Generalized Temporal Focus + Context Framework for Improved Medical Data Exploration

by Nadezhda Lyubomirova Radeva

B.A. in Computer Science and Astrophysics, May 2009, Connecticut College
M.S. in Computer Science, January 2012, The George Washington University

A Dissertation submitted to

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

January 31, 2014

Dissertation directed by

James K. Hahn
Professor of Computer Science

The School of Engineering and Applied Science of The George Washington University certifies that Nadezhda L. Radeva has passed the Final Examination for the degree of Doctor of Philosophy as of December 03, 2013. This is the final and approved form of the dissertation.

Generalized Temporal Focus + Context Framework for Improved Medical Data Exploration

Nadezhda Lyubomirova Radeva

Dissertation Research Committee:

James K. Hahn, Professor of Computer Science, Dissertation Director

Abdou Youssef, Professor of Computer Science, Committee Member

Gabe Sibley, Assistant Professor of Computer Science, Committee Member

John Philbeck, Professor of Psychology, Committee Member

Lucien Levy, Professor of Radiology, Committee Member

Acknowledgements

I would like to thank my advisor Dr. James Hahn for his guidance and support, helping me to pursue my research interests and to fully utilize my potential of becoming an independent researcher. By providing a productive environment where research ideas were always welcome to be discussed, he helped me to expand my knowledge in many areas and to shape my career path.

I would also like to extend my thanks to my dissertation committee - Dr. Abdou Youssef, Dr. Gabe Sibley, Dr. John Philbeck and Dr. Lucien Levy for providing constructive criticism, suggestions for improvement that have strengthened my work, and for their help with the dissertation process. My work would not be possible without the help of Dr. Levy who provided a very interesting idea that became the base of my research project.

I would also like to thank Dr. Kirmizibayrak, Dr. Wakid and Dr. Yim for helping me to transition to the research environment in the computer graphics lab, all anonymous volunteers who participated in the conducted user study and Samar Alsaleh for her help in the design and execution of the study.

This work has been sponsored by the GW Centers and Institutes Facilitating Fund and the GW Office of Graduate Student Assistantships and Fellowships.

And last but not least, I would like to extend my warmest thank you to my parents, sister and Dr. Marek Kucka for always believing in me and supporting me through the challenging times that were part of the process of me becoming an experienced researcher.

Abstract of Dissertation

Generalized Temporal Focus + Context Framework for Improved Medical Data Exploration

Radiologists and surgeons have different visualization demands due to the diversity in work performed by each group of specialists. Many techniques have been developed to visualize 2D slices and 3D volumes for dataset exploration, surgical planning and practicing of intervention procedures, however, they do not address the individual visualization needs. We introduce a generalized temporal focus + context framework that provides a unique novel way of combining the visualization needs of specialists while working individually or collaboratively. Thus, we can use the framework to describe and compare existing visualization techniques and to identify novel combinations of rendering techniques that can be applied to multimodal medical datasets to create visualizations beneficial to both radiologists and surgeons during their everyday work.

The generalized focus + context framework defines various regions that can be used to render a combination of visualization modalities. An examination of how data is rendered inside each region provides us with improved visualizations for 2D and 3D rendering that are beneficial to both radiologists and surgeons. We use the focus region defined by a magic lens to render 2D slices and a 3D sub-volume. The interactive rendering of sub-volumes augmented by slices provides a way to explore the inner structure of target objects. A user study showed that users obtained a more precise structural understanding of complex structures when allowed to explore sub-sections of the whole focus region. We apply a new hierarchy of context areas defined by the temporal positions of the focus region to improve explicit spatial perception. Previous

lens positions define a new context region named focus-driven context that can be used to render past focus or as a painting brush to create a region where a separate co-registered dataset can be displayed. A user study experiment showed that when using focus-driven context, users are faster and more accurate in inferring the spatial relationship between multiple objects than when using a small lens. The generalized framework also provides a new visualization that addresses the need of surgeons to explore possible trajectories to target structures. We create a visualization with an arbitrary-shaped focus region defined by the surgeon's lens movement. A user study experiment asking users to virtually sculpt a volume until they reach a target showed that the above-mentioned visualization is significantly faster and more accurate than using an opaque sculpting tool. No context rendering, on the other hand, can provide an endoscopic visualization with improved visibility of occluded areas. All of these techniques can be utilized within the same interactive session by multiple users. Therefore the framework satisfies the visualization needs of different types of medical specialists during team meetings.

A fast GPU-based raycasting algorithm is used to support the interactivity of the visualizations. A combination of depth peeling and a novel way to compute parity is used when an arbitrary-shaped focus region is to be rendered. All raycasting techniques are combined in a way such that the system maintains its interactive speeds even when multiple rendering modalities are used simultaneously.

In conclusion, the described work addresses the lack of visualizations designed for the work of both surgeons and radiologists during dataset exploration and surgical planning. The unifying approach of the focus + context framework intends to show how such a framework can be efficiently incorporated in the clinical workflow.

Table of Contents

List of Figures:

List of Tables

Chapter 1: Introduction

1.1 Motivation

In the everyday practice of medical diagnosis and surgical planning, physicians rely on exploring 2D images of the patient's anatomy. This task requires physicians to look at large amounts of patient data, where one dataset often contains hundreds of slices. Depending on the medical case, a patient might undergo different imaging modality scans - Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), X-ray or others, in order for a correct diagnosis or the most suitable surgical plan to be established. Physicians have to mentally register and reconstruct the multiple imaging modality scans taken at different time intervals and with different imaging systems. At the same time they have to explore the volume and locate any target structures. Currently employed techniques for medical image exploration involve scrolling through 2D slices while the physician has to mentally reconstruct the 3D volume relying mostly on his experience and knowledge in the field of specialty. Medical training is crucial for the correct implicit volume reconstruction. High concentration is necessary for processing all small details captured on the images. Lack of experience, extreme fatigue or heavy workload can cause inaccurate mental registration of multiple modalities, as described in [1].

Even though physicians use the same visualization systems that show 2D slice or a 2D+3D visualization, there is an observed differentiation of preferred visualization methods between the different groups of medical teams. While radiologists examine 2D orthogonal slices to mentally reconstruct the patient's volume, surgeons may prefer to examine slices that will improve their understanding of the structures that make up a

planned exploratory path. Furthermore, radiologists may use 3D volume renderings to get an overview of the whole dataset or the target objet. Surgeons, on the other hand, may use such 3D visualizations to explore the structures that lie along a planned resection path and have to be avoided or removed. The preferred visualization modality is often dependent on the group of specialists. For example, radiologists have extensive training in reading radiographs and are very skillful in reading the 2D images and implicitly creating 3D models of the observed anatomy. In some cases radiologists might have to use volumetric renderings due to the nature of the object (i.e. the complex organization of vascular structures), but they would still refer to the 2D scans to ensure they can see as much detail as possible and to measure distances without the presence of occluding objects. In contrast, surgeons prefer to examine the patient's dataset using non-axis aligned 2D slices or a 3D volumetric rendering. In this approach surgeons can study the shape and position of all organs as they would be seen during the actual surgical procedure. Even though surgeons use 2D scans for the target object exploration, they need to also visualize structures that are far from the target anatomy but are important for the intervention planning. In such cases, a 3D volumetric rendering is preferred to slices.

A recent study presented by Kainz et al. [2] explored some of the possible reasons why medical specialists still prefer looking at sequences of 2D slices over exploring 3D renderings of the same dataset. As seen on Figure 1-1, a strong preference for 3D volume renderings was observed when they would speed up the finding of abnormalities compared to looking at cross-sections; when team members want to present their finding in a faster and easier to understand way, and when the mental reconstruction of high-dimensionality datasets is difficult. If personal preferences were compared, 2D images

were used for general diagnosis and accurate planning of interventions. The survey shows that the only application in which specialists would not use 3D volume rendering is accurate intervention planning. Slight preference of 3D over 2D visualizations was observed in cases where inter- and intra-medical communications had to be performed, and in surgery and intervention planning. On the other hand, 3D rendering algorithms were always preferred for the investigation of multiple advanced imaging sequences.

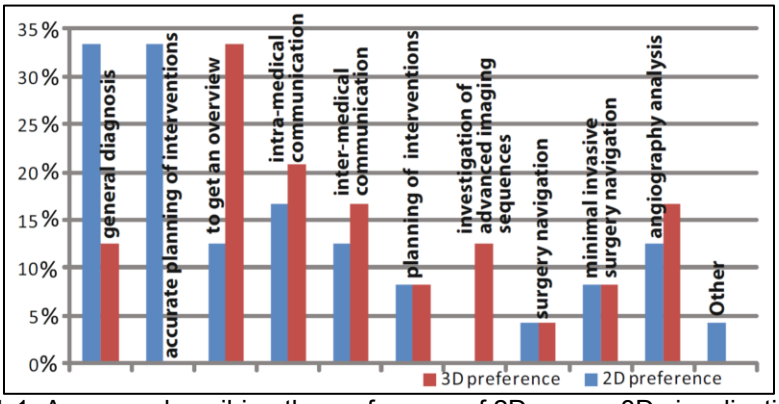

Figure 1-1: A survey describing the preference of 2D versus 3D visualizations. [2]

To help physicians with the task of data exploration, multiple visualization techniques have been developed. The most common of them allow segmentation of anatomical structures of interest and the 3D visualization of the segmented objects. When examining the 3D rendered volume, physicians face several disadvantages of the utilized visualization techniques. If the volume is extracted using iso-surfaces [3], only structures with the specified threshold are extracted, therefore not all important structures are visible. Seeing other structures requires adjusting the threshold value. Direct volume rendering techniques, for instance raycasting, can successfully display a wide range of anatomies if an appropriate transfer function is applied. However, similar to 3D surface rendering techniques, occlusion can cause less important structures to hide the target. In such cases the user has to adjust the transfer function, rotate the object to find a better

viewing angle, or use additional software that automatically calculates the optimal transfer function and viewing parameters.

Many research projects have concentrated on the fact that while 2D slices provide detailed information, volume visualizations reduce the mental burden of volume reconstruction. Examples like 3DSlicer [4] and ExoVis [5] demonstrate the successful combination of slices and 3D volume visualizations by rendering the 3D volume, as well as providing axis-aligned or arbitrary-oriented slices that display details at important areas. Even though such techniques are incorporated in medical software and are used by physicians, there are still problems with these visualizations. For example, while rendering an extracted 3D volume, slices might be occluded by the volume rendered in front. Decreasing the opacity of the volume to ensure proper visibility of the 2D slice causes a reduced amount of visible volumetric information. Rendering the slices on projection planes outside the volume or on a separate window provides less spatial relationship information between the volume and the slice.

To solve the problem of limited spatial relationship cues between internal structures, researchers have proposed many independent techniques that display the desired structures with enhanced correlation. Some of these techniques include exploded views [6], clipping volumes [7], and volume peeling [8]. While most of these techniques combine only one or two datasets at a time, the focus + context technique, first introduced by Viega et al. [9] to display a focus region rendered differently from the rest of the volume (considered context), can be applied to multiple datasets simultaneously. The focus region displays important structures from one dataset, while context is used to provide overview information extracted from a second dataset. Thus, a physician can use

this visualization paradigm to better understand how important structures are spatially related to other anatomical landmarks by allowing the removal of occluding objects from the context or by zooming in on focus areas of interest. This has been shown in the work of Kirmizibayrak et al. [10, 11] where the interactivity of their method provides real-time feedback of the voxel-based data exploration. Even though this technique provides a lot of contextual information, it still faces the problem of volume occlusion due to improper transfer functions. Focus + context has been applied to medical datasets in terms of 3D volume rendering only, where the magic lens is considered a volumetric rendering of all slices that fall within the bounding box of the lens geometry. A physician who wants to explore the inner structure of a target structure would sometimes not benefit from such a volume rendering due to occlusion and will have to refer back to 2D slices.

