make the lens always perpendicular (or again, parallel) to the current viewing direction. These kinds of interactions can be easy to use outside the operating room because of the familiarity and ubiquity of the mouse interface, especially for surgical planning applications. However, there are practical limitations imposed by the operating room setting as discussed earlier, which complicates to use of the mouse. Object space interaction in the operating room has traditionally been done by tracking hardware: usually either optical or electromagnetic. There are some aforementioned problems associated with both of these technologies such as cost, accuracy and stability and need for sterilization.

The proposed gesture-based interaction methods can overcome these problems. By using a depth camera (Microsoft Kinect) to extract hand locations, we can perform the interactions using gestures. This eliminates the concerns about maintaining sterile/non-sterile boundaries because the camera unit can be located outside the sterile area, and no additional equipment is necessary for interactions. This interface can be applied to control the Magic Lens location, results of which are shown in Figure 3.8.

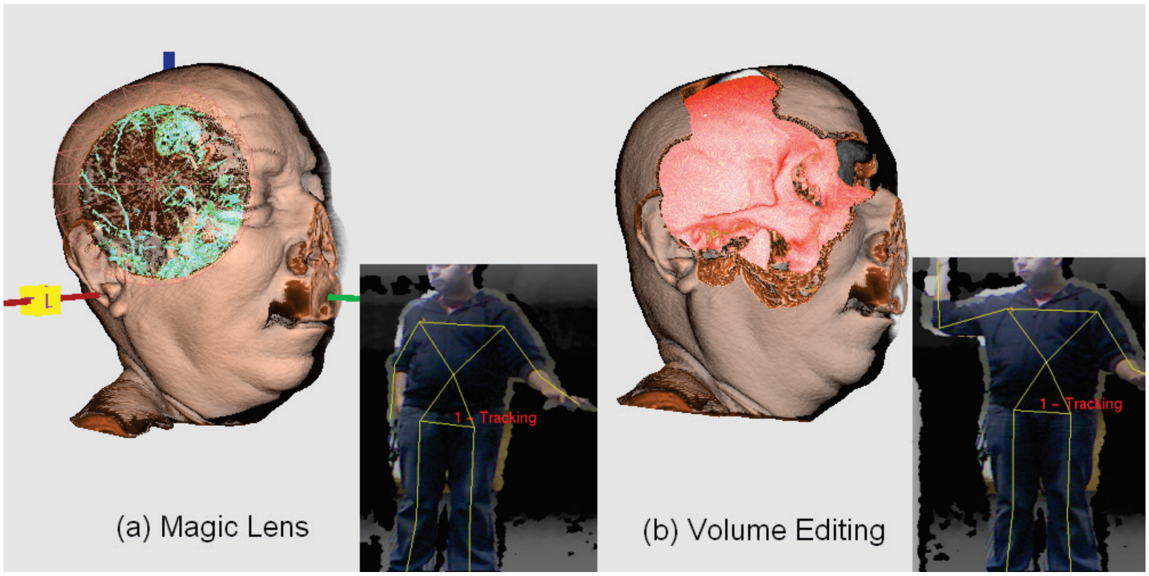


Figure 3.8. Examples of Magic Lens visualization and volume editing. The volumetric lens location in (a) and brush location used to create (b) are controlled by the user's right hand, while the editing mode is activated by raising left hand.

To evaluate the success and intuitiveness of the gesture-based interaction interface, user studies were conducted. Our aims in the user studies were twofold: first, we wanted to prove that gesture-based interfaces can perform comparably to traditional interfaces in basic interaction tasks. Secondly, we wanted to compare the effectiveness of the Magic Lens interface to slice-based visualizations and if 3D visualizations such as the Magic Lens can be used to explore volumetric datasets successfully.

Two studies were designed for these purposes. For both studies, the gesture-based interface was compared with the mouse, as it is the most widely used interface in surgical applications. Trackers were not included in the study since they generally fall into the same category (object space) of interaction, and the goal of gesture-based interface was eliminating additional equipment such as trackers. The first study aimed to compare the performance of the rotation tasks using the location of two hands versus using the mouse. The objective of the second experiment was finding the targets inside a volumetric dataset. Both of these tasks are widely used in surgical visualization applications. This section will explain the gesture-based interfaces used in these experiments, and give detailed information about the experiment process. The results of the experiments will be analyzed in Chapter 5.

3.5.1. Experiment I: Rotation

The first experiment compared the performance of a gesture-based interface (GBI) with that of the mouse in a volume rotation task. In the GBI, the two hand locations of the user are used to perform the rotation, as can be seen in Figure 3.9. This interaction method resembles the action of holding an imaginary object from its sides and rotating it in X- and Y-planes. With the mouse, the rotation is performed by clicking and dragging the mouse. The center of the rotation is denoted by the principal axes shown on the visualization (which also help users to maintain a frame of reference for rotations, which is shown to help user understanding of rotations [78]). The axis of rotation is perpendicular to the current rotation of the mouse movement. The users are given target rotations on the right side of the screen, and are told to match the interactive visualization on the left side to the target (Figure 3.10). After training for a limited amount of time with both interfaces, a fixed number of targets are shown to users successively and their performances were recorded both in accuracy and time. The Stanford Bunny [79] was used in both experiments as the volumetric dataset.

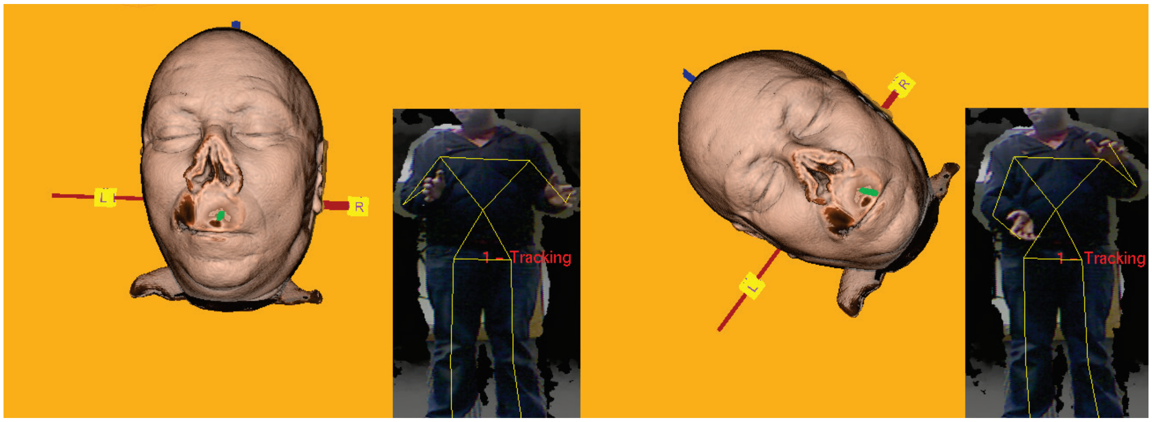


Figure 3.9. Rotation by hand locations. The three main axes of rotation are used as rotation references. The yellow cubes denoted by L and R in the visualization respectively correspond to left and right hand locations of the user, the correspondance to user's hand locations can be seen in the lower right sides of the volume renderings.

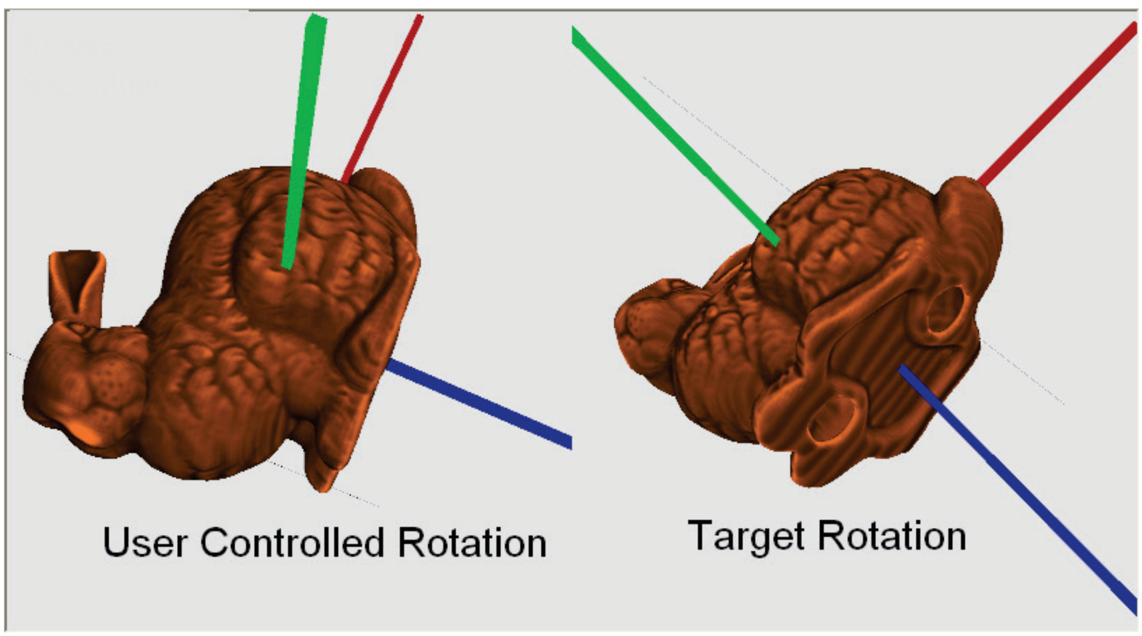


Figure 3.10. A sample screen for Experiment I. The users try to match the rotation on the left side of the screen to the target orientation seen on the right.