While current visualization systems support generic 2D and 3D renderings, they fail to provide tools that can unify the different approaches of how radiologists and surgeons utilize them. Therefore, radiologists and surgeons often use multiple different software systems in order to place an accurate diagnosis and plan a surgical procedure. This can be distracting as different users utilize their own preferred approach and it often requires data to be exported in certain formats that are compatible with the input requirements of different systems. We are unaware of any work done in the area of unifying different visualization techniques within one framework that satisfies the different exploratory demands of specialists. In this dissertation we describe a solution that provides an easy way of comparing and selecting the most appropriate visualizations, and a way of improving current 2D and 3D rendering techniques by defining a generalized temporal focus + context framework developed to classify the most widely used visualization

techniques applied to medical data. It can also be used to compare different approaches employed by researchers and medicals specialists when deciding which visualization to use for a specific case. Having a generalized framework for medical visualizations gives insights on possible visualization paradigms that have not been explored before but would be beneficial in the work of physicians when additional information is to be explored. Five such new paradigms are presented in the following chapters, with their primary goal to provide the visualization modalities needed by both surgeons and radiologists simultaneously. Furthermore, being part of a unifying framework, such visualizations can be used in multidisciplinary medical team meetings during which radiologists present medical cases and their findings by showing 2D slices while surgeons explore the 3D structures important for an accurate resection path planning. The introduced research is important because it provides a novel visualization paradigm that can satisfy the needs of all team members and can display the desired information from different angles and in any visualization dimensionality, making the meeting more time efficient. Moreover, various visualizations can be easily compared when using the framework, allowing users to choose the visualization most suitable for a specific task.

1.2 Research method

To address the need of having a way to unify existing and novel visualization paradigms that can further enhance both dataset exploration and preoperative planning, we developed a unified framework for 2D, 3D and 3D sub-volume visualizations in orthogonal and arbitrary planes. In our work we use the magic lens focus + context paradigm as a means to establish a generalized framework, where the context and lens

geometries can be changed to fall within any of the existing techniques. By adjusting the x, y and z dimensions of each separate geometry, we can define visualizations that provide 2D, 3D, and 2D+3D renderings, as well as the well-known magic lens visualization.

In our generalized focus + context framework, the context area is a volume rendering of a user-specified dataset, while the focus region is an interactive geometry that can be moved in 3D space. Using the presented framework and relying on the temporal position change of the focus region, we identified new visualization paradigms that address disadvantages of currently used techniques.

The focus region of the new visualization paradigms provides a new way of time-dependent display of 2D slices that can be further augmented by a 3D sub-volume rendering defined by the oldest viewed and the currently viewed slices. The problem of volume occlusion due to too much irrelevant information in front of the target object is addressed by enabling a sub-volume rendering within the focus region so that only important areas are displayed. We created a new interactive visualization method that does not rely on the limited implicit memory of a physician to deduce the overall shape and spatial relationship between multiple structures of interest. Furthermore, it provides added contextual information which is directly related to the exploratory actions performed by the user. Two other new techniques based on the framework can help surgeons explore possible incision trajectories by either volume sculpting using the focus region, or by using the focus region as an improved endoscopic visualization with reduced field of visibility. 2D and 3D renderings inside the sculpting tool, for example, provide additional accuracy of the tool's position, allowing for precise exploration of the

regions to be cut out. The visualization techniques can be used interchangeably to explore multimodal datasets, find abnormalities, and study the surrounding volumetric structures.

The development of new visualization techniques with large flexibility in rendering modalities and interaction methods resulted in applying depth peeling and a novel approach for parity computations to the utilized raycasting method. All of the described visualizations therefore support multimodal datasets with multiple regions in real-time.

To evaluate and validate the advantages of our proposed framework and the new visualizations, we executed a user study that compared existing and the novel techniques.

1.3 Original contributions

The developed new generalized temporal focus + context framework can be used to classify and compare existing visualization techniques so that the optimal visualization can be easily selected for every unique case. Furthermore, by examining the general placement of focus and context in time we developed new visualization paradigms that can be used in both data exploration and intervention planning. These new paradigms aim to combine the advantages of all three methods (2D slice exploration, 3D volume rendering and focus + context visualization) in order to provide an optimal time-dependent set of spatial relationship, shape and content cues. The successful combination of most requirements for medical data exploration by both radiologists and surgeons is achieved by a unified focus + context visualization that supports advanced imaging sequences, minimally invasive surgery navigation, and accurate intervention planning.

The original contributions of our work are:

1. Create an easy to manipulate generalized temporal framework that is used to
    - Classify and compare existing 2D, 3D and 4D techniques in order to allow for a fast determination of the most appropriate visualization technique,
    - Define new visualization paradigms that provide explicit spatial and structural cues,
    - Support the simultaneous combination of multiple renderings into one visualization that can be used during collaborative medical meetings.
2. Apply the framework to multimodal medical datasets to
    - Incorporate 2D slice rendering inside the focus region,
    - Improve the 3D structural understanding of focus structures by allowing physicians to interactively define and display sub-sections inside the focus,
    - Establish time-dependent focus-driven context used for explicit spatial relationship and shape perception between multiple important regions further supported by a hierarchy of context areas.
3. Provide a way of combining the medical visualizations and modalities typically used by surgeons and radiologists within the same system. Therefore, different users can enable their preferred rendering technique, while also being able to see what other specialists explore.
4. Develop an algorithm based on depth peeling and parity computation needed for improved ray segment computation when volume sculpting is performed. We provide an improved way for parity computation so that an unlimited number of saved tool positions and a complex focus geometry are supported.

5. Execute a user study that explores the possible advantages of the proposed generalized framework-based visualizations when compared to existing ones.

1.4. Organization of dissertation

This dissertation is organized as follows.

In Chapter 2 we review work related to the topics of combining 2D and 3D visualizations, focus + context visualizations and depth perception improvements for medical visualizations. Chapter 3 presents the main contributions of our work. The generalized focus + context framework is introduced, and the resulting new visualizations are presented in detail. Chapter 4 describes the challenges encountered during the ray traversal computations necessary for all introduced techniques and the implemented improvements of the used raycasting algorithm. Performance evaluation for all techniques is provided. Chapter 5 presents the results of a user study that compared the new generalized focus + context framework-based approaches with existing ones by evaluating user performance and accuracy. Chapter 6 summarizes the results obtained by the new visualization approaches and the improved raycasting method. Future work is presented.

Chapter 2: Related Work

2.1 2D and 3D renderings

2.1.1 2D slice visualization

Magnetic Resonance Imaging (MRI), Computed Tomography (CT), X-ray and ultrasound (US) imaging are some of the most widely used methods for scanning the human body in order to study any abnormalities, place a diagnosis and plan an interventional procedure. MRI, CT, US and X-ray imaging systems allow the obtained slices to be oriented in an arbitrary or axis-aligned direction [12]. The slices are taken at different spacing (between 0.5 and 5 mm) depending on the scanned organ. Therefore, a typical head scan can contain hundreds of slices, so that even the smallest changes in anatomical structure can be detected.

The widespread use of computers and their increasing power has shifted slice image exploration from light boxes to a computer screen where software is used to display the series of images in the three most widely used directions – axial, sagittal and coronal as shown in Figure 2-1, and supports scrolling through the slices. Some software also allows arbitrary slice orientation to further enhance the understanding of the examined anatomy.

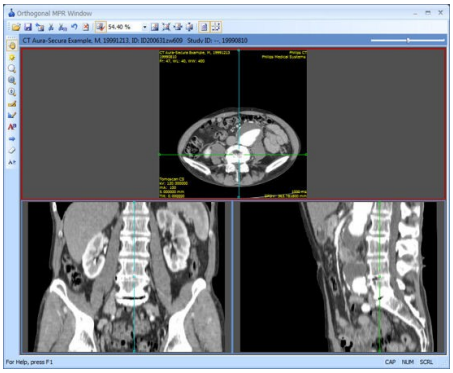

Figure 2-1: Axial, sagittal and coronal views used by physicians for the exploration of medical datasets. [13]

The main disadvantage of the approach of looking at a series of cross-sectional images is that the radiologist has to mentally reconstruct the 3D volume by scrolling through the slices. Well-trained radiologists are successful in the 3D implicit reconstruction, however, novices find this task very difficult. Even the most experienced physicians can misinterpret scans, especially when the examined structures are blood vessels. For example, scans of blood vessels are difficult to read because the blood vessels represented on each cross section can take many shapes, including circular or oval shapes. The physician has to keep track of the changes in shape and size while scrolling through the images so that he can mentally reconstruct the 3D model of the complex blood vessel structure.

2.1.2 3D volume visualization

Due to the limitations of viewing 2D slices, researchers have done extensive work in extracting the information from the datasets and creating 3D volume visualizations of the anatomy. Such algorithms consider the sequence of 2D slices as a 3D grid structure, where each pixel from the original slice is a three-dimensional point called a voxel and contains a value equal to the intensity value of the pixel from the slice. There are two main methods that have become state-of-the-art 3D volume rendering algorithms in the medical domain – isosurface extraction and direct volume rendering (DVR).

3D volume extractions can be done using the polygonal isosurface extraction approach first proposed by [3]. Many improvements of the initial method have been proposed - adding more cases for the polygon extraction [14], introducing tetrahedral decomposition [15], or performing calculations on the GPU [16, 17]. The final result is a

polygonal representation of the extracted data, which is then rendered using any polygon rendering hardware and software. There are a few disadvantages of the isosurface approach that make it unsuitable for some datasets. False positives or false negatives in the extracted polygons result in inaccurate surface reconstruction. Depending on the complexity of the data scan, one object might consist of hundreds of thousands of polygons, which causes a large overhead in the geometric rendering pipeline. Due to the dependence on a single parameter, this technique cannot display multiple objects with different intensities from the same scan, thus it ignores much of the data.

The DVR approach provides a more accurate 3D volumetric visualization algorithm. In this type of visualizations no intermediate representation of the data is used, unlike the iso-surface extraction. One of the most studied techniques in volume rendering is raycasting. It employs the use of virtual rays that start at the position of the virtual camera and point at the direction of the pixels that belong to the final image. The cast rays pass through the 3D dataset and accumulate color and opacity while sampling the dataset along the specified ray position. The sampling along the ray can be done at equal intervals or at varying spacing, the latter approach speeding up the rendering time.

Due to the high number of rays (for a final image of size 512 pixels by 512 pixels there are over 2.6×10^5 rays) and samples within the dataset, multiple techniques have been proposed to speed up the calculations on CPU. Early ray termination and empty space skipping are often used for faster ray traversal. Octree representations of the dataset can speed up the sampling of the ray since the ray is sampled only if it passes through a tree node that contains information [18]. In the past few years Graphics Processing Units (GPUs) have been used for achieving interactive raycasting speeds. Other work has

concentrated on defining different types of transfer functions for detailed and efficient representation of the dataset values. Transfer functions can be 1D or 2D. 1D transfer functions map intensity values from the original dataset to color and opacity. 2D transfer functions [19] depend on the intensity and gradient magnitude, allowing for a more precise mapping of intensity to RGBA and for extracting material boundaries. Improvements of the 2D transfer function concept have been explored so that multimodal datasets can be merged together and important structures from each dataset are visible. Such techniques include a transfer function design based on isosurfaces extracted from each dataset and their similarities [20], general information theory - based approaches [21] or the notion of visibility of each co-registered dataset within pre-segmented regions of interest [22].

2.1.3 3D volume rendering augmented with 2D slices

Physicians often view datasets by using software designed to display one slice at a time, where slices can be in the axial, sagittal or coronal planes. There are multiple commonly used techniques to combine 2D slices and 3D volume visualizations inside the interface of a single environment. Many commercial and research systems [4, 23, 24] display the three orthogonal views in their own windows, and the user controls a scrollbar to go through the sequence of slices. A 3D visualization is displayed in a separate window so that the user can get a better volumetric understanding of the dataset and can also improve his understanding of the slice location with respect to the volume. The 3D visualization is augmented by displaying a combination of the three axis-aligned slices in the same window. While the user changes the slice position, he can see the exact position

of the slice in the volume visualization window and a detailed view of the selected slice on the side as in Figure 2-2. However, such an approach creates additional requirements of adjusting the threshold values used in the calculation of the transfer function or the iso-surface extraction in order to visualize specific 3D features. Furthermore, it has to be able to overcome occlusion of the 3D volume from other anatomical structures or the displayed 2D slices.

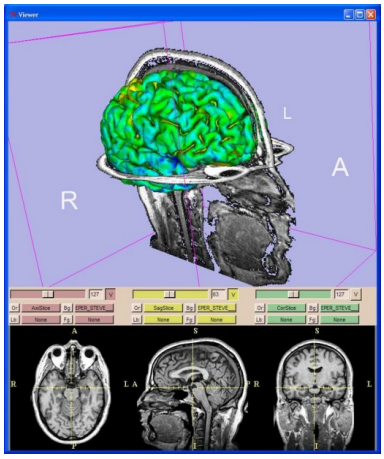

Figure 2-2: Combining 2D and 3D renderings. 3D rendering of the segmented brain from the dataset with the corresponding orthogonal slices is shown in the top window. The three orthogonal views are also shown on the bottom. [25]

Research in the area of picking target objects from a visible slice and 3D object selection has further extended the work in combining 2D and 3D visualizations. When an object from a slice is selected, 3D rendering parameters are adjusted to make sure that the objects on the picked slice are visible. For example, LiveSync [26] relies on picking on the 2D slice in order to adjust the transfer function and viewing direction. The system automatically computes the optimal viewing direction. It adjusts the initial transfer function so that the picked object is visible, ensuring no occlusion of the selected object as shown in Figure 2-3. LiveSync uses the slice position to optimize the viewing parameters and then renders the 3D volume in an optimal way, which does not provide an

explicit way of combining 2D and 3D data in the same visualization. When there are multiple successive slices of interest, the rendering parameters for two consecutive images might vary significantly, causing perception confusion.

Figure 2-3: 2D slice picking. 2D slice picking (**a**) results in the optimal transfer function and viewing direction for the dataset (**b**). The same picked point is made visible and pointed at in the optimal 3D rendering (**c**). [26]

The other way of incorporating slices and 3D volume renderings by picking is to allow the user to select a point on the 3D rendering and then display the slice corresponding to the picked position. The slice orientation and position are adjusted so that the slice provides the most information about the picked 3D point. For example, in the work of Wiebel at al. [27] a 3D rendering of the dataset with a predefined transfer function is used by the user to select a point of interest. Once the point is selected (Figure 2-4a), an automatic algorithm uses the first and second derivative of the accumulated opacity curve to find the 3D position of the picked object. Finally, it displays a slice that goes through the middle of the selected 3D object. While the 3D rendering can be disabled so that an unobstructed view of the slice can be examined as in Figure 2-4c, the preferred visualization is a 3D volume augmented with the 2D slice as visible on Figure 2-4b. A disadvantage of 3D volume picking to select a slice by translating 2D picked screen coordinates into 3D virtual world space is that a physician might face difficulties when he wants to explore consecutive slices.

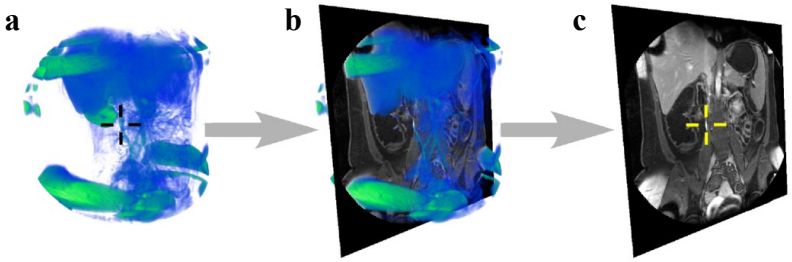

Figure 2-4: 3D volume picking. Picking on a 3D object feature (**a**) finds the corresponding slice and renders it at the center of the picked object (**b**). The 3D rendering can be disabled so that only the computed slice is visible (**c**). [27]

Researchers in medical visualizations have developed various techniques for improved simultaneous visualization of slices and 3D volumes. Tietjen et al. [28] have created a 2.5D visualization using LiftChart which is shown in Figure 2-5. This technique displays the range of slices that a segmented anatomical structure spans while the user scrolls through the slices. The slice visualization is augmented by rendering the segmented 3D structures using different transfer functions. This allows the user to see an overview of the volume while exploring the slice and to explicitly see the slice position with respect to the whole volume. However, this technique requires initial segmentation of the important structures, otherwise the span of slices cannot be determined and the 3D volume visualization can become cluttered.

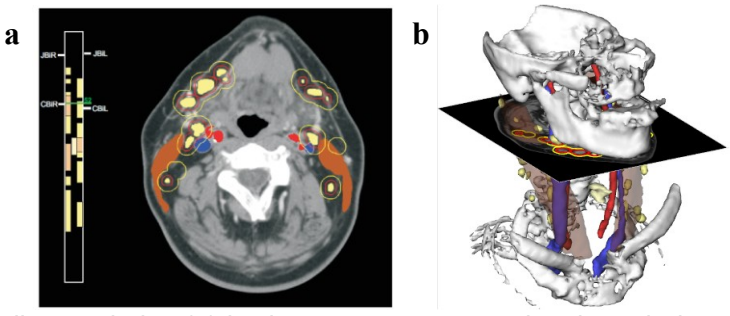

Figure 2-5: The slice rendering (**a**) is shown as a cross section through the volume rendering of the dataset (**b**). [28]

Other ways of combining 2D and 3D data renderings include manipulation of the 3D volume so that it is removed using clipping planes or pushed to the side as exploded

views so that a 2D slice or the inside of the volume are displayed. The simplest approach is to cut the 3D volume along a 2D plane and remove anything that lies in front of the plane as shown in the related work in [5]. While a detailed 2D slice at a non-axis aligned direction can be displayed, the disadvantage of this approach is that anything in front of the slice is no longer visible, therefore spatial and structural contextual information about the volume is lost. To prevent loss of spatial information about the clipped regions, [6] presents an approach where one or multiple cutting planes determine into how many regions the outer layers of the visualization are cut – Figure 2-6. These regions are translated and rotated to the side so that the inside of the volume is visible. The plane along which the cut is performed is used to display a detailed slice-like view of the dataset. A similar exploded view technique is presented in [7]. The depth structure and a voxelized object are used to determine the clipping geometry so that the correct volume is cut away and moved to the side. Unlike [6], this approach allows for a cut in an arbitrary shape, therefore it provides more structural information of the removed object analogous to drilling out large and deep segments of the anatomy with convex surgical tools. Once the cut structures are removed or moved to the side, the user cannot visualize slices inside them, or explore the inner structure of a certain region, which makes such approaches inapplicable to the work in this dissertation.

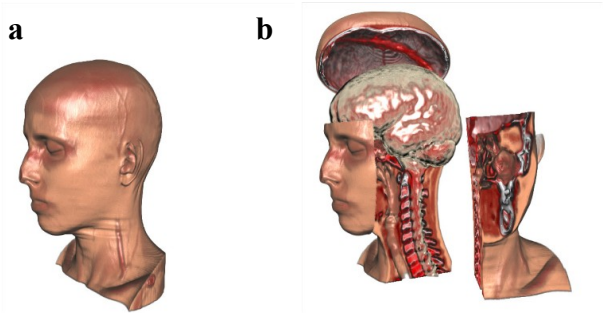

Figure 2-6: Exploded views. (**a**) The original volume rendering. (**b**) Three cuts of the outer layer with a large degree-of-explosion. [6]

Another way to explore datasets is to peel away layers from the 3D visualization [8]. Each layer can be axis-aligned, surface-aligned or feature-aligned - each of these techniques is shown in Figure 2-7. A GPU real-time rendering allows for a surgery simulation with a peeling effect, for example simulating the opening of an incision so that the surgeon can reach the target organ. Different sizes and orientations of the peeled layer can simulate various tasks, however they usually serve exploratory purposes only and do not provide detailed information of the target structure and the path to it which is important for both surgeons and radiologists. Furthermore, 2D slice rendering or exploration of inner structures is not possible with this approach.

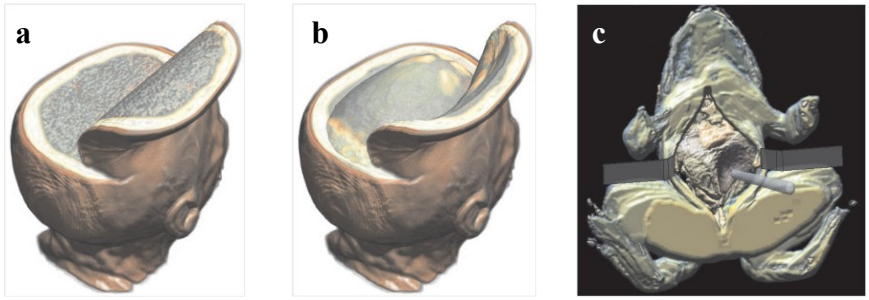

Figure 2-7: Axis-aligned peel (**a**), surface-aligned peel (**b**), peeling operator (**c**). [8]

The necessity to further improve the techniques used to display a 3D volume and slices within the same view, as well as to provide more explicit focus and context, have been addressed in [5]. In general, focus is considered the target slice or volume of interest, while context provides additional information about nearby structures necessary for placing diagnosis and planning surgical interventions. In Tory et al.'s proposed technique ExoVis, the user can explore the volume in two ways, both of which are shown in Figure 2-8. The first one provides a volume rendering where arbitrarily positioned slice placeholders display the corresponding slice on a "wall". While the placeholder provides contextual position information, the slice can be seen in full detail on the side. The

placeholder can take any position and orientation to allow for better volume exploration. Its transparency can be adjusted to give a varying degree of explicit spatial perception between the volume and the slice. A fully opaque placeholder would occlude the 3D volume behind it, while a fully transparent placeholder corresponds to a wireframe slice representation. ExoVis also provides 3D callouts that display focus and context. While the context is volumetrically rendered, the focus area is identified as a wireframe cube in the context area and rendered on the side as an isosurface. This technique is one of the early implementation of overview and detail visualizations, however the rendering of focus far from the context requires implicit spatial relationship understanding. The advantage of the presented callout technique is that it can display a 3D volume rendering or a 2D slice view of the area of interest. However, the user cannot examine only a subsection of the focus 3D volume.

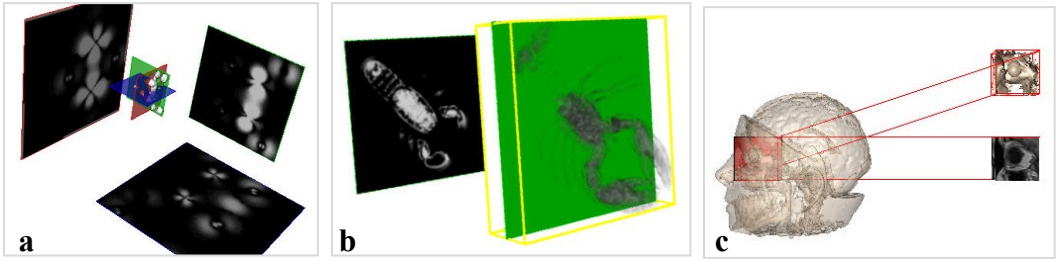

Figure 2-8: ExoVis wall and callout visualizations. (**a**) Three orthogonal slices are projected on walls. (**b**) Placeholder shows position of the slice with respect to the rendered volume. (**c**) 3D callout example, the eye is rendered using isosurface extraction. [5]

The rendering of slices on "walls" surrounding the volume has also been explored by König et al. [29] to provide a better understanding of the position of high functional activity areas with respect to the anatomical rendering of a brain dataset. The visualization displays different modality 2D renderings of functional data on the sides and under the main dataset. This approach provides limited spatial perception cues and can cause disorientation since the user has to look at four different rendering areas.

None of the presented techniques in combining 2D and 3D dataset renderings addresses the different requirements that radiologists and surgeons have. While radiologists would prefer an approach that allows scrolling through slices and a 3D rendering to give an overview of the volume, surgeons might require a different volumetric visualization that shows a 3D rendering from the position of the optimal incision point. A surgeon may also prefer to see non-axis aligned 2D slices to improve the understanding of areas that are ambiguous in the volume visualization, but are not essential for the visualization. Therefore, medical specialists often have to use separate systems, which can cause confusion when discussing a case and developing a treatment plan.