3.5.2. Experiment II: Target Localization

This experiment's objective was to compare the performance of the GBI with the mouse in a target localization task. Finding targets inside volumetric dataset is a crucial task used in many surgical visualization applications. Since the de facto standard for such visualizations is using a 2D slice whose location is controlled via a mouse, we compared our GBI with a mouse-controlled slice-based visualization. Artificially created targets were placed inside a volumetric dataset in randomized locations, and users were asked to locate these targets. For each experiment, a fixed number of distractors were also created, which were smaller than the targets. The users were asked to explore the volume and find the target inside the volume by judging the sizes of these artificially created structures. The first interface used in this experiment was using a slider control and a mouse, where the sliders range was set to the volume's z-axis length. The second interface used the location of the user's right hand with respect to the torso joint, and the height difference between these two locations was used to control the slice position. In both of these experiments, the left side of the screen showed an opaque volume rendering that did not reveal any of the targets, while the slice was shown in 2D on the right half of the screen for exploring the volume dataset (Figure 3.11). A placeholder for the slice location was shown on the left side to help the users understand the relative location of the slice with respected to the rest of the dataset.

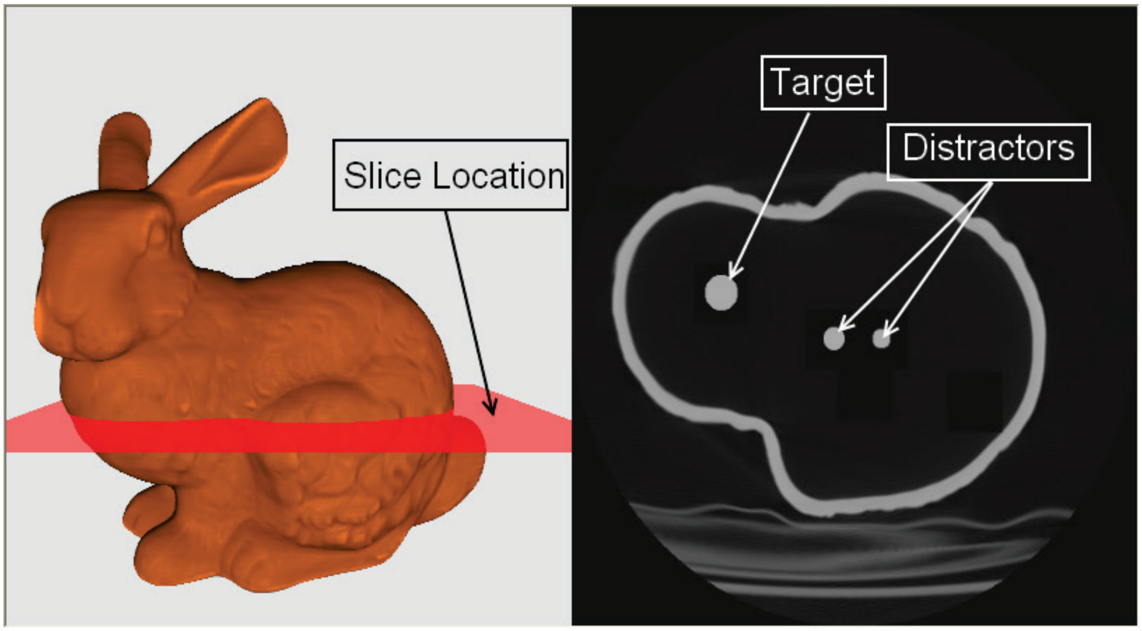


Figure 3.11. A sample screen for Experiment II (for slice-based visualizations).

These two slice-based visualizations were compared with a Magic Lens interface (Figure 3.12). In this interface, the Magic Lens was used to reveal the targets by making the object boundaries transparent and showing the target volumes located inside. Again, the users had to find targets that are larger than the distractors.

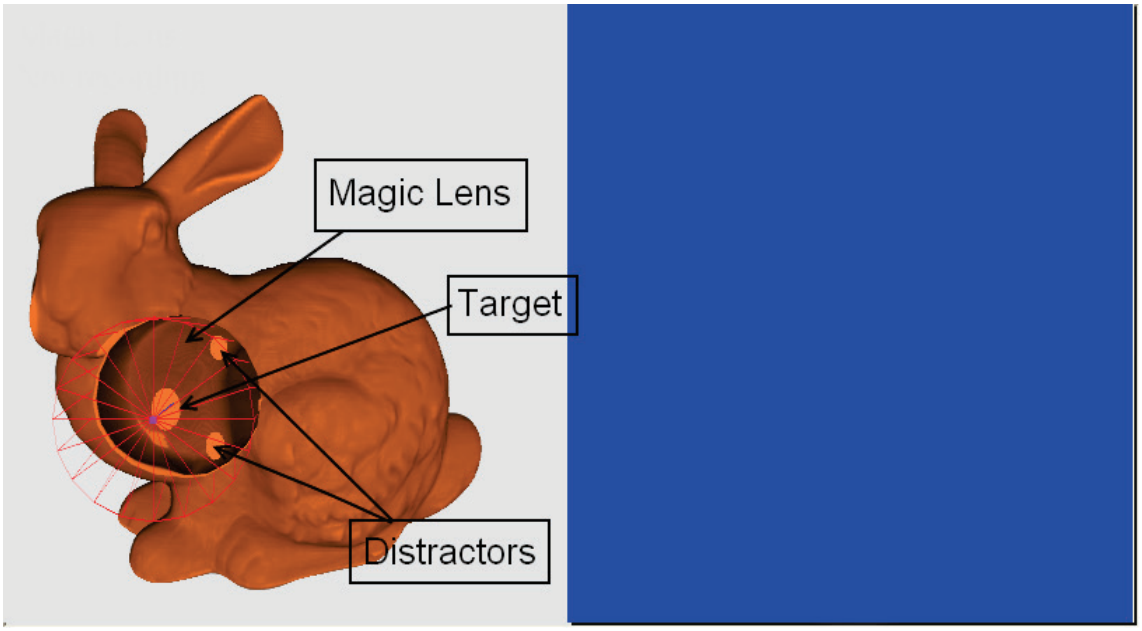


Figure 3.12. A sample screen for Experiment II (for Magic Lens visualization, note that right half of the screen was empty in this experiment because only a 3D visualization is used).

3.6. Summary

The methods presented in this chapter are aimed to improve the understanding of volumetric visualizations by interactive exploration and editing. Real-time rendering methods were introduced for Magic Lens visualization, which were designed to take advantage of the GPU to avoid any performance impact compared to traditional volume rendering. By using the lens volume as a volumetric brush, focus+context volume exploration was extended to handle volume editing tasks, using novel polygon-assisted volume selection methods. These techniques were applied to an interactive gesture-based interface. User studies were undertaken to evaluate if these interfaces can compare to traditional mouse interfaces. The performance results and the analysis of these user studies will be introduced in Chapter 5.

Chapter 4 - APPLICATION DOMAIN

The focus of this dissertation is the application of interactive volume visualization techniques to surgical applications. Visualization is an application-oriented research topic, therefore having an understanding of the overall picture where the proposed visualization methods will be used is important in determining the importance of this work. The next sections will provide a broad overview of computer aided surgery paradigm, and common components found in such systems. The pre-processing and registration steps are essential in the success of the overall application, but the exact nature of how this is done is not the focus of this dissertation. Our research focuses on how the available information provided by the pre-processing steps is presented to the user.

4.1. Computer Aided Surgery

As the name implies, any surgical intervention where computer technology is used to improve surgical outcome can be classified as computer aided surgery. Some recent technological developments were instrumental in the increased importance of CAS systems. The first is, as discussed in the introduction, is the widespread availability of medical imaging technologies that provide volumetric datasets. The second is the prevalence of minimally invasive surgery, where the surgeon use endoscopic cameras to perform surgical interventions with limited visual access to minimize unnecessary trauma to the patient in surrounding areas to reach the surgical target. This limited visual input and visual detachment from the intervention site increases the importance of volumetric datasets to help the surgeons understand anatomical structures that are not visible with endoscopic cameras. The third development that made CAS more common is the increasing use of robotic surgery. With the deployment of systems such as Da Vinci surgical robot [67, 80], computers are used both for the control of such systems and also for presenting visual information to the surgeon by combining intra-operative (e.g. endoscopic cameras, X-ray fluoroscopy) and pre-operative (e.g. CT, MRI images) imaging modalities. These systems are used in a multitude of applications, including cases where the doctor is performing the surgery remotely; where the visualization becomes the only link to the patient.

4.2. Image Guided Surgery

In most cases, CAS systems have the general workflow of acquiring and analyzing the data about the patient, finding the correspondence between these datasets and the patient during the surgery and finally presenting this information to the surgeon to improve the surgical outcome. Excellent surveys by Peters [81] and Perrin [82] can be referred to for detailed analysis of image guided surgery (IGS) systems. Furthermore, Peters [83] provides a comprehensive list of IGS systems applied to various surgical procedures. Visualization can be considered the final step in these systems, the success of which is predicated on the prior steps. The pre-processing (i.e. data acquisition and analysis) steps are significant because they prepare the datasets to be effectively used in the application. Segmentation is one of the most commonly used pre-processing approaches and many different ways to perform segmentation have been proposed. A detailed list is beyond the scope of this dissertation, interested readers can refer to [84, 85] for more information.

The visualization approach presented here does not explicitly require the datasets to be automatically (and correctly) segmented; on the contrary, our approach uses interaction to improve the segmentation of the datasets in local regions selected by the user. However, a successful segmentation approach can increase the effectiveness of our approach by providing users distinct regions that contain different kinds of information, which can then be fused together to produce an informative image.

The second step in image guided surgery is registration. Registration can be described as finding the relationship between datasets that are acquired or computed before the procedure and the patient during the procedure. The main reason why registration is required is technological: currently, imaging technologies are not fast, safe, cheap or portable enough to provide real-time information about internal body structures without disrupting the surgical workflow. Therefore, a very commonly used method for image guidance for surgeries is using a pre-operative dataset and finding the correspondence to the patient during the surgery. This is a very challenging problem and an active research topic [86-89]. Moreover, when multiple datasets of the same patient is present, finding correspondences between multiple medical image datasets is another registration problem, and might be crucial in the success of consequent visualization approaches. A successful visualization system aims to present the information from these multiple sources of information and help with the mental registration of the datasets to the patient.