2.2 Focus + Context visualizations

Focus + context techniques have been widely used in graph visualizations where clutter and occlusion can severely decrease data understanding. Since it was first proposed by Bier et al. [30] to handle filtering of 2D data, it has been extended to 2.5D and 3D visualizations. The interactive lens in the form of a 2D shape, also called a magic lens, causes everything under it to be rendered in a way specified by one of the selected filters. This paradigm has been applied to graph visualizations where the focus region removes some of the clutter to display only the most important information as shown in [31]. Magic lenses have also been used as a distortion tool for better exploration while the focus area is being magnified or subjected to other deformations as in Figure 2-9. Linear and non-linear transformations have been applied to 2D datasets to zoom in on areas of

interest by preserving spatial relationship or to provide multiple zoomed areas with varying size [32-37].

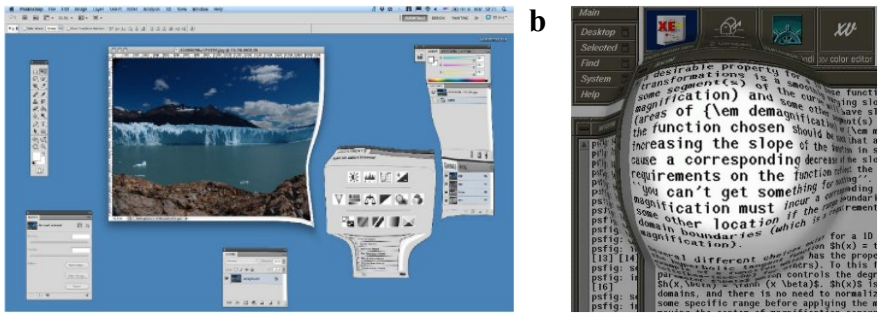

Figure 2-9: Examples of 2D lenses. A distortion lens (**a**) [33], and a magnification lens (**b**) [34].

The first application of a magic lens in the medical domain was proposed by Viega et al. [9]. Magic lens focus + context visualizations provide a context area that gives general spatial perception cues. The focus area is a volume rendered in a way different from the context so that the user extracts more knowledge about the dataset and structures of interest. For example, Viega et al. renders the volumetric focus region as an x-ray tool while the context provides a rendering of the whole volume. Their approach uses rendering of the six sides of the lens frustum, making this rendering not interactive.

Focus + context techniques have been used broadly in medical dataset exploration. An example of such a visualization provides an illustrative dataset visualization. A focus region is rendered with a different transfer function to improve the visibility of the target area with respect to the illustrative context. The visualization can be adjusted to a surgeon mode, where the illustrative rendering of the context is switched to a more realistic rendering of the focus and context as if the surgeon was looking at the patient in a similar manner to what is shown in Figure 2-10 [38]. However, this approach renders context

using illustrative techniques, and it does not support a volume augmentation by a cross-sectional slice, or exploration of the inner structure of the focus structures.

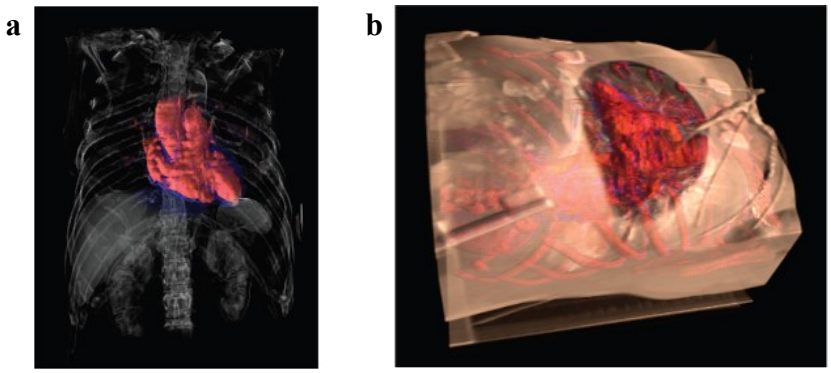

Figure 2-10: Focus + Context visualization of the heart. (**a**) The focus object (the heart) is made visible by adjusting the rendering of the rest of the volume. (**b**) The same focus region rendered in a surgical view. [38]

ClearView [39] is another example of a focus + context rendering that uses a magic lens region to render a defined area in a specific way, separating it from the context rendering. ClearView is one of the few detail and overview publications that give examples of rendering 2D slices inside the focus region and augmenting the slice visualization by extracting some of the volumetric features inside the focus as seen in Figure 2-11. However, the given example is used to show that the author's context-preserving 3D volume rendering inside the lens outperforms the same volume visualization when the focus is displayed using 3D rendering augmented by a 2D slice. This is different from our goal of a visualization that provides flexibility to accurately explore the focus dataset by displaying it in a way that helps both radiologists and surgeons. ClearView's goal is to preserve as much context as possible even in the focus region. We believe that this might not be of particular importance to physicians whose

goal is to accurately plan an intervention and who want to see very specific areas from the focus dataset.

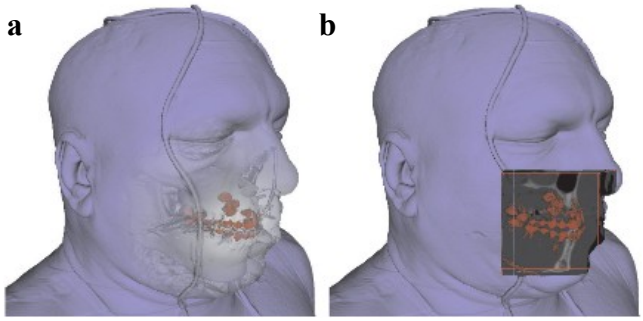

Figure 2-11: Focus + Context with context-preserving rendering - the focus object is dental enamel. (**a**) Bone and skin are preserved even though they are considered context. (**b**) No context preserving – only the focus region is visible. [39]

Much work has been devoted to 3D magic lens deformations that allow preservation of focus and context relationship while the focus volume is distorted for better exploration. Yang et al. [40] presented multiple perspective deformations applied to a 3D volume to create spherical and cylindrical distortions, as well as to magnify the volume inside the lens. LaMar et al. [41] and Wang et al. [42] have introduced a magnification lens in 3D that allows zooming in on small areas while still preserving the context. It provides a smooth transition between zoomed and unaltered areas for a better understanding of how focus and context are related as shown in Figure 2-12. Conformal mapping has been used in [43] to create a magnifying magic lens and preserve the relationship between context and magnified focus. Similar relationship preservation is achieved in [44] - 3D data is mapped to 4D space where the magnification is performed. Cowperthwaite [45] presents multiple approaches for focusing on areas within a 3D volume without encountering occlusion. Data filters and layout adjustments achieve visibility of targets. In contrast, our work tries to provide a unified tool that resembles tools used in the medical practice for data exploration and intervention planning. We do

not want to introduce distortion effects in the focus region since they might cause misinterpretation of position and shape.

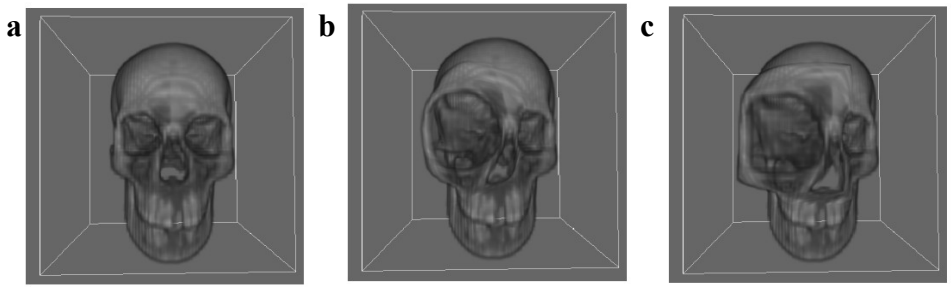

Figure 2-12: Distortion volumetric lenses. (**a**) Original volume. (**b**) Circular lens with cubic warping. (**c**) Square lens, linear warping. [41]

Novel rendering techniques within the focus region, and how to define focus, have also been explored. The focus region can be in the shape of an anatomical region of interest, thus only the important structure is visible [46]. A specific anatomy, for example blood vessels from the whole dataset as in [47], can also be used to define focus. The main disadvantages of such approaches are the need for dataset segmentation and the lack of an interactive way of defining different regions of interest at run time respectively. Focus-dependent regions used for displaying data by means of rendering modalities that are different from the ones in the focus area have been previously explored [48]. This work is different from ours since the rendering region we introduce does not depend on the shape of the focus, provides visibility of past focus regions, and changes over time. Animation has been used to predict the focus region and interactively change the rendering parameters [49]. However, our work uses animation to record past focus regions and to provide explicit cues about the spatial relationship between past and current focus.

A rendering technique that can display 2D, 3D and zoomed-in 3D focus regions has been presented in the work of Ropinski et al. [50]. This work does not use an interactive

tool to define focus and context. The zoomed areas are chosen by the user and a side window close to the selected area displays the region of interest. 2D rectangular slices, 3D volume visualization, or a zoomed volume visualization can be displayed. Even though this approach provides context which is important for our approach, the target is an area selected by the user and rendered on the side. Therefore, similar to the ExoVis callouts, this approach does not provide explicit spatial relationship between context and focus areas. While the user is shown a zoomed 3D volume rendering, there is no information about how this volume is related to the rest of the volume or to other focus areas rendered nearby in their own windows. Therefore, we find that this approach is not applicable when the direct relationship between focus and context has to be known.

Diepenbrock et al. [51] presented a combination of 3D volume rendering, overlaid axis-aligned slices and a focus + context tool that provides a detailed rendering to display what would be seen by a resection tool following a chosen surgical path. A separate side view that renders the target object together with surrounding structures is provided in order to avoid occlusion of the target structure. 2D and 3D renderings are provided in the first step of the planning interaction (Figure 2-13a), while the second step is aimed at surgeons planning the exact path to the target (Figure 2-13b). All necessary parts are however presented in two separate steps and the user has to switch the visualization back and forth between the two main steps. Furthermore, the exploratory surgical path planning tools are presented in separate windows, providing little explicit spatial perception between the overview window of the path to the target and the view of the target augmented with a slice. Therefore, this tool would cause mental burden on the physician, and it will not be applicable in situations when radiologists and surgeons

would want to work together on the same window but extract the information which is most relevant for their expertise.

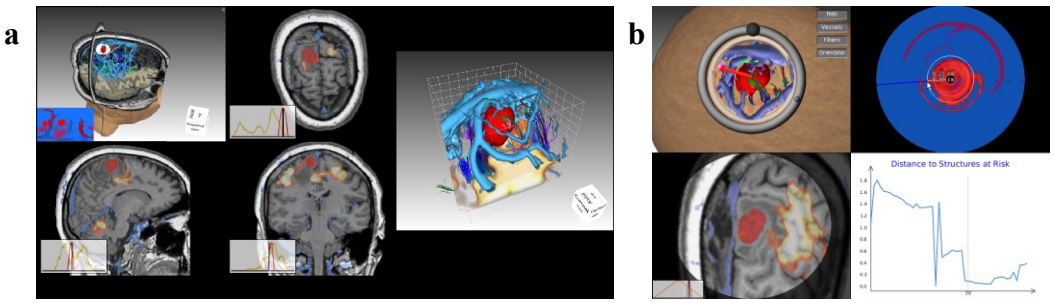

Figure 2-13: Focus + Context visualization with additional views for improved surgical planning. (**a**) 2D and 3D renderings for the selection of access path; (**b**) focus + context views that provide detailed information of the selected resection path. [51]

Similar surgical path views and rendering of a target structure surrounded by the rest of the volume is introduced in [52]. A region of interest is selected by the user after examining a multimodal volume rendering. Once the region of interest is selected by the user, the system computes the target position. Only then is the target displayed fully opaque with surrounding data from other datasets. To simulate a surgical path view, the proposed approach assumes that the incision path is a direct cylindrical path from the point of incision to the target. Depth perception has been enhanced by color adjustment, rendering of a coordinate system and a distance ring. This approach is aimed at surgeons only. Therefore the visualization technique would not be of interest to radiologists since no slices can be displayed. Moreover, the user has to first click on the target so that a region of interest is determined, and then the system renders surrounding structures in a way that ensures visibility. This rendering provides limited explicit spatial relationship understanding of the target and other structures from the context, and no inner structure exploration of the target can be achieved. Another disadvantage of this technique is the surgical path visualization which assumes that the access path is of cylindrical shape.

This assumption is arguable because there are many cases where objects along the planned path have to be avoided and the taken path can be of a very complex shape.

A technique most similar to the work in this dissertation has been presented by Burns et al. [53]. The authors augment transfer functions by means of an importance parameter such that different features defined by separate transfer functions can be identified as important or not. This is helpful since no segmentation of important features is required, and the important objects are always visible. A novel contextual cutaway region defined by two angle values is introduced. The two angles and the existence of focus and context regions ensure full visibility of objects of interest inside the overlay region, and a transition between 2D focus and 3D context is displayed in a transition region. Modifications of the angles and the overlay parameter achieve different rendering results as visible in Figure 2-14. Similarly to our work, this visualization combines 2D focus and 3D rendering and allows for inner structure exploration by defining focus and context regions. However, the work of Burns et al. requires the extraction of importance values from the transfer function. This limits the possible visible target objects to only those that are defined as highly opaque by the transfer function. The shape of the focus slice is predefined by the scanning device and the user has restricted control over the size of the focus region and how much surrounding context is present. Furthermore, the transition area has the shape of the rendered slice and it displays a combination of the focus and context datasets. This approach does not allow a combination of more than two datasets, thus cannot give additional spatial information cues about anatomical structures visible on other dataset modalities. If we assume that the target region is defined by the volume in front of the rendered slice and the two angles define the side clipping planes, there is

no way the user can render only a subsection of the volume in front of the slice. Instead, the clipped target volume always ends at the focus slice. Furthermore, no surgical intervention path planning can be achieved by the presented visualization.

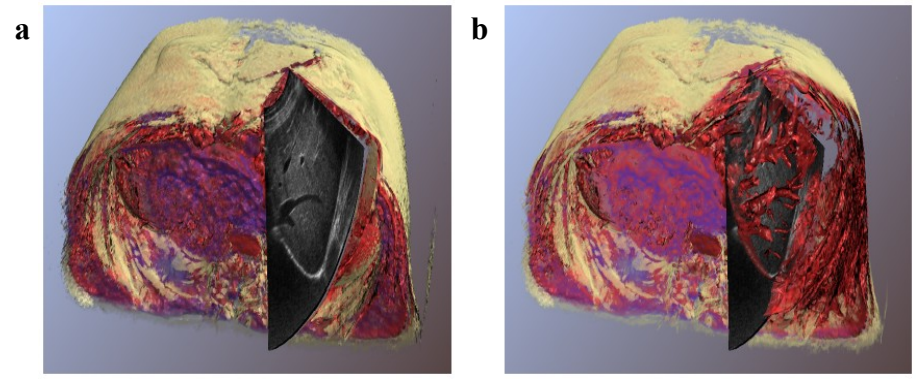

Figure 2-14: Longitudinal ultrasound (US) image rendered as target within an MRI dataset. (**a**) Rendering of the target volume with no transition and overlay regions. (**b**) Overlay and transition regions allow the target volume to be rendered so that it augments the US image.

Luo et al. [54] present an improvement of the focus + context metaphor by turning the magic lens into a focal probe in order to improve the context preservation and avoid cluttering by revealing more focus features. The focal probe acts similarly to the overlay region defined in [53], but it does not provide any possibility for 2D cross section rendering. It also clips the volume in front of the focal point and renders everything all the way to the back of the probe. Therefore, there is little flexibility about the size of volume that is displayed inside the probe. Furthermore, it serves as a tool used mostly for data exploration, and would not be beneficial for intervention planning.

Magic lenses have also been applied to 4D data and 3D-flow data. Fuhrmann and Groeller [55] use a cube-like magic lens to display filtered 3D flow data. This concept has been further extended by Gasteiger et al. [56] to allow the real-time exploration of 4D PC-MRI and CFD simulation data, where the focus-providing region is a 2.5D lens.

2.3 Raycasting approaches in volume visualizations

Our raycasting approach is based on the work presented by Kirmizibayrak [10] where a single lens is used to show multimodal datasets. Different dataset modalities with their own transfer functions can be assigned to every region, making this tool extremely useful when physicians work with images of high complexity. In their approach, a pre-processing step is used to extract the front and back depth layers of the volume and the lens. The GPU vertex and fragment shaders break the ray for each pixel into three sub-regions: 1. volume before the lens, 2.volume within the lens, and 3. volume after the lens. The same raycasting method is used to paint the volume (Figure 2-15) by utilizing an additional shader that pushes back the context polygonal mesh used for the start of the rays. The change of mesh shape depends on the position and shape of the magic lens. In this approach, areas from different data scans can be visualized on the same volume visualization. This raycasting approach, however, does not work well when the lens is of concave shape or if there are multiple lenses used when material is carved out, which is one of the goals of our proposed visualization framework.

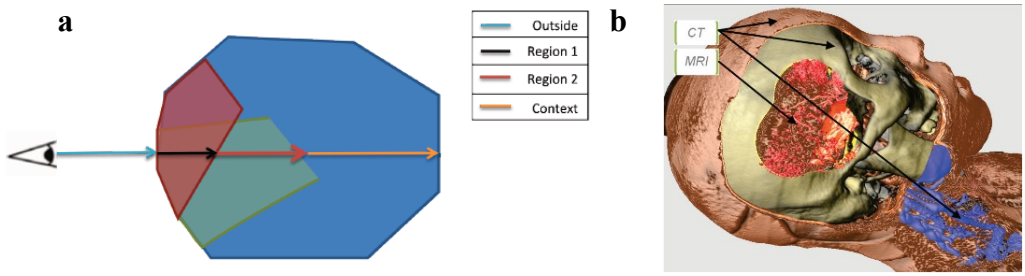

Figure 2-15: Focus + Context for volume painting. (**a**) Diagram explaining the computation of the ray positions in the volume. (**b**) The volume is painted using three different lens volume modalities. [10]

Multivolume raycasting has gained significant popularity in the graphics community since graphics cards have become powerful enough to handle the necessary computations. Rossler et al. [57] have proposed a new method of dynamically generating

shaders to raycast multiple intersecting volumes. The intersecting volumes can be assigned the same or different datasets. The ray computation and an example of the approach applied to a dataset are shown in Figure 2-16. Their approach allows the construction of an abstract render graph that determines what shaders are used for specific volume areas. The advantage of this work is the reduction from a three-pass rendering to a two-stage rendering algorithm by employing depth peeling approaches. Plate et al. [58] developed an octree-based algorithm to handle raycasting of arbitrary intersecting volumes. Their work has been developed in the context of overlapping volume exploration using lenses, which is similar to our approach. In this work, if a lens intersects multiple volumes, an octree that stores information about the boundaries for each subvolume is created. Depth peeling is used to sort all sub-lens volumes represented in the octree in order to correctly visualize the overlapping volumes. The application of multiple lenses has been further explored in the work of Borst et al. [59]. Their work requires no view-dependent generation of textures necessary for the raycasting algorithm, and can reuse cached results. The composition of lenses and the number of volumes lying within the lenses are determined at every time step using bitwise operations, after which an in/out test for the lenses determines the sampled volumes, providing significant time speedup compared to any previous methods.

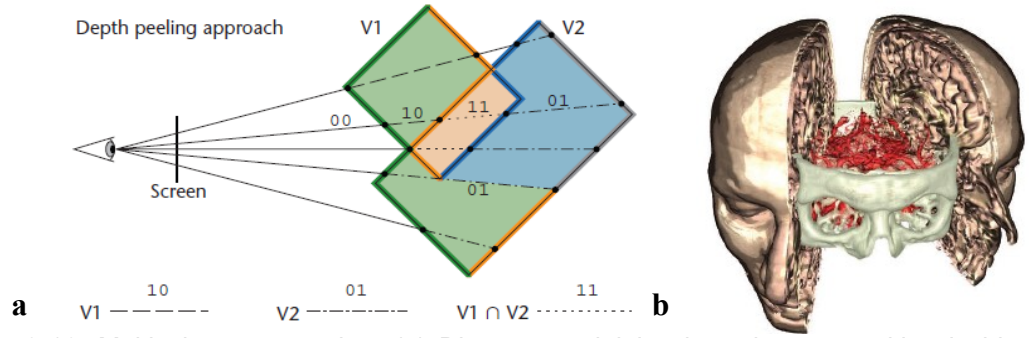

Figure 2-16: Multivolume raycasting. (**a**) Diagram explaining how the ray position inside each volume is computed. (**b**) Interactive multivolume raycasting implemented for moving the two head cuts and ensuring skull visibility. [57]

To compare the performance of our visualizations with the work described in Chapter 2, we have summarized the reported frame rates of various visualizations in Table 2-1.

Table 2-1: Rendering speeds of visualizations presented in Chapter 2.

Paper	Reported frame rate		Paper	Reported frame rate
[6]	2 - 9 fps		[41]	4 - 6fps
[7]	2 - 28 fps		[42]	3-14fps
[8]	4 - 21 fps		[50]	11 - 33fps
[10]	17 - 30 fps		[52]	9 - 24 fps
[26]	50 - 100 ms		[57]	11- 31 fps
[38]	4 - 20 fps		[58]	5 - 30 fps
[39]	5 - 21 fps		[59]	3 - 90 fps

While focus + context provides an effective tool for medical dataset exploration, the volume displayed within the lens is always the whole 3D volume that falls within the boundaries of the lens tool - polygonal or any arbitrary convex or concave shape. There have been cases where the volume is clipped at the front, however no technique also provides a back clipping plane for the focus region. Occlusion of less important anatomical features can hide the object of interest and there is little detailed information similar to the slice visualization that can be extracted. Our work attempts a solution to the occlusion problem and also provides detailed slice information which the medical community is familiar with. As shown, the focus + context visualization technique has

been applied to 2D, 3D and 4D data. However, we are unaware of any focus + context techniques that utilize time to define the size and shape of visible focus data that is of interest for the data exploration and diagnosis, while still preserving context data.

2.4 Perception improvements in volume rendering

2.4.1 Information retention

In our work we employ studies about information retention and the extent of periphery vision that is helpful for a user in perceiving spatial relationship information to further support explicit spatial and shape perception.

Researchers in perception and psychological cognition have paid extensive attention to how much information is retained while the test subject changes attention area during an active top-down attention exploration task. The central (focus) vision area is where the user's gaze is fixed for a certain amount of time in order to perceive the most information about the observed region [60]. Once the visual attention area is moved from one place to another, the user relies on limited implicit memory to deduce relationship between the current focus and the past focus areas. However, implicit memory about objects in past focus areas is soon replaced by newer sensory data, which would impair any future necessary cognitive conclusions about the objects' relationship.

On the other hand, when we observe a scene and have an area of visual attention, contextual (gist) information and relationship cues are obtained through our peripheral vision. Peripheral vision provides immediate important spatial cues between the focus area and the rest of the scene due to the brain's ability to recognize low spatial frequency information [60]. The peripheral vision area serves a major role in understanding how the

attention area is related to the rest of the scene. It has been proven that peripheral vision cues can be retained in memory for days, and thus provide much stronger spatial relationship understanding of the explored scenes [61].

Current focus + context visualizations lack the ability to provide any relationship cues between the current focus area and the focus areas that have been explored a few seconds earlier. Such a relationship is of importance when a physician wants to limit the size of the focus area so that enough context is visualized, but is expected to infer distances between structures that are visible only in the focus dataset and do not lie within the lens geometry. Other cases where such cues might be of importance are during surgical planning when a physician moves the magic lens to see the target region, and also needs to see the objects lying along the path taken to reach the target.