Our methods can be used in various image guided surgery applications to combine information from multiple sources in a single view to improve the surgeon's understanding. One example application we have explored [90] is Medialization

Laryngoplasty [91], a surgical procedure that aims to correct vocal ford deformities by implanting a uniquely configured structural support in the thyroid cartilage. The implant shape and location is very critical, which makes the revision rate for this surgery as high as 24% even for experienced surgeons. Our choice of this procedure was motivated by the number and type of data modalities used in the decision making: namely volumetric CT data, pre- and intra-operative laryngoscopic video and patient-specific CFD simulation that shows the air flow necessary for phonation. In current medical practice, the surgeon has to consider all these sources of information and mentally combine them to make correct surgical decisions. By using lens-based data exploration and volume editing, this process can be improved since the spatial relationships between these available modalities can be understood better in a single view, and occlusion problems can be avoided by choosing appropriate transfer functions [90] or using volume editing to remove unimportant parts of the data. This example can be extended to various surgical procedures that use multiple medical datasets. For interaction, if trackers are used the registration of tracker space and virtual world becomes important, but as mentioned above, various techniques have been proposed to solve this problem (for instance, computer vision techniques have been proposed for laryngoplasty [92]). After the registration, the surgeon can use the lens to either explore or edit the dataset by pointing the tracker at the desired location on the patient, and the visualization is updated according to this interaction. The gesture-based interface can be used to skip this registration step and use the surgeon's joint locations to perform the registration. For instance, in our implementations we used the a point with a fixed offset in front of the torso joint as the origin in the virtual world, and used the user's shoulder width to normalize virtual space. This way, the physician can always interact with the space in front of him regardless of his current position.

4.3. Surgical Planning

Like most complex tasks, surgical interventions can benefit from planning and analysis done prior to the surgery. Computer based surgical planning systems are increasingly being used to improve surgical outcomes by offering methods to analyze the available data, and to preview and simulate different surgical scenarios that can arise during the surgery. The properties of such systems can vary based on the task on hand: for instance, for a tumor resection surgery the doctor might want to analyze the surrounding vascular structures to decide on the initial incision location and size. For implant placement surgeries, the effects of the implant location and size might be analyzed by computer based simulations (e.g. simulations of air flow for vocal ford correction surgeries [93, 94]), thereby giving the surgeon information about possible surgical outcomes for different scenarios. From a visualization system perspective, this approach can be considered very similar to image guided surgery, in that the goal is to improve surgical outcomes by displaying available information to the surgeon in an effective way. One fundamental difference can be the interaction methods: image guidance systems are aimed to be used during the surgery *on* the patient, therefore *data registration* (i.e. finding the correspondence between pre-operative datasets and the patient) as well as *mental registration* (i.e. the process of understanding these relationships) are important. For surgical planning systems, since the patient usually would not be present in the room, understanding and manipulating available information takes precedence over the mental registration process. This subtle conceptual difference should be taken into account while designing visualization approaches for surgical planning.

The methods presented in this dissertation are aimed to be flexible to be used in both surgical planning and navigation contexts, giving the surgeon consistent visual information across both applications which can improve the surgical outcome. Image-space based interaction techniques can be used in the office by the surgeons to explore and manipulate the datasets to improve their understanding. Different surgical scenarios can be tested with the volume editing tool as a preview to the surgical procedure. Since our methods can be used to save multiple copies of editing states, these can be stored to act as guidelines during different stages of the procedure. Another possibility is using these saved states as key-frames to create animations, which can be used as a surgical planning or teaching tool.

Chapter 5 - EXPERIMENTAL SETUP AND RESULTS

This chapter will present the results of implementation and evaluation of our methods. Since interactivity is a key part of our visualization approach, both the rendering performance achieved and the usability analysis of the proposed interaction techniques will be introduced.

5.1. Experimental Setup

The methods described in earlier chapters have been implemented using C++, MFC and CG GPU Programming. The system used to run the visualizations is a DELL Precision 690 Workstation with a Quad Core Intel Xeon 2.66MHz X5355 processor, 3.21 GB of usable RAM. The graphics card on the system is an NVidia Quadro FX 4600 with 768 MB of video memory. The operating system used was Windows XP, programming and compilation was done on Microsoft Visual Studio 2005/2010.

We mainly used two datasets for the results presented in this work: the first are CT Angiogram (CTA) and MRI images of the same patient. The resolution of the CT dataset was 512x512x345 voxels in x,y,z directions with 0.4297(mm), 0.4297(mm), 1.0(mm) spatial resolution respectively; while the MRI images used had 512 x 512 x 174 voxels with 0.4688(mm), 0.4688(mm), 1.0(mm) spatial resolution. As can be inferred from these values, the CT scan covered a larger area of the head (starting from below the neck and including the whole head), while the MRI data was available only for around the nose and the eyes. The second dataset (Cerebrix dataset from the OsiriX database [95]) had an MRI, CT and PET scans of the same patient, with 512 x 512 x 174 (MRI), 176 x 224 x 256 (CT), 336 x 336 x 81 (PET) voxels. These datasets were co-registered rigidly using Marching Cubes [96] to extract the bounding polygonal surfaces and performing ICP [97]. However, the exact nature of this registration is not important for a visualization approach, since the goal of using these datasets is showcasing possible uses of our approaches. The surfaces extracted are also used to define the entry and exit locations for raycasting, which is the only pre-rendering step necessary.

5.2. Rendering Performance

In this section, we will present the performance analysis of the visualization methods discussed in the previous sections. Performance of volume rendering with raycasting depends on the number of times the rays are sampled before termination (e.g. because a pixel becomes opaque or the ray exits the volume), which depends on a number of factors such as window size, fill rate of the window, sample spacing or transfer function used. For instance, selecting a large lens size with a transparent transfer function might result in a lower frame rate, but this is because more samples need to be processed rather than our rendering modifications. We believe that our modified raycasting approach should have the same performance compared to a ‘simple’ raycasting scheme. The difference in performance is due to the pre-rendering step necessary for calculating depth values for different selection regions and lens sub-volumes, and our assumption was these steps can be performed fast enough for real-time frame rates. The results presented in this section show that our assumptions hold true and real-time rendering rates can be achieved using our methods.

For analysis of Magic Lens frame rates, a fixed lens size and location was used. The same size and location was also used to perform volume editing operations, as can be seen in Figure 5.1. We wanted to demonstrate two results: the first is that Magic Lens rendering can be performed with negligible overhead compared to unmodified volume raycasting. The second result is related to the accuracy of our mesh deformation scheme: in general, our volume editing method performs with less visible artifacts when the resolution of the bounding mesh is higher. Even though rendering higher resolution meshes requires less computational complexity and memory storage than rendering higher resolution volumes, the performance impact can become noticeable if the mesh becomes extremely complex. In Figure 5.1, we wanted to highlight these possible artifacts and wanted to demonstrate these become less noticeable when the resolution of the mesh is increased. The corresponding frame rates and pre-rendering times can be seen in Figure 5.2 and Figure 5.3. These results show that the Magic Lens rendering can be performed with no performance impact to volume rendering (as the pre-processing takes about 0.5 msecs for lens rendering), as our modified raycasting avoids branching operations in raycasting. For volume editing, the performance can be impacted when the mesh resolution is increased, but the artifacts become less noticeable. As shown in Figure 5.1(b) and (c), the visual quality achieved in volume editing is satisfactory with 30218 and 59042 vertices, and frame rates of 96% and 92% respectively of traditional raycasting is achieved performing volume editing.

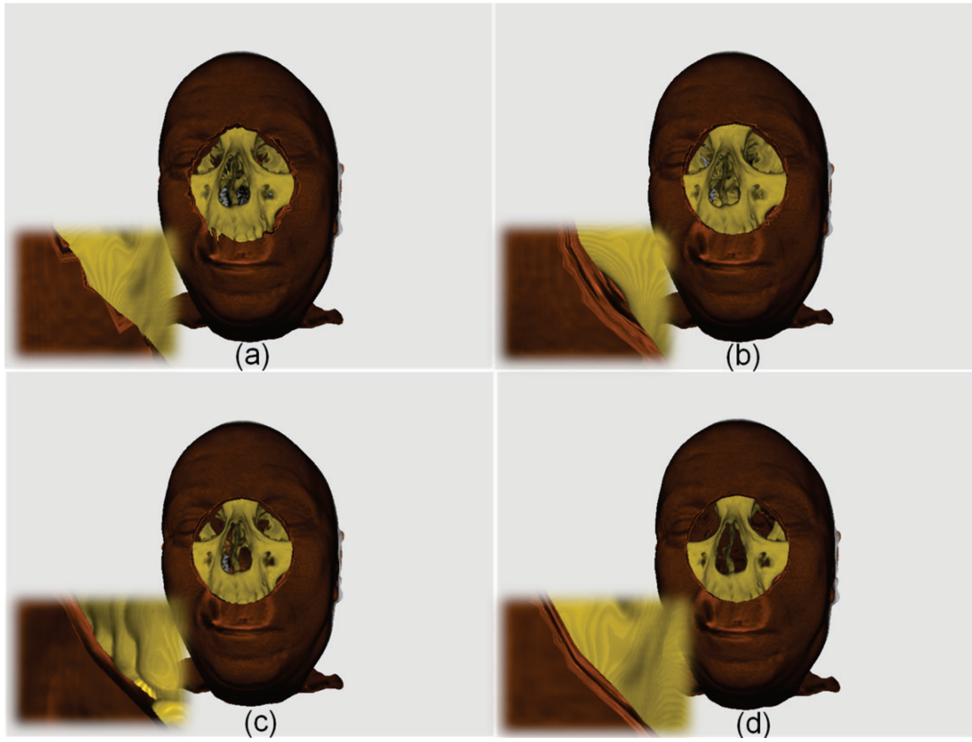


Figure 5.1. Volume editing results with varying number of bounding mesh vertices, (a) 7828, (b) 30218, (c) 59042, (d) 95424 vertices. Zoomed in results show the disappearance of artifacts with increasing mesh resolution.