2.4.2 Depth perception

3D volumetric renderings on a 2D screen often suffer from depth understanding problems. Users have to interact with the object and rely on motion parallax to perceive what objects are in front. Much research has been devoted to this problem, and a large range of solutions have been proposed.

One way of improving depth perception is by applying depth of field effects for the volume rendering. The target object is in focus, while objects in front or further away are blurred with a blur coefficient proportional to the distance to the target [62]. An evaluation of this approach has been performed by a means of a user study. The highest accuracy of depth perception was perceived when the target object was at the front and the visualization was done using perspective projection [63].

In non-photorealistic rendering, depth perception has been improved using multiple techniques like tone shading, silhouettes and color depth cueing. Depth perception studies have shown that closer sections of an object are perceived to be in warmer tones than points in shadow or further away. [64] have introduced change of color hue based on the distance of an object's point with respect to the eye position. Closer objects can be painted in warmer colors, for example red, while further objects are in bluish hues. Another used method is silhouette rendering, which displays the objects' silhouettes and the user infers depth based on occlusion of the rendered contours. [65] uses the techniques described above, implementing them in real time by GPU computations and applying them to medical datasets that can benefit from depth perception improvements. An example of tone adjustment and the depth perception improvement that it achieves is pictured in Figure 2-17. Similar techniques like depth cuing by contrast reduction, depth color blending between source color and background, blurring and line fading have been used for illustrative depth improvement in [66]. Halos are another proposed technique [67, 68] for depth perception improvement.

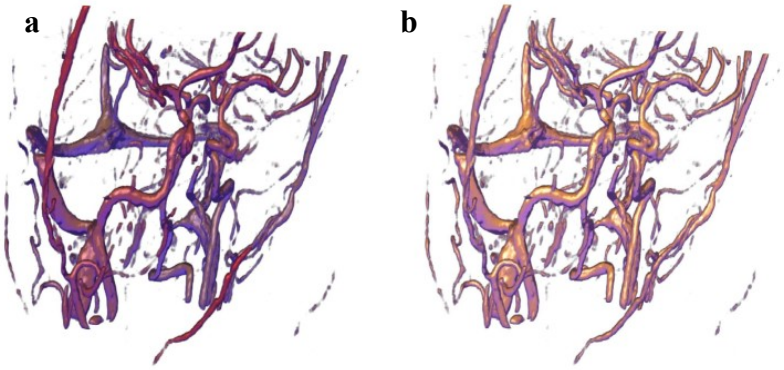

Figure 2-17: Depth perception improvements. Depth-based tone adjustment (**a**) results in an improved depth perception of the vascular structure, compared to the same dataset without such improvement (**b**). [65]

The above-mentioned techniques of color correction have been further extended by [69] where edge enhancement and two types of color correction – chromadepth and pseudo chromadepth (shown in Figure 2-18) have been implemented for the rendering of blood vessels. It was shown that pseudo chromadepth gives the most depth cues without burdening the user to remember different color ranges, and is therefore used in the work presented in this dissertation.

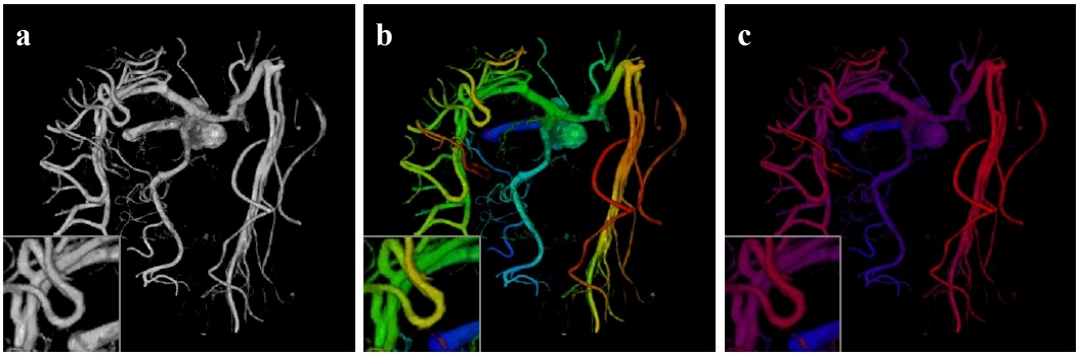

Figure 2-18: Depth perception improvements using hue adjustments. (**a**) Original structure. (**b**) Chromadepth coloring. (**c**) Pseudo-chromadepth coloring

Chapter 3: Generalized temporal focus + context framework

As described in Chapter 2, there are various ways of exploring datasets. All existing visualization approaches provide 2D slices, 3D volume renderings, or a combination of the two that can also be extended to 4D if we consider temporal changes in the visualization or time-dependent datasets. Medical visualizations are often developed to help with specific medical tasks in mind, and thus there is no work that provides a theoretical comparison between the various approaches that shows how they differ when used in the same application and which visualization is most suitable for certain tasks. Therefore, we conceptually explored existing 2D and 3D visualizations and the user interaction with a slice or a volume. As a result, we developed a generalized temporal framework that can be used to classify existing visualizations and to investigate new visualizations that have not been applied to medical data before [70]. Such a framework can be successfully incorporated within any visualization system so that it provides a wide range of rendering approaches that can be easily adjusted or switched, and would therefore speed up the diagnosis and procedure planning for individual or multiple users.

3.1 Generalized framework overview

We chose to employ the magic lens focus + context paradigm and generalize it into a temporal framework that can display various commonly used dataset exploration methods, some of which have been presented in Chapter 2. The developed framework is based on the observation that any combination of 2D and 3D visualizations can be generalized as consisting of two 3D geometric regions of any shape, for example two

cuboids as shown on Figure 3-1a. The large black cuboid represents the main (context) dataset, and the red cuboid is the focus geometry which will be adjusted so that it resembles 2D, 3D or focus + context visualizations. For example, a volume visualization augmented by a 2D slice can be represented by having the dimensions of the black cuboid equal the context dataset, while the red (focus) cuboid has dimensions equal to a single slice from the focus dataset as shown in Figure 3-1b. The red cuboid is moved along the z-direction to reflect the slice position. If the x and y dimensions of the focus cuboid are decreased as in Figure 3-1c, then the displayed slice is a sub-section of the original dataset slice, and it is surrounded by 3D volume similar to [53]. In both of these cases, we can allow the depth of the focus cuboid to be larger than the thickness of a single slice, resulting in a 3D volume rendering inside the focus region. Adjusting all three dimensions of the red cuboid to match the dimensions of the original focus dataset, the user achieves a 3D rendering of the whole focus volume as in Figure 3-1d. The 3D focus + context visualizations described in Chapter 2 can be represented as adjusting the size of the red cuboid to be less than the x, y, and z dimensions of the original dataset, but greater than the thickness of a single slice as in Figure 3-1e. Having such a generalized framework allows us to represent most of the already existing approaches as summarized in Table 3-1 so that a user can explicitly compare techniques and select the one most appropriate for the specific medical case. Providing framework-based visualizations for every user, we can support the visualization requirements of both radiologists and surgeons who need to discuss a case together.

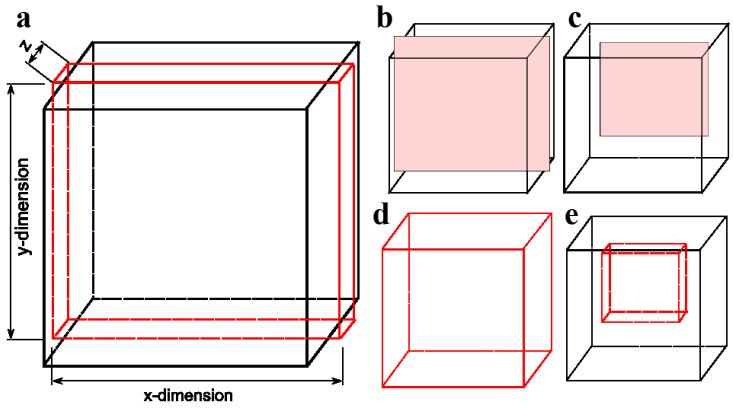

Figure 3-1. Conceptual representation of the generalized focus + context framework. Any combination of 2D and 3D volumes can be viewed as having two cuboids of different size as in (**a**). Changing the size of the red cuboid results in: (**b**) 2D slice visualization with surrounding 3D context. (**c**) Slice with surrounding 3D context. (**d**) 3D rendering of the focus dataset only. (**e**) A focus + context visualization.

Table 3-1. Existing visualization techniques explained in terms of the generalized temporal focus + context framework.

Visualization	**Representation using generalized temporal focus + context framework**	
	Context	**Focus**
3D volume	Cube with dimensions equal to the whole context dataset	None
3D volume	None	Cube with dimensions equal to the whole focus dataset
Clipped 3D volume	3D geometry with dimensions smaller than the context dataset dimensions	None
Clipped 3D volume	None	3D geometry with dimensions smaller than the focus dataset dimensions
2D slice	Cube with dimensions equal to a single slice from the context dataset. The slice position can change over time.	None
2D slice	None	Cube with dimensions equal to a single slice from the focus dataset. The slice position can change over time.
3D with 2D slice	Cube with context dataset dimensions	Cube with size equal to a single slice from the focus dataset. The slice position can change over time.
3D with 2D slice	Cube with size equal to a single slice from the context dataset. The slice position can change over time.	Cube with focus dataset dimensions.
Focus + Context	Cube with context dataset dimensions	3D geometry with reduced focus dataset dimensions but not equal to a single slice. The focus position or size can change over time.

Table 3-1 (cont.) Existing visualization techniques explained in terms of the generalized temporal focus + context framework.

Visualization	Representation using generalized temporal focus + context framework	
	Context	**Focus**
4D Focus + Context	Cube with context dataset dimensions	3D geometry with reduced focus dataset dimensions but not equal to a single slice. The dataset changes over time.

Further consideration of the focus and context regions from the introduced generalized framework and their time-dependent interaction allows for the combination of 2D and 3D renderings in ways not previously used in medical exploratory visualizations. While a single 2D slice or a 3D volume can be rendered inside the lens, we developed five new methods of visualizing multimodal datasets that have not been previously applied to magic lens visualizations, as described in Table 3-2. These visualizations provide explicit information about the shape, structure and position of multiple dataset landmarks. In all of these techniques except the last one (*Inner sub-volume exploration*) the context is a 3D volume with dimensions equal to the context dataset. *Inner sub-volume exploration* can have a context region of any size, but this data region is not rendered.

In the first new technique - *3D Focus (with 2D rendering) and 3D Context*, a lens of a three-dimensional geometry of dimensions smaller than the focus dataset is used to probe the focus dataset and explore every slice that falls within the lens region. The slices are scrolled through by the user. In the second approach - *2D+3D Focus and 3D Context*, we allow the simultaneous rendering of 2D and 3D data inside the lens so that the inner structure of target volumes is explored without modifying the used transfer function as proposed in other work on inner structure exploration. We utilize a time-dependent

visualization - the user is asked to scroll through slices he thinks are important for the volume visualization, and thus define a 3D sub-volume inside the lens geometry that will be rendered. The resulting volume is augmented by rendering the back-most visible 2D slice. The next new technique (*focus-driven context*) helps users create an improved spatial relationship understanding between past and the current lens positions. A time-dependent focus visualization renders a trace of past lens positions by rendering the 3D volume inside the trace region. The user can now explicitly relate currently and previously viewed volumetric structures without having to memorize details about areas no longer in the focus. The newly defined trace area can be viewed as a hierarchy of context areas and thus it creates three regions where different datasets, transfer functions, or rendering methods can be assigned – for example 2D, 3D or 3D sub-volume.

The user is allowed to save multiple focus regions that have been explored, therefore creating a focus region of user-dependent shape and size. If the rendering inside these saved positions is turned off, a volume sculpting visualization for planning of incision trajectories can be achieved - *arbitrary focus region definition*. Unlike previous surgical path planning systems, our framework supports a very accurate exploration of possible incision paths. When the user needs to see only the region that falls inside the lens geometry, *inner sub-volume exploration* visualization is achieved.

Table 3-2. Proposed new visualization techniques defined by the generalized temporal focus + context framework.