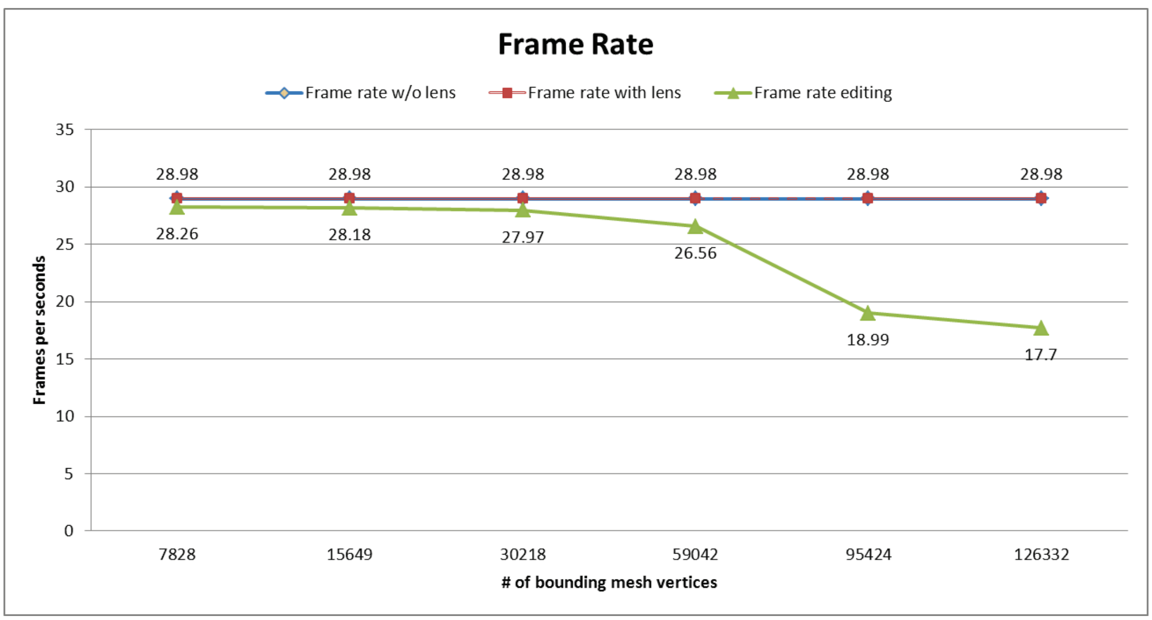


Figure 5.2. Frame rates vs. number of vertices in proxy mesh.

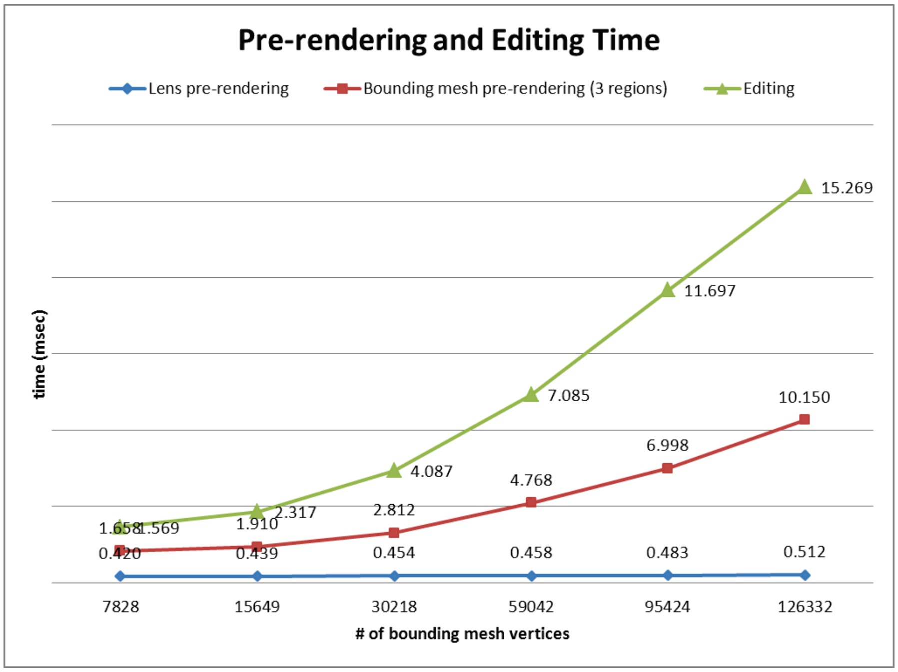


Figure 5.3. Pre-rendering/editing times vs. number of proxy mesh vertices. Note that these times are for rendering three distinct regions.

Another important feature of our volume editing approach is that the performance is largely independent from the brush size used. To demonstrate this, we have performed editing operations with varying brush sizes, as can be seen in Figure 5.4, with corresponding pre-rendering time shown in Figure 5.5. The results show that the editing time is largely unchanged even when the volumetric brush size is increased to cover a large portion of the dataset.

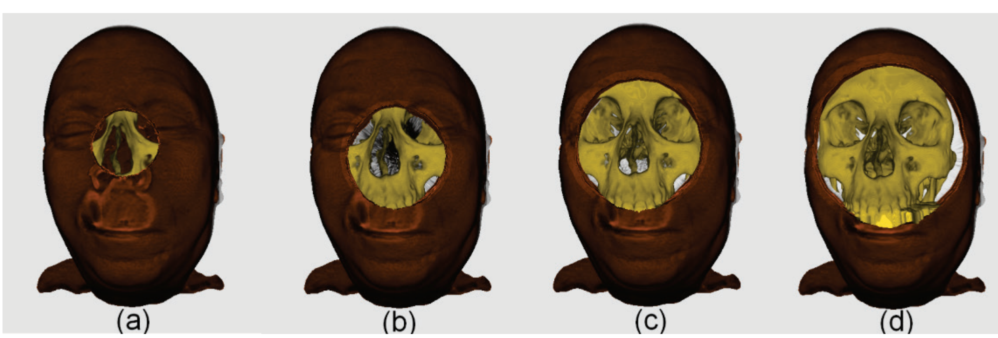


Figure 5.4. Different size lenses for performance comparisons.

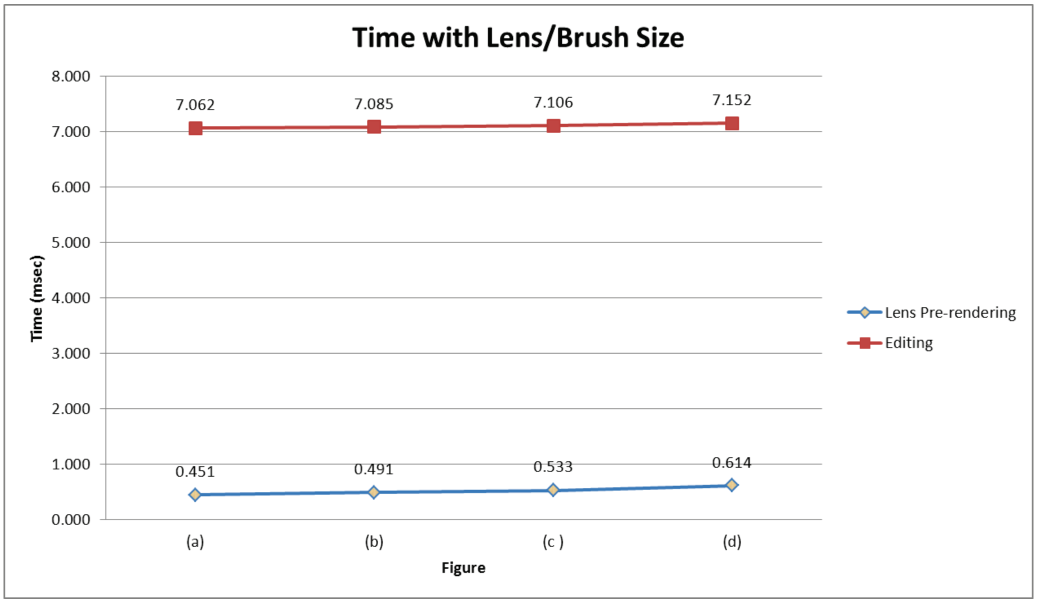


Figure 5.5. Pre-rendering/editing times of different lens/brush sizes.

These results show that our goals of real-time rendering of Magic Lenses and performing volume editing tasks can be realized with the proposed rendering schemes. Figure 5.6 shows an example of such a visualization, while the user has selected arbitrarily shaped regions from co-registered CT, MRI and PET datasets to create a combined visualization that shows the relationships between different anatomical structures.

Another advantage of our approach is the ability to save multiple vertex states for undo/redo operations. We have implemented undo operations that can save these states to memory or hard drive in less than 0.5 seconds with smaller storage space compared to saving a volumetric dataset. For instance, each saved state for the volume editing operation for the results shown in Figure 5.6 takes about 1.63 MBs, which enables multiple undo operations.

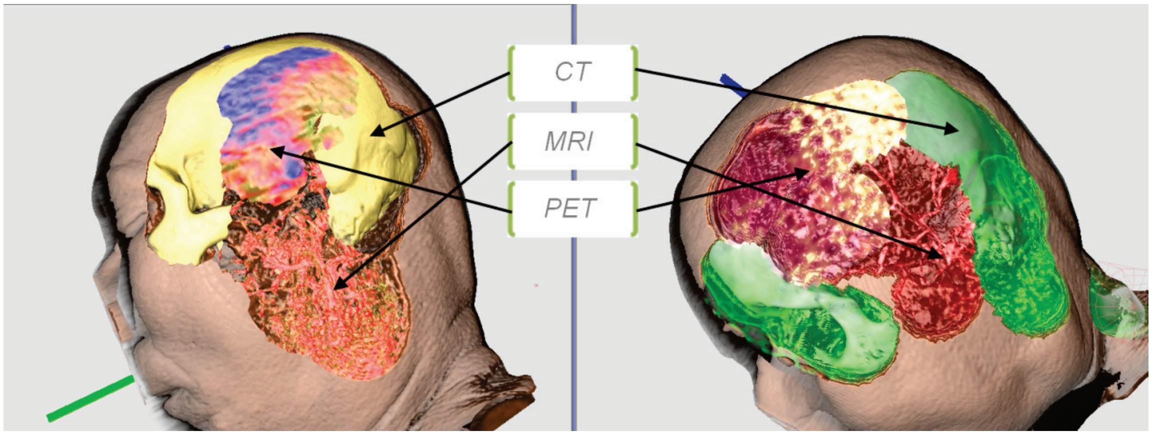


Figure 5.6. Example volume editing result that displays information from three co-registered modalities.