Visualization	**Representation using generalized temporal focus + context framework**	
	Context	**Focus**
3D Focus (with 2D rendering) and 3D Context	Cube with dimensions equal to the whole context dataset.	3D geometry of dimensions smaller than the focus dataset. Show only one slice at a time.

Table 3-2 (cont.) Proposed new visualization techniques defined by the generalized temporal focus + context framework.

2D+3D Focus and 3D Context	Cube with dimensions equal to the whole context dataset.	Cube with dimensions smaller than the dataset. 3D sub-volume shown inside the lens. Back slice of sub-volume shown.
Focus-driven context	1. Context cube with dimensions equal to the whole dataset; 2. New 3D region defined by past positions of the focus 3D geometry.	3D geometry of dimensions smaller than the focus dataset. 3D volume, 2D slice or a 2D+3D rendering as explained above can be shown in this region.
Arbitrary focus region definition	Cube with dimensions equal to the whole context dataset.	Arbitrary shape defined by moving the lens geometry in time, volume inside this region is shown or skipped.
Inner sub-volume exploration	Cube with any dimension. This volume is skipped.	3D geometry that displays 2D or 3D data inside it. Focus position determined by the user.

Therefore, a focus + context visualization can be viewed as an overview and detail visualization framework that can simultaneously achieve standard 3D volume rendering, a 2D slice rendering, 3D sub-volume rendering or a hierarchy of focus areas. By having a generalized framework we can define new ways of rendering focus and context so that we can provide explicit spatial relationship between multimodal datasets rendered as 2D or 3D, as well as provide visualizations that help in surgical planning or medical training by seeing only relevant regions. All proposed visualizations can be combined in any way to provide multiple views that can be used by physicians during team meetings or for accurate intervention planning.

3.2 3D focus showing 2D slices

3.2.1 Overview of technique

The lack of contextual information and spatial and structural understanding of objects visible on slices has driven our work of combining 2D slice scrolling with a magic lens

visualization to significantly improve the spatial relationship perception between a volumetric rendering and slice images. We extended the possible visualization modalities inside a volumetric focus region to include rendering of 2D data. Context is kept as a 3D rendering so that it provides volumetric clues regarding the 2D images inside the focus.

Unlike previous techniques where the focus is a single slice moved in three-dimensional space, we introduce a unification of 2D and 3D techniques inside the lens region. Focus is still in the shape of a 3D lens geometry but we render only one slice at a time inside the lens. Therefore, we can still define target structures in terms of a 3D geometry, but restrict their visualization to 2D. This technique can be used in the exploration of volumetric datasets to see the original data that comprises the selected target volume slice by slice. The visible slice is oriented in an axis-aligned way if the lens is positioned perpendicular to the dataset, or it is in any arbitrary orientation depending on the orientation of the lens, thus supporting the visualization needs of different groups of medical specialists.

Moreover, the advantage of having a generalized focus + context framework to define 2D slices visible inside a 3D region is that we can easily accommodate a necessity of exploring larger or smaller sections of a 2D slice while ensuring visibility of the 3D context. For example, the need to see a larger slice while there is little context rendered around is achieved by increasing the dimensions of the lens. The bigger the lens is, the more slice details are visible and less context is rendered. Having multiple 3D shapes for the lens geometry, we provide easy adjustments of the shape of the visible slice - they can be in the shape of a circle, oval, square as in most existing medical visualizations, or others depending on the loaded lens geometry and the orientation of the lens.

3.2.2 Dataset exploration application

Physicians often have to explore a combination of datasets - MRI, CT, US or others to understand how various anatomical objects visible on different modality scans are spatially related. Once the target feature is found, a physician might require a detailed view of the identified structure of interest. While 3D volume visualizations are excellent for understanding shape, occlusion can cause important aspects of the anatomy to be hidden by the object's surface or other structures. For example, when exploring a brain tumor, a physician would like to know not only the exact position of the tumor and its shape, but also its composition and if there are any structural differences between the various areas within the tumor. This information is often not visible in 3D visualizations due to occlusion by the surface of the structure, and physicians go back to exploring individual slices to get a better understanding of the tumor.

Using our 2D rendering inside a volumetric lens, once the user has identified a structure of interest using the standard magic lens tool (Figure 3-2a), he can switch the visualization to a 2D slice rendering (Figure 3-2b). Thus the user can explore the slices that make up the volume previously rendered in the lens geometry. Our approach is driven by the series of tasks that physicians often perform - explore the whole volume and find the target area; then, concentrate on the selected region and explore it paying attention to detail by looking at slices.

The position of the displayed slice varies between the front (Figure 3-2b) and back (Figure 3-2c) ends of the lens. Visualizing a slice deeper within the volume would require the lens to be moved along the z-direction or to increase the size of the lens. The user can scroll through the slices using the keyboard or with gestures using a tracking device such

as Microsoft Kinect®. The lens position and orientation in 3D space is controlled using the mouse or a 3D tracker - for example a magnetic tracker (EM tracker) or Kinect. This process is successfully executed using a standard focus + context visualization as described in Kirmizibayrak [10] and allows interactive movement of the magic lens in 6 DOF.

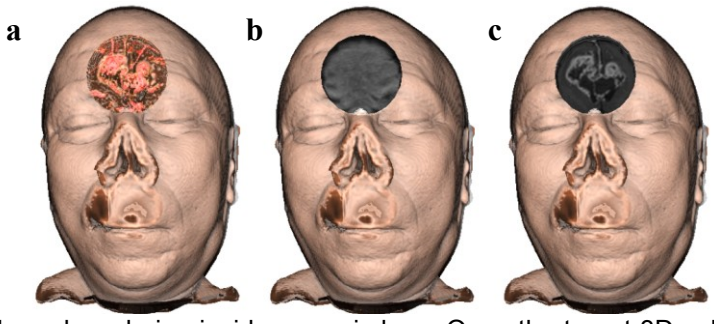

Figure 3-2: Slice-based rendering inside a magic lens. Once the target 3D volume is found using a magic lens (**a**), 2D rendering is enabled to show any slice between the front-most slice inside the lens (**b**) and the back-most slice (**c**).

The extension of the focus + context visualization to show slices within the focus region aims at helping radiologists improve their understanding of the internal structure of the target volume by eliminating occlusion common for 3D visualizations. Radiologists can use this technique to scroll through slices and examine them, while the 3D context rendered around the slice gives them more information about surrounding structures visible on another dataset modality and how the structures are related. Surgeons can use the same visualization to explore structures along the planned resection path by seeing slices at arbitrary orientation. It also provides a slice visualization tool that physicians are comfortable with and often use when they have to clearly see objects.

3.3 2D+3D Focus and 3D Context

3.3.1 Overview of technique

Existing focus + context visualization techniques allow users to limit the rendered volumetric regions by means of clipping planes as described in Chapter 2. When the focus is a 2D slice, a front clipping plane and the 2D slice define the size of the rendered volume. When the target is a 3D volume, clipping planes or the position of the lens focal point determines what region in front of the target is skipped. However, no technique explores the possibility of having a large 3D focus region and the need to render different sections of it to provide an improved structural understanding while preserving information about the overall size and position of the target structure. Using our generalized temporal framework, we can address this lack of interactive visualization by redefining what segments of the 3D structures inside the magic lens are rendered.

This introduced extension is driven by the commonly used and proven to be helpful visualization of a slice and its surrounding 3D volume. In this type of visualizations there might be so much information before the slice that structures in proximity to the slice of interest become occluded. To overcome this disadvantage, we extend the slice visualization described in Chapter 3.2 by saving the positions of the viewed slices over a period of time. While the slices that make up the target volume are scrolled through, the position of the oldest and most recently viewed slices over a predefined time span are used for the definition of two virtual cutting planes that outline a 3D focus sub-region - Figure 3-3. The slice positions are saved in an array of size N using the first-in first-out (FIFO) principle, where N is proportional to a user-defined time span.

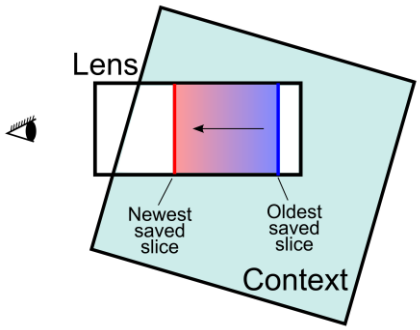

Figure 3-3: Diagram showing the area rendered as 3D sub-volume. The oldest saved slice (in blue) and the newest saved slice (red) determine the region between which the 3D rendering inside the lens is enabled.

This interaction technique combines volume visualization and slice visualization within the lens in a way that provides a better understanding of the internal structure of the focus volume. Our visualization is more flexible than existing work where a single clipping plane is used to define what is being removed while everything behind this plane is rendered in 3D. We do not fix the position of the two slices, which in our visualization can also be thought of as clipping planes. Therefore, our approach can achieve a rendering of the whole volume within the lens. It can also be successfully used to display the volume only between two or several slices, providing a clear visibility of only selected structures of interest.

To provide important details that might not be captured by the current transfer function but could greatly contribute to the user's volume structural understanding, the proposed technique displays the backmost slice visible at every frame. If the slice position is moved forward, the oldest saved slice will be displayed. If the slice is moved deeper inside the lens, the most recently saved slice will be displayed. When the user stops changing the slice position, the 3D volume visualization continues as an animation where the visible 3D volume gets smaller since the distance between the current and oldest slice positions over a predefined time span decreases until the two slice positions

are identical and only one slice is visible, becoming the technique described in Chapter 3.2.

3.3.2 Dataset exploration application

Surgeons and radiologists have extensive training in understanding slices and the volume they represent. They are also familiar with and widely use systems that allow for axis-aligned or arbitrary slice orientation display. However, slice visualizations have one disadvantage - they require physicians to mentally reconstruct the focus 3D volume while scrolling through the slices. If slices are augmented by a 3D volume rendering and clipping of occluding structures is supported, more information about the three-dimensional structure of the volume can be obtained. The improved volume understanding is often of significant help to clinicians who have to understand the precise depth spatial relationship between objects. However, to our knowledge, no existing magic-lens based visualization provides a flexible definition of a sub-focus region within the target volumetric structure with the front and back clipping planes being the explored slices. Thus, the newly introduced paradigm can be used to view any subsection of the 3D focus volume in order to provide a better understanding of the inner structure of the target anatomy.

Figure 3-4a shows a snapshot of a rendering of a brain tumor MRI dataset. The context area is used to provide the physician with explicit cues about the relative position of the tumor with respect to the patient's head. For example, Figure 3-2a shows the whole volume that falls within the lens, and Figure 3-2c displays one of the slices that reveal the internal structure of the tumor. However, the shape of the tumor and its composition right

before or right behind this slice are not conveyed. To correct for this, Figure 3-4a displays a sub-volume rendering of the 3D tumor geometry defined by the oldest and the current slice positions as shown on the left side of the image. In this example the slice is moved deeper inside the volume and the current slice position is visible in the back. Similar visualizations can be achieved in displaying the sinuses rendered from a CT dataset as in Figure 3-4b and c. Each sub-volume spans a different section of the lens geometry depending on the user interaction.

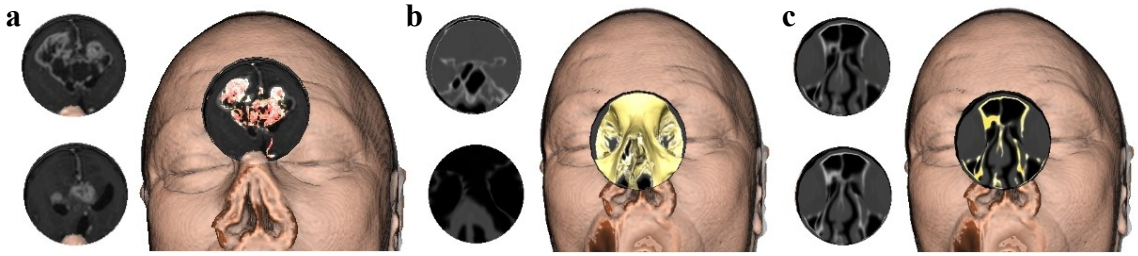

Figure 3-4: 3D sub-volume rendering inside the lens. The volume between the two slices shown on the left of each image is rendered using a predefined transfer function. (**a**) Rendering of a brain tumor from an MRI. (**b**) Rendering of a CT scan with the nose and sinuses visible. (**c**) Rendering of the sinuses volume positioned between two consecutive slices.