5.3. Analysis of User Studies

In this section, we will analyze the results of the conducted user studies comparing the mouse and gesture-based interfaces by presenting quantitative and qualitative results. This will be followed by a discussion about the implications of these results.

5.3.1. Quantitative Results

Both of the aforementioned experiments were conducted with the same group of volunteers successively, as we believed the tasks were reasonably different and learning effects would not significantly alter the results. The study group consisted of 15 people between the ages of 22 and 38, with an average age of 29.4. Out of the fifteen users, 12 were male and 3 were female. Our subjects were all college-educated adults. None of the users indicated they used the Kinect platform before. 7 of 15 users said they occasionally use software that produces 3D renderings, while the remaining 8 indicated they never use such software. We have performed quantitative analysis using the data collected from the experiment, and qualitative analysis of the interfaces by analyzing a survey users filled out after the experiments. In both experiments, the independent variable was the interface used (Kinect two-hand rotation (K2HR) and traditional mouse rotation (TMR) for Experiment I; mouse slice (MS), Kinect slice (KS) and Magic Lens (ML) for Experiment II). The dependent variables that were measured for both experiments were time and accuracy. The order of interfaces was selected in random for both experiments to offset learning effects, and both experiments were performed twice in independently random orders. The training consisted of performing the same task before the data collection began using a different dataset, for the users to become familiar with the interfaces.

In Experiment I, the users were asked to match a target rotation by using two interfaces. The target rotations were set to one of three possibilities (-45°, 0° and 45°) in X and Y directions, resulting in 9 possible rotation pairs. By performing with all of these rotations twice, each subject performed 18 trials for each of the two interfaces. The accuracy was defined using the quaternion notation, using the quaternion norm of the difference between target and user selected rotations as the accuracy measure. To analyze the performance of the interfaces, the mean value for each user for each interface was used. The time and accuracy distributions of the results are presented with boxplots in Figure 5.7. To test the statistical significance of the results, a 2-tailed, paired-sample t-test with 14 degrees of freedom was used. The t-test produced the following results: for error, the p-value was 0.0066. For time, the p was $< .0001$. These results indicate that the interfaces are significantly different from each other. Therefore, by using the results shown in Figure 5.7 and Figure 5.8, we can conclude the gesture-based interface performed better consistently than the mouse interface for this rotation task both in time and accuracy.

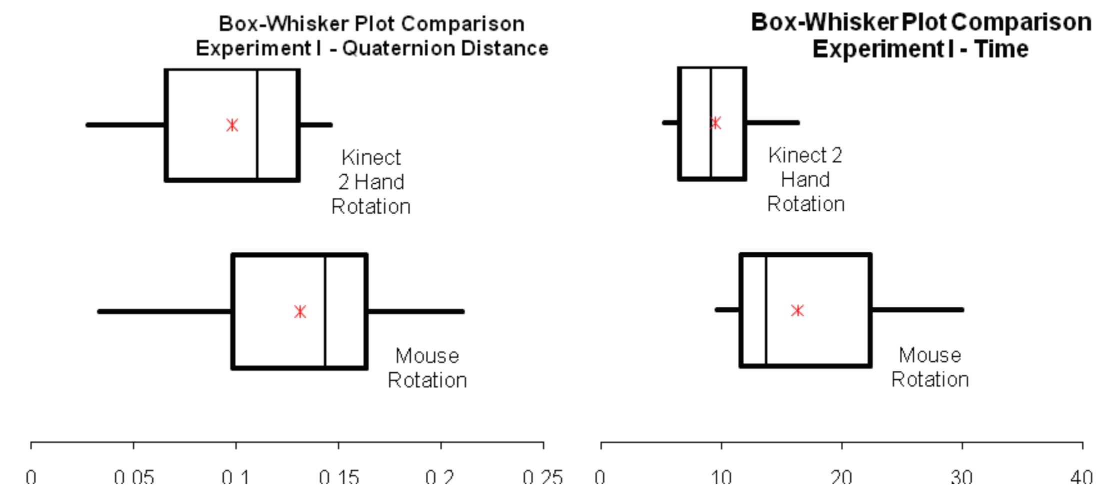


Figure 5.7. Boxplots[1] of Experiment I results.

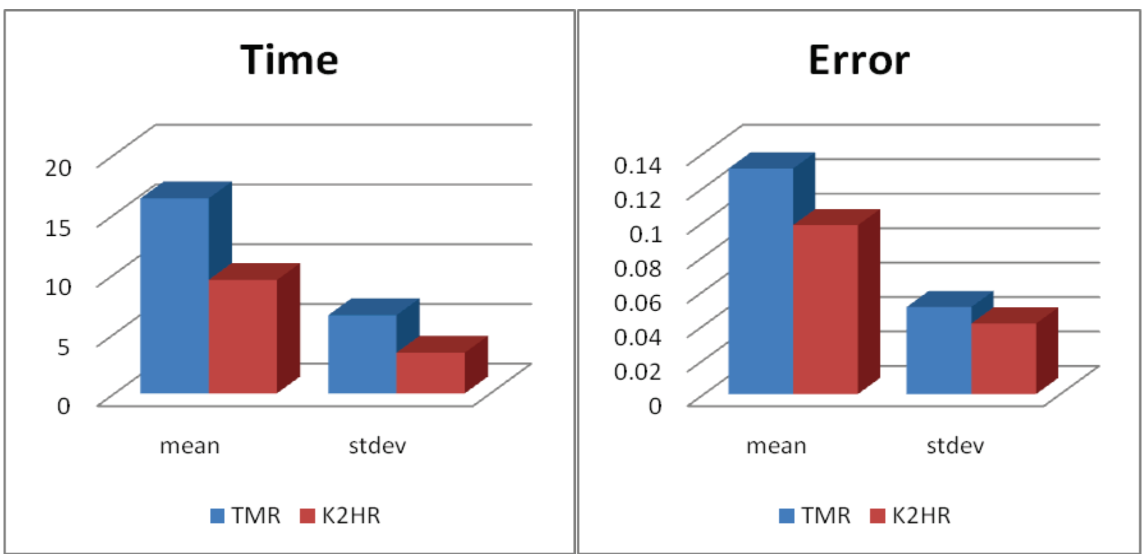


Figure 5.8. Statistical results of interfaces for Experiment I.

---

[1] Boxes extend from 1st to 3rd quartile of observations, with * representing the mean and the vertical line denoting the median. Whiskers extend to observations less than 1.5 interquartile-range (the difference between the third and first quartiles) from the edges of the boxes. Any observations outside that range would be represented with red squares as outliers. As a generalization, a smaller box closer to the left side of the plot can be considered as performing consistently better for this experiment.

The second experiment compared three interfaces: MS, KS and ML. The slice-based interfaces (MS and KS) can be considered 2D visualizations, while the ML was a 3D visualization method. We conducted and analyzed Experiment II in a similar manner to Experiment I: the same training approach, randomized order of trials and mean values of each user for analysis was used. The targets are placed in one of 5 possible locations for each trial, and 9 remaining artificial structures were used as distractors. This meant collecting 10 samples for each interface, since each interface was tested twice for each user. The users were instructed to center the targets in each trial. The boxplot and statistical properties for this experiment is presented in Figure 5.9 and Figure 5.10. The error analysis for this experiment is not as straightforward since two of the interfaces use 2D visualization while ML is 3D. For comparison, we chose to use the projected location of the Magic Lens and used the distance in the Y-axis to the projected center of the target as the error measure. For MS and KS, we used the slice's distance to the target's center. Even though the former error measure is in pixels and the latter in voxels, in our rendering a voxel roughly occupied a single pixel when rendered, so we assumed a voxel and pixel to be the same unit in our analysis. However, other factors might have skewed this error comparison between 3D and 2D visualizations, which will be discussed in more detail later in this section.

For statistical analysis, we used the Analysis of Variance (ANOVA) test. The p-value of when all three interfaces were considered was 0.056. Compared to each other, only ML and MS demonstrated statistically significant difference in terms of time ($p=0.017$). In terms of error, KS and MS significantly outperformed ML ($p<0.01$) and the p-value between MS and KS was 0.068. We believe that these results indicate even though ML can be used as an interface that can be used for quick exploration of datasets; the mouse can perform precise targeting tasks better.

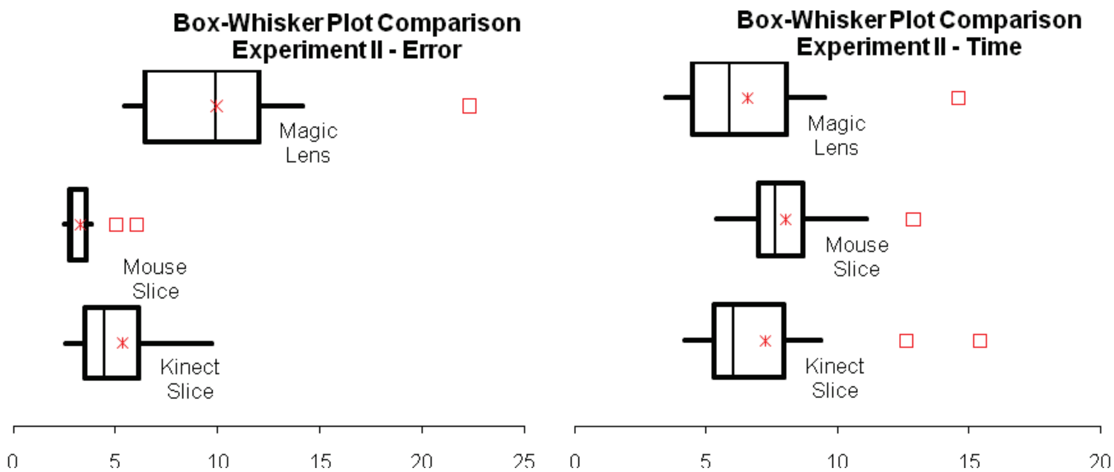


Figure 5.9. Boxplots of Experiment II results.

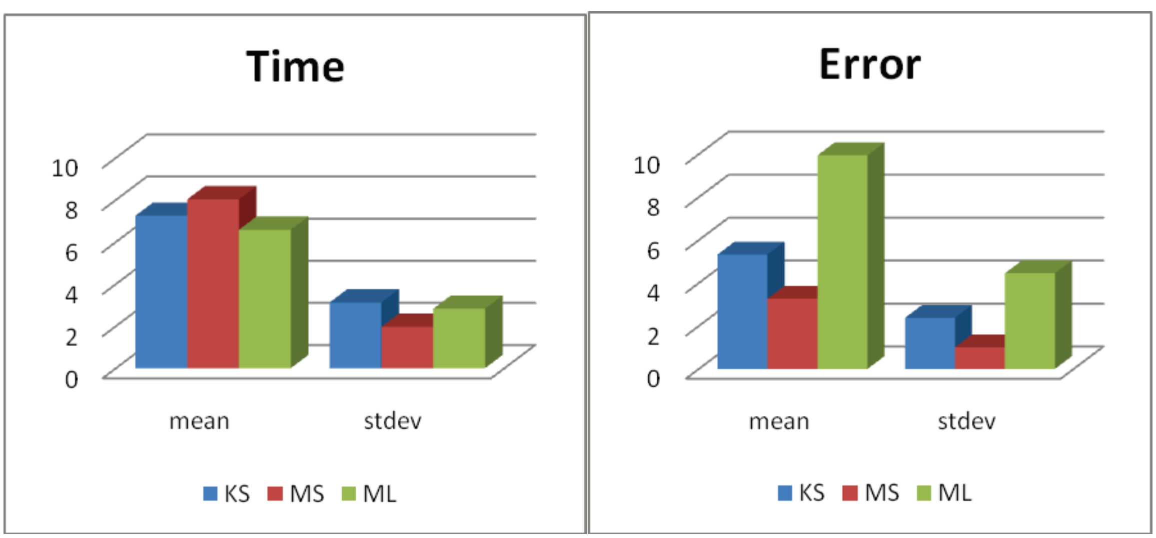


Figure 5.10. Statistical results of interfaces for Experiment II.

5.3.1. Qualitative Results

One of the important aspects of natural user interfaces is intuitiveness and ease of use. The informal feedback was in general very positive, with several users indicating the interface to be ‘fun’ and ‘interesting’. To analyze how users perceived the usability of the gesture-based interface further, users filled out a survey after performing experiments.

For the rotation experiment, the K2HR interface was mostly preferred by the users; with 11 out of 15 (69%) saying K2HR was easier to use than TMR. When asked which interface helped them understand the shape of the object, an even larger preference (14 out of 15, 93%) towards K2HR was indicated.

The Magic Lens interface was also received positively. When asked to rate the ease of use of the interface on a scale of 1 (very easy) to 5 (difficult), an average of 2.06 difficulty (mostly very easy (5) and somewhat easy (6) responses) was given. Similarly, users responded with an average difficulty of 2.13 to the question asking the ease of exploring the internal structures of the object. Moreover, the users showed a preference toward the ML interface compared to the slice-based visualizations, with 11 out 15 indicating the ML interface helped them understand the internal structure of the object better than KS and MS interfaces. The details of these results are given in Table 5-I.

Table 5-I. Survey Results for the Magic Lens interface

How easy was it to use the Magic Lens interface:				
Very easy	*Somewhat easy*	*Neutral*	*Somewhat difficult*	*Difficult*
5	6	2	2	0
How easy was it to explore the internal structures of a 3D dataset using the Magic Lens:				
Very easy	*Somewhat easy*	*Neutral*	*Somewhat difficult*	*Difficult*
4	6	4	1	0
Do you think this tool would improve your understanding of 3D datasets and their relation to the real world (e.g. the patient)?				
Yes	Maybe	No		
12	3	0		

Comparison of the KS and MS interfaces produced more balanced results, with 8 users indicating KS interface was easier to use compared to 7 for MS. However, 10 out of 15 users said KS slice traversal helped them understand the internal structures of the object better.

5.3.2. Discussion

The experiments to evaluate gesture-based interfaces yielded several interesting results. Experiment I showed that a GBI can outperform the mouse in a rotation task. The success of the interface might have come from its similarity to an action that users can relate to (holding and rotating an object), as opposed to the mouse rotation, which is a more abstract mapping. Furthermore, these results were achieved after a short training time using an unfamiliar interface, which points to the intuitiveness of using gestures for rotation tasks.

In the second experiment, the mouse outperformed both gesture-based interfaces in terms of accuracy, which was an expected result given the suitability of the mouse in making precise movements. However, for the Magic Lens interface, some other factors might have contributed to the high error rate. In slice-based visualizations, an accurate match requires the slice to be in an exact position since only a cross-section of the data is displayed. However, even though when the Magic Lens is not perfectly centered at the target location, the target might be inside the lens volume and completely visible. Furthermore, due to perspective projection, the orientation of the Magic Lens might change depending on its location, making it more difficult to center it exactly on the target. These factors, combined with the fact that the Magic Lens outperformed the mouse in terms of time makes us believe that it is still a suitable interface for exploration of datasets. It should be noted for the Magic Lens that the subjects performed the experiments after training with this interface for less than two minutes, while the mouse interactions are extremely familiar, which again points to the intuitiveness of the interface. Moreover, the Magic Lens interface was received favorably by users, and the fact that it can present the inner structures of the dataset in 3D can contribute to the understanding of medical datasets and shapes of internal structures. Furthermore, the users could locate targets more quickly with Magic Lens, therefore in situations where the user has to compare information between several spatial locations (e.g. if the experiment had more than one target with varying sizes larger than the distractors), the Magic Lens can prove to be effective for quick spatial exploration.

Another interesting result of Experiment I was the fact that the subjects were more accurate as well as faster, even though as Experiment II suggests the mouse interface might be better at precise movements. Several factors might have contributed to this result. The first reason is technical: in Experiment II most of the interactions were made with the hand location around the area between the shoulder and camera (making the hand almost perpendicular to the camera), which might sometimes cause problems in the accuracy of pose extraction algorithm used in Kinect. To alleviate this, working space location and arm poses used should be considered in designing gesture based interaction systems. Second possibility comes from the fact that users had the control to indicate when to advance to the next trial. This result might be interpreted as that the users actually understood that they had a match more accurately using the Kinect, possibly using the cues presented by their inherent knowledge of relative locations of their hands. Yet another possible factor is that the mouse interface was simply more difficult to use and to get a rotation match, and the users were more likely to be frustrated and advance to the next trial even though a good match was not achieved. All of these factors could be interesting to study in future research and interface design.

Chapter 6 - CONCLUSION AND FUTURE WORK

This dissertation presented a visualization framework that improves the current medical approaches for surgical planning and intra-operative image guidance. Our methods are predicated on the idea of rendering user-specified local regions differently to improve the information content, and providing the rest of the dataset to give context. This approach aims to alleviate the problems associated with volume rendering, while helping with the mental registration task between the 3D renderings and the patient. The techniques were implemented using the advantages modern GPUs provide, and novel methods were proposed to define and visualize arbitrarily shaped 3D volumetric regions in real-time. These visualization methods were applied to a gesture-based interface to overcome the problems associated with the use of tactile interactions inside the operating room. User studies were undertaken to analyze the feasibility and intuitiveness of the proposed methods.

Our methods are flexible in terms of how the rendering methods are defined in focus and context regions. In this work, we proposed using different transfer functions based on intensity values and using different datasets for inside/outside user selected regions. Many transfer function approaches proposed so far can be easily incorporated to our framework, examples include gradient [18], texture [15] and size-based [17] transfer functions. As additional datasets, computer simulations are increasingly being used to provide medical information. For instance, vocal fold correction surgeries can be improved by computational fluid dynamics simulation datasets [93, 94] that show the airflow necessary for phonation [90]. Fusion of 2D images and volume datasets can also improve the success of endoscopic surgical procedures [98]. Addition of these different rendering techniques and datasets to our framework can improve the effectiveness of surgical tasks.

Even though satisfactory rendering performance and visual quality is achieved, some improvements to performance and usability of our approach are possible. To improve the pre-rendering times, techniques such as dual-depth peeling [99] or stencil routed k-buffering [100] can be used to extract multiple depth layers to perform pre-rendering passes more quickly. Our method for changing vertex positions of bounding meshes was proposed for its performance and similarity to our focus+context rendering framework, but other real-time mesh deformation methods can be used to ensure continuity along selection boundaries and to eliminate possible visual artifacts completely.

In our implementation, even though the users can interactively change datasets/rendering parameters inside each region, we assumed a fixed front-to-back ordering of regions to maximize performance. To perform interactive reordering of regions or for effects such as blending of different regions, depth sorting can be applied before the raycasting step, and rays can be segmented and rendered in the desired order. For this, compositing approaches [101] or a shader factory approach [30] can be considered, with necessary modifications to ensure real-time performance for volume raycasting. The same approach can be used for rendering multiple Magic Lenses and a single-pass rendering method for multi-lens rendering can be developed using our polygon-assisted approach.

The user studies showed that gesture-based interfaces can be effective at rotation tasks, and can be used for quick exploration of volumetric datasets. However, depending on the task, precision of these interfaces can be worse than using the mouse. To improve this further processing such as adding smoothing filters might be considered. Another important improvement would be robust methods for engagement/disengagement of actions. This can be achieved by image processing methods, and we believe adding support for hand gestures will increase the possible kinds of actions possible. Previous research suggests that people naturally use gestures when they are looking at or discussing visualizations [66], therefore robust ways need to be defined to let the interaction system know which gestures are aimed for interaction, and which are expressive or explanatory. Especially in tasks that use both hands, the users need to have an intuitive method to let the system know they want to interact with the system. Multimodal inputs such as voice commands can also be considered. Furthermore, being able to recognize familiar gestures may improve the intuitiveness of the interaction. For instance, recognition for hand gestures such as grasping can improve the intuitiveness of rotation and translation tasks. Our volume editing methods can be improved by applying cut-outs and translating them to different locations by using grasping gestures to avoid occlusion.

The experiments presented in this work used non-medical and synthetic datasets to evaluate the intuitiveness of the proposed interfaces to unfamiliar users. To test the applicability of these approaches to medical settings, further controlled experiments using medical datasets and trials in the operating room will be necessary. We believe our results are very encouraging for the future of gesture-based interfaces and Magic Lens visualization in surgical applications.

Another area of medical visualization that needs further analysis is the cognitive aspects of how users understand 3D renderings. Effects of things like user abilities [51, 52, 62], interactivity [61, 102], different types of depth cues or projection methods used [103, 104] have been studied with sometimes conflicting findings, and will be important to take into account for future volume visualization applications. Surgical interventions are complex tasks that have many variables affecting the performance. Analyzing specific aspects of the why visualizations methods are successful or even unsuccessful can give us valuable insight for future improvements and ideas. In particular, we believe the mechanisms of mental registration of real and virtual spaces in an image guided surgery context requires further research.

REFERENCES

[1] U. Sure, O. Alberti, M. Petermeyer, R. Becker, and H. Bertalanffy, "Advanced image-guided skull base surgery," *Surgical Neurology,* vol. 53, pp. 563-572, 2000.

[2] J. Wadley, N. Dorward, N. Kitchen, and D. Thomas, "Pre-operative planning and intra-operative guidance in modern neurosurgery: a review of 300 cases," *Ann R Coll Surg Engl,* vol. 81, pp. 217-25, Jul 1999.

[3] R. W. Lindeman. (2010, March 2011). *Acceptance Rates for Publications in Virtual Reality / Graphics / HCI / Visualization / Vision*. Available: http://web.cs.wpi.edu/~gogo/hive/AcceptanceRates/#CHI

[4] C. Boucheny, G. P. Bonneau, J. Droulez, G. Thibault, and S. Ploix, "A Perceptive Evaluation of Volume Rendering Techniques," *Acm Transactions on Applied Perception,* vol. 5, pp. -, Jan 2009.

[5] (2011). *Usability First, Usability Glossary.* Available: http://www.usabilityfirst.com/glossary/focuscontext/

[6] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*: Morgan Kaufmann, 1999.

[7] J. Lamping, R. Rao, and P. Pirolli, "A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Denver, Colorado, United States, 1995.

[8] M. Cohen and K. Brodlie, "Focus and context for volume visualization," in *Theory and Practice of Computer Graphics, 2004. Proceedings*, 2004, pp. 32-39.

[9] Y.-S. Wang, C. Wang, T.-Y. Lee, and K.-L. Ma, "Feature-Preserving Volume Data Reduction and Focus+Context Visualization," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 17, pp. 171-181, 2011.

[10] H. Doleisch, "SIMVIS: interactive visual analysis of large and time-dependent 3D simulation data," presented at the Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come, Washington D.C., 2007.

[11] H. Doleisch, M. Gasser, and H. Hauser, "Interactive feature specification for focus+context visualization of complex simulation data," presented at the Proceedings of the symposium on Data visualisation 2003, Grenoble, France, 2003.

[12] H. Doleisch, H. Hauser, M. Gasser, and R. Kosara, "Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data," *Simulation,* vol. 82, pp. 851-865, 2006.

[13] H. Doleisch, H. Hauser, M. Gasser, and R. Kosara, "Interactive focus plus context analysis of large, time-dependent flow simulation data," *Simulation-Transactions of the Society for Modeling and Simulation International,* vol. 82, pp. 851-865, Dec 2006.

[14] S. Bruckner and M. E. Gröller, "Style Transfer Functions for Illustrative Volume Rendering," *Computer Graphics Forum,* vol. 26, pp. 715-724, 2007.

[15] J. J. Caban and P. Rheingans, "Texture-based Transfer Functions for Direct Volume Rendering," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 14, pp. 1364-1371, 2008.

[16] 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," presented at the Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics, Tokyo, Japan, 2003.

[17] C. Correa and M. Kwan-Liu, "Size-based Transfer Functions: A New Volume Exploration Technique," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 14, pp. 1380-1387, 2008.
[18] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 8, pp. 270-285, 2002.
[19] C. Cheng-Kai, R. Thomason, and M. Kwan-Liu, "Intelligent Focus+Context Volume Visualization," in *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, 2008, pp. 368-374.
[20] I. Viola, A. Kanitsar, and M. E. Groller, "Importance-Driven Volume Rendering," presented at the Proceedings of the conference on Visualization '04, 2004.
[21] P. Rautek, S. Bruckner, and M. Eduard Groller, "Semantic Layers for Illustrative Volume Rendering," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 13, pp. 1336-1343, 2007.
[22] P. Rautek, S. Bruckner, and E. Gröller, "Interaction-Dependent Semantics for Illustrative Volume Rendering," *Computer Graphics Forum,* vol. 27, pp. 847-854, 2008.
[23] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose, "Toolglass and magic lenses: the see-through interface," presented at the Proceedings of the 20th annual conference on Computer graphics and interactive techniques, Anaheim, CA, 1993.
[24] K. Perlin and D. Fox, "Pad: an alternative approach to the computer interface," presented at the Proceedings of the 20th annual conference on Computer graphics and interactive techniques, Anaheim, CA, 1993.
[25] M. C. Stone, K. Fishkin, and E. A. Bier, "The movable filter as a user interface tool," presented at the Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence, Boston, Massachusetts, United States, 1994.
[26] S.-J. Lee, J. K. Hahn, J. A. M. Powell, and G. Greene, "INSPECT: a dynamic visual query system for geospatial information exploration," in *Visualization and Data Analysis 2003*, Santa Clara, CA, USA, 2003, pp. 312-322.
[27] J. Viega, M. J. Conway, G. Williams, and R. Pausch, "3D magic lenses," presented at the Proceedings of the 9th annual ACM symposium on User interface software and technology, Seattle, Washington, United States, 1996.
[28] C. M. Best and C. W. Borst, "New Rendering Approach for Composable Volumetric Lenses," in *Virtual Reality Conference, 2008. VR '08. IEEE*, 2008, pp. 189-192.
[29] C. W. Borst, J. P. Tiesel, and C. M. Best, "Real-Time Rendering Method and Performance Evaluation of Composable 3D Lenses for Interactive VR," *IEEE Transactions on Visualization and Computer Graphics,* vol. 16, pp. 394-406, May-Jun 2010.
[30] C. W. Borst, J.-P. Tiesel, E. Habib, and K. Das, "Single-Pass Composable 3D Lens Rendering and Spatiotemporal 3D Lenses," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 17, pp. 1259-1272, 2011.
[31] T. Ropinski and K. Hinrichs, "Real-Time Rendering of 3D Magic Lenses having arbitrary convex shapes," presented at the Proc. Int'l Conf. in Central Europe on Computer Graphics (WSCG '04), 2004.
[32] D. Weiskopf, K. Engel, and T. Ertl, "Interactive clipping techniques for texture-based volume visualization and volume shading," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 9, pp. 298-312, 2003.
[33] M. Trapp and J. Döllner, "Efficient Representation of Layered Depth Images for Real-time Volumetric Tests," in *EG UK Theory and Practice of Computer Graphics (2008) Conference*, 2008, pp. 9-16.

[34] L. Wang, Y. Zhao, K. Mueller, and A. Kaufman, "The magic volume lens: an interactive focus+context technique for volume rendering," in *Visualization, 2005. VIS 05. IEEE*, 2005, pp. 367-374.

[35] J. Looser, M. Billinghurst, and A. Cockburn, "Through the looking glass: the use of lenses as an interface tool for Augmented Reality interfaces," presented at the Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Singapore, 2004.

[36] M. Erick, K. Denis, and S. Dieter, "Interactive context-driven visualization tools for augmented reality," presented at the Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, 2006.

[37] J. Plate, T. Holtkaemper, and B. Froehlich, "A Flexible Multi-Volume Shader Framework for Arbitrarily Intersecting Multi-Resolution Datasets," *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, pp. 1584-1591, 2007.

[38] J. Kruger, J. Schneider, and R. Westermann, "ClearView: An Interactive Context Preserving Hotspot Visualization Technique," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 12, pp. 941-948, 2006.

[39] N. Svakhine, D. S. Ebert, and D. Stredney, "Illustration motifs for effective medical volume illustration," *Computer Graphics and Applications, IEEE,* vol. 25, pp. 31-39, 2005.

[40] D. Ebert and P. Rheingans, "Volume illustration: non-photorealistic rendering of volume models," presented at the Proceedings of the conference on Visualization '00, Salt Lake City, Utah, United States, 2000.

[41] N. A. Svakhine, D. S. Ebert, and W. M. Andrews, "Illustration-Inspired Depth Enhanced Volumetric Medical Visualization," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 15, pp. 77-86, 2009.

[42] S. Bruckner and M. E. Gröller, "VolumeShop: An Interactive System for Direct Volume Illustration," 2005, pp. 85-85.

[43] A. Lu, C. J. Morris, D. S. Ebert, P. Rheingans, and C. Hansen, "Non-photorealistic volume rendering using stippling techniques," presented at the Proceedings of the conference on Visualization '02, Boston, Massachusetts, 2002.

[44] R. van Pelt, A. Vilanova i Bartroli, and H. van de Wetering, "Illustrative Volume Visualization Using GPU-Based Particle Systems," *Visualization and Computer Graphics, IEEE Transactions on,* vol. PP, pp. 1-1, 2010.

[45] W. Chen*, et al.*, "Volume Illustration of Muscle from Diffusion Tensor Images," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 15, pp. 1425-1432, 2009.

[46] C. D. Correa, D. Silver, and M. Chen, "Feature Aligned Volume Manipulation for Illustration and Visualization," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 12, pp. 1069-1076, 2006.

[47] S. Bruckner, "Exploded Views for Volume Data," *IEEE Transactions on Visualization and Computer Graphics,* vol. 12, pp. 1077-1084, 2006.

[48] K. Burger, J. Kruger, and R. Westermann, "Direct Volume Editing," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 14, pp. 1388-1395, 2008.

[49] BrainLab. (2011). *Cranial Navigation Software*. Available: http://www.brainlab.com/art/2811/4/cranial-navigation-application/

[50] R. S. Sidhu*, et al.*, "Interpretation of three-dimensional structure from two-dimensional endovascular images: implications for educators in vascular surgery," *Journal of Vascular Surgery,* vol. 39, pp. 1305-1311, 2004.

[51] M. Hegarty, M. Keehner, C. Cohen, D. Montello, and Y. Lippa, "The Role of Spatial Cognition in Medicine: Applications for Selecting and Training Professionals," 2007.
[52] E. Zudilova-Seinstra*, et al.*, "Exploring individual user differences in the 2D/3D interaction with medical image data," *Virtual Reality,* vol. 14, pp. 105-118, 2010.
[53] K. R. Wanzel*, et al.*, "Visual-spatial ability correlates with efficiency of hand motion and successful surgical performance," *Surgery,* vol. 134, pp. 750-757, 2003.
[54] A. R. Teyseyre and M. R. Campo, "An Overview of 3D Software Visualization," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 15, pp. 87-105, 2009.
[55] M. St. John, M. B. Cowen, H. S. Smallman, and H. M. Oonk, "The Use of 2D and 3D Displays for Shape-Understanding versus Relative-Position Tasks," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 43, pp. 79-98, January 1, 2001 2001.
[56] S. H. Park and J. C. Woldstad, "Multiple two-dimensional displays as an alternative to three-dimensional displays in telerobotic tasks," *Human Factors,* vol. 42, pp. 592-603, Win 2000.
[57] M. Tory, A. E. Kirkpatrick, M. S. Atkins, and T. Moller, "Visualization task performance with 2D, 3D, and combination displays," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 12, pp. 2-13, 2006.
[58] M. Tory, T. Moller, M. S. Atkins, and A. E. Kirkpatrick, "Combining 2D and 3D views for orientation and relative position tasks," presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, Vienna, Austria, 2004.
[59] M. Tory, S. Potts, and T. Moller, "A parallel coordinates style interface for exploratory volume visualization," *IEEE Transactions on Visualization and Computer Graphics,* vol. 11, pp. 71-80, Jan-Feb 2005.
[60] M. C. Velez, D. Silver, and M. Tremaine, "Understanding visualization through spatial ability differences," in *Visualization, 2005. VIS 05. IEEE*, 2005, pp. 511-518.
[61] M. Keehner, M. Hegarty, C. Cohen, P. Khooshabeh, and D. R. Montello, "Spatial Reasoning With External Visualizations: What Matters Is What You See, Not Whether You Interact," *Cognitive Science,* vol. 32, pp. 1099-1132, 2008.
[62] P. Khooshabeh and M. Hegarty, "Inferring Cross-Sections: When Internal Visualizations Are More Important Than Properties of External Visualizations," *Human–Computer Interaction,* vol. 25, pp. 119 - 147, 2010.
[63] Z. Yaniv and K. Cleary, "Image-guided procedures: A review," *Computer Aided Interventions and Medical Robotics,* 2006.
[64] J. Tanz. (2011) Kinect Hackers Are Changing the Future of Robotics. *Wired Magazine*. Available: http://www.wired.com/magazine/2011/06/mf_kinect/
[65] R. Johnson, K. O'Hara, A. Sellen, C. Cousins, and A. Criminisi, "Exploring the potential for touchless interaction in image-guided interventional radiology," presented at the Proceedings of the 2011 annual conference on Human factors in computing systems, Vancouver, BC, Canada, 2011.
[66] A. G. Hauptmann, "Speech and gestures for graphic image manipulation," *SIGCHI Bull.,* vol. 20, pp. 241-245, 1989.
[67] D. A. Bowman, *3D user interfaces : theory and practice*. Boston: Addison-Wesley, 2005.
[68] Y.-K. Ahn*, et al.*, "3D spatial touch system based on time-of-flight camera," *WSEAS Trans. Info. Sci. and App.,* vol. 6, pp. 1433-1442, 2009.
[69] P. Breuer, C. Eckes, and S. Müller, "Hand Gesture Recognition with a Novel IR Time-of-Flight Range Camera–A Pilot Study," in *Computer Vision/Computer Graphics*

Collaboration Techniques. vol. 4418, A. Gagalowicz and W. Philips, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 247-260.

[70] J. Wachs*, et al.*, "Gestix: A Doctor-Computer Sterile Gesture Interface for Dynamic Environments," in *Soft Computing in Industrial Applications*. vol. 39, A. Saad*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 30-39.

[71] R. A. Griggs, *Psychology: A concise introduction*: Worth Pub, 2008.

[72] T. Sielhorst, C. Bichlmeier, S. Heining, and N. Navab, "Depth Perception – A Major Issue in Medical AR: Evaluation Study by Twenty Surgeons," ed, 2006, pp. 364-372.

[73] J. Kruger and R. Westermann, "Acceleration Techniques for GPU-based Volume Rendering," presented at the Proceedings of the 14th IEEE Visualization 2003 (VIS'03), 2003.

[74] W. Leung, N. Neophytou, and K. Mueller, "SIMD-Aware Ray-Casting," presented at the Volume Graphics 2006, Boston, MA, 2006.

[75] A. Joshi*, et al.*, "Novel interaction techniques for neurosurgical planning and stereotactic navigation," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 14, pp. 1587-1594, 2008.

[76] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," presented at the Proceedings of the 14th annual conference on Computer graphics and interactive techniques, 1987.

[77] M. J. Sholl and T. L. Nolin, "Orientation Specificity in Representations of Place," *Journal of Experimental Psychology: Learning, Memory, and Cognition,* vol. 23, pp. 1494-1507, 1997.

[78] A. T. Stull, M. Hegarty, and R. E. Mayer, "Getting a handle on learning anatomy with interactive three-dimensional graphics," *Journal of Educational Psychology,* vol. 101, pp. 803-816, 2009.

[79] (2010). *The Stanford 3D Scanning Repository*. Available: http://graphics.stanford.edu/data/3Dscanrep/

[80] T. M. Peters and K. Cleary, *Image-guided interventions : technology and applications*. New York: Springer, 2008.

[81] T. M. Peters, "Image-guidance for surgical procedures," *Physics in Medicine and Biology,* p. R505, 2006.

[82] D. P. Perrin*, et al.*, "Image Guided Surgical Interventions," *Current Problems in Surgery,* vol. 46, pp. 730-766, 2009.

[83] *Image-guided interventions : technology and applications*. New York: Springer, 2008.

[84] D. L. Pham, C. Xu, and J. L. Prince, "Current Methods in Medical Image Segmentation," *Annual Review of Biomedical Engineering,* vol. 2, pp. 315-337, 2000.

[85] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3D medical image segmentation: A review," *Medical Image Analysis,* vol. 13, pp. 543-563, 2009.

[86] D. J. Hawkes*, et al.*, "Tissue deformation and shape models in image-guided interventions: a discussion paper," *Medical Image Analysis,* vol. 9, pp. 163-175, 2005.

[87] D. J. Hawkes*, et al.*, "Computational Models In Image Guided Interventions," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, 2005, pp. 7246-7249.

[88] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley, "A Survey of Medical Image Registration on Multicore and the GPU," *Ieee Signal Processing Magazine,* vol. 27, pp. 50-60, Mar 2010.

[89] H. Lester and S. R. Arridge, "A survey of hierarchical non-linear medical image registration," *Pattern Recognition,* vol. 32, pp. 129-149, Jan 1999.

[90] C. Kirmizibayrak, M. Wakid, S. Bielamowicz, and J. Hahn, "Interactive Visualization for Image Guided Medialization Laryngoplasty," presented at the Computer Graphics International 2010, Singapore, 2010.

[91] S. Bielamowicz, "Perspectives on medialization laryngoplasty," *Otolaryngologic Clinics of North America,* vol. 37, pp. 139-+, Feb 2004.

[92] G. Jin*, et al.*, "Image guided medialization laryngoplasty," *Computer Animation and Virtual Worlds,* vol. 20, pp. 67-77, 2009.

[93] H. Luo, R. Mittal, and S. A. Bielamowicz, "Analysis of flow-structure interaction in the larynx during phonation using an immersed-boundary method," *Journal of The Acoustical Society of America,* vol. 126, 2009.

[94] H. Luo*, et al.*, "An immersed-boundary method for flow–structure interaction in biological systems with application to phonation," *Journal of Computational Physics,* vol. 227, pp. 9303-9332, 2008.

[95] F. OsiriX. (2011, 06/01). *DICOM sample image sets*. Available: http://pubimage.hcuge.ch:8080/

[96] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.,* vol. 21, pp. 163-169, 1987.

[97] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision,* vol. 13, pp. 119-152, 1994.

[98] Y. Yim, M. Wakid, C. Kirmizibayrak, S. Bielamowicz, and J. K. Hahn, *Registration of 3D CT Data to 2D Endoscopic Image using a Gradient Mutual Information based Viewpoint Matching for Image-Guided Medialization Laryngoplasty* vol. 4, 2010.

[99] L. Bavoil and K. Myers, "Order Independent Transparency with Dual Depth Peeling," 2008.

[100] K. Myers and L. Bavoil, "Stencil Routed K-Buffer," 2007.

[101] S. Bruckner*, et al.*, "Hybrid visibility compositing and masking for illustrative rendering," *Computers & Graphics,* vol. 34, pp. 361-369, 2010.

[102] T. Sando, M. Tory, and P. Irani, "Effects of animation, user-controlled interactions, and multiple static views in understanding 3D structures," presented at the Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization, Chania, Crete, Greece, 2009.

[103] A. Corcoran, N. Redmond, and J. Dingliana, "Perceptual enhancement of two-level volume rendering," *Computers & Graphics,* vol. 34, pp. 388-397, 2010.

[104] K. Votanopoulos, F. C. Brunicardi, J. Thornby, and C. F. Bellows, "Impact of three-dimensional vision in laparoscopic training," *World Journal of Surgery,* vol. 32, pp. 110-118, Jan 2008.