The rendering of a 3D sub-volume can help both radiologists and surgeons achieve their exploratory goals. While a radiologist examines the rendered 2D slice, he can relate the slice position to the context volume or can visually examine the 3D volume right before or right after the slice of interest. Radiologists can also use this visualization to remove unnecessary objects and achieve clear visibility of the target structure in both 2D and 3D. Surgeons can use this rendering technique to establish a better sub-volume structural understanding of both the target anatomy and objects lying along the planned resection path without worrying that important areas are occluded. A detailed view of the explored sub-volumes is provided by the 2D slices augmenting the focus region.

3.3.3 Depth perception improvements of proposed technique

An interaction session with the 3D sub-volume visualization can lead to a lack of understanding of the exact depth and size of the displayed sub-volume. Since our technique can render only a very small subsection of the volume, a lot of space might be skipped before the rays from Figure 3-3 start accumulating color and opacity, and it is not guaranteed that the end of the rendered sub-volume coincides with the back of the lens as in other approaches.

Even though motion parallax provides one of the easiest to understand depth cues, we explored other depth improvement techniques that do not require movement of the explored geometry as this could cause confusion or inaccurate procedure planning especially in surgical simulations. As explained in Chapter 2.4.2, perception research shows that people often associate nearby objects with warmer colors (red, orange), while further objects are perceived to be more blue. Therefore, color-coding can be applied to the rendered sub-volume, and thus we extended our 3D sub-volume visualization by including chromadepth color coding [69]. In cases where depth color coding is included, color change happens only in the 3D sub-volume rendered. The slice displayed at the back of the volume is left as grayscale. To ensure visibility of depth color-coding without ignoring the original color extracted from the transfer function, we do not assign each sample along the ray colors ranging from red to blue as in [69]. Instead, the color obtained from the transfer function at every sample point is multiplied by a color-adjusting value which starts as red and transitions towards the blue. The speed with which the color-adjusting value is transitioned from red to blue depends on the distance between the front and back slices and the number of samples taken along each ray.

Therefore, no matter how deep the rendered volume is, our coloring scheme ensures that the closer parts will be rendered in red, while parts further away will be blue as seen in Figure 3-5. Figure 3-5a shows pseudo-chormadepth coloring of the whole volume inside the lens, therefore blood vessels at the front are red, while the tumor near the center of the brain is in blue hues. If the rendered sub-volume is much smaller than the lens size, as in Figure 3-5b, the front-most points will be red, while the back of the sub-region is blue.

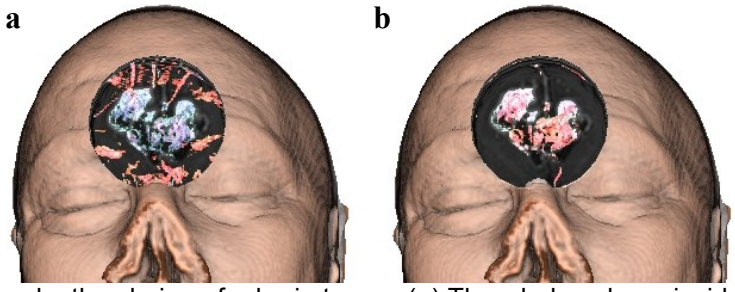

Figure 3-5: Chromadepth coloring of a brain tumor. (**a**) The whole volume inside the lens is color-adjusted. (**b**) Only the brain tumor is rendered and therefore has color adjustment.

The proposed 3D sub-volume technique, similarly to the other visualizations defined by the generalized focus + context framework, uses an interactive lens to define a region of interest. However, the color-coded volumetric rendering inside the focus region might not be enough to infer how deep the lens goes inside the context volume and what the depth of the rendered volume is. Therefore, any of the visualizations considered within our generalized temporal focus + context framework can benefit from a depth visualization tool. This need becomes even more apparent in the 3D sub-volume rendering technique since the rendered volume can be anywhere within the lens geometry and it is of varying depth. Therefore, we extended the distance ring tool presented in [52] which shows the depth of a selected organ with respect to the whole dataset, where no focus tool is used. A modified pie-like depth visualization tool (Figure 3-6a) can provide explicit information about the depth position of the lens when used in any of the

visualizations defined by our framework. Moreover, it can be extended to show what portion of the lens is spanned by the 3D sub-volume at every time step of the interaction. This tool's purpose is to give approximate depth information about the focus geometry and the rendered volume inside it in order to help one understand where the rendered structures are with respect to the lens geometry.

We assume that the circumference of the depth tool corresponds to the full depth of the context volume. Depending on the lens size, the focus volume can be equal or less than the depth of the context. The lens depth as visible in Figure 3-6 is displayed in dark grey and shows the portion of the whole volume that falls within the lens geometry (on Figure 3-6 the lens is 30% of the whole volume depth). Within the lens, the two slices define the rendered sub-volume. Its depth is depicted in green and the position of the green part starts and ends at the corresponding slice locations within the lens. While the user changes the slice positions, the position and length of the green section keeps changing in order to provide real-time depth position information.

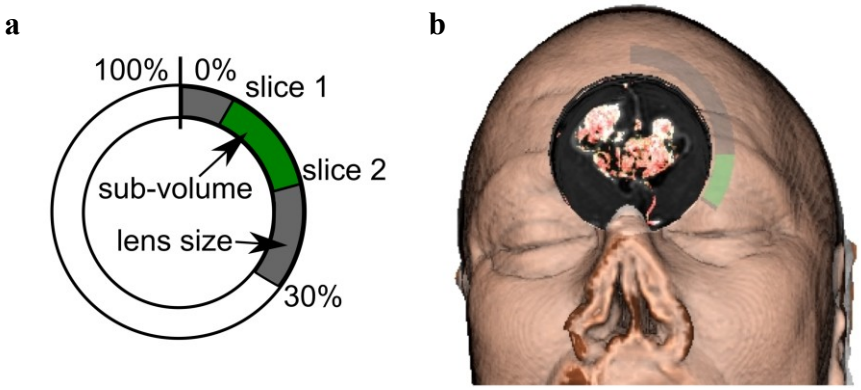

Figure 3-6: A depth visualization tool used for 3D sub-volume rendering. (**a**) The tool shows the length of the lens (in grey) with respect to context and the volume spanned between slice 1 and slice 2 (green). (**b**) The tool applied in rendering an MRI dataset.

3.4 Temporal focus-driven context

3.4.1 Overview of technique

Current magic lens focus + context visualizations support limited explicit spatial relationship between current and past focus areas. Structures that are considered important are replaced by context as soon as the lens tool is moved away. Because the user cannot see structures from successive focus regions at the same time, he or she has to rely on memory to compare them. Therefore, we explored ways of using our generalized focus + context framework to create an additional area for contextual visualization which alleviates the mental burden of implicitly inferring the distance and shape relationship between multiple dataset landmarks. Using the framework to define a new rendering region allows us to also incorporate improvements in shape and relationship understanding for a wide range of existing visualizations. Past focus positions over a user predefined time span define a complex new region used to render various datasets and modalities and to provide explicit spatial relationship. The new rendering region, called time-dependent focus-driven context, can be perceived using the user's peripheral vision and its goal is to provide as much multimodal contextual information as possible without removing the important structures rendered in the magic lens area. Conceptually, we can think of this new area as a subsection of the framework's context region, but its shape is defined by the temporal lens positions.

To achieve correct position calculations of the focus-driven context, we employ a FIFO saving scheme of the most recent n focus position while the lens is being moved within the context. n is calculated based on a user-specified value measured in seconds which is translated into how many positions should be saved. Regions that have been

considered focus-driven context for more than the user-defined time span will be replaced by more recent focus positions. This approach creates a geometry that changes its shape over time and follows the focus region as a trace. Our animation-driven approach supports pausing so that the focus-driven context region can be fixed until further user interaction. Furthermore, a toggle switch is provided in order to turn on/off the display of focus-driven context in cases where the user would like to clearly see context structures lying within this region.

The new region defined by our generalized framework allows us to introduce a multi-level hierarchy of context volumes that supports explicit relationship between structures from multiple multimodal datasets. The new focus-driven context visualization paradigm shown in the volume rendering in Figure 3-7a can be viewed as a hierarchy of context regions where the context area from the generalized framework is now split into context-only and temporal focus-driven context areas as shown diagrammatically in Figure 3-7b. Context-only shows the same data as the context from the hierarchical level above. The focus-driven context area defined by the user's interaction with the lens can show any combinations of user-selected datasets and visualization dimensionality. In the example in Figure 3-7a, the new region shows both context-only and lens data merged together. Combining context-only and focus-driven context gives us the overall context for the current focus region. However, since this temporal region is aimed to provide more spatial cues, it is not restricted to having the same rendering parameters as focus or context-only. Thus, this hierarchy of context regions further benefits the dataset spatial cognition by allowing relationship inference between focus and context-only, focus and focus-driven context, and context-only and focus-driven context. The possible rendering

techniques in each of the three regions are user-defined, thus they can be viewed as areas that provide horizontal relationship between the multiple regions rather than the two-fold relationship provided by the standard focus + context paradigm.

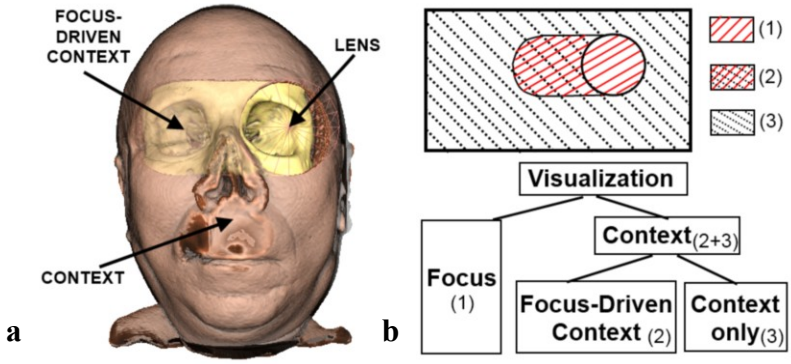

Figure 3-7: An overview of the focus-driven context paradigm. (**a**) CT-MRI data visualization using focus + context + focus-driven context with labeled areas. (**b**) A graph showing the two-level context hierarchy created by the focus-driven context.

The scope of the introduced focus-driven context is to provide additional information that can help the user understand better the relationship between distant anatomical or functional structures that cannot fit inside the lens. Therefore, the trace is rendered using a weighted alpha blending between context and focus-driven context so that we can assign more importance to areas closer to the current magic lens position that are also expected to be the most recently saved lens positions. Regions far from the current position are most likely to be the oldest lens positions stored. Thus, they would still be visible but will be faded out to a bigger extent in order to allow more visible context, as further explained in Chapter 4.

3.4.2. Dataset exploration application

Having three separate regions, the user can assign unique dataset and rendering modalities to each region in order to achieve optimal spatial relationship understanding

between landmarks from the explored datasets. The focus and focus-driven context areas can be assigned the same rendering modality and dataset, for example a 3D MRI scan as in Figure 3-8a so that an explicit relationship between the visible brain tumor in the focus and structures viewed in the past can be obtained. The hierarchical nature of our approach supports the assignment of a different dataset or a transfer function to the focus-driven context. This third dataset can provide spatial relationship information between the different modalities and how the focus region (a brain tumor from an MRI scan in Figure 3-8) is related to the skin and soft tissue of the patient (MRI dataset shown in the context area) and how it is related to the immediate surrounding bone - Figure 3-8b, or brain activity - Figure 3-8c. Focus-driven context also addresses the lack of explicit relationship between 2D slices within the lens and the 3D rendering of the focus volume. The main focus area can be assigned to display the 2D slice visualization within the lens, while the 3D volume is rendered inside the focus-driven context region as in Figure 3-8d.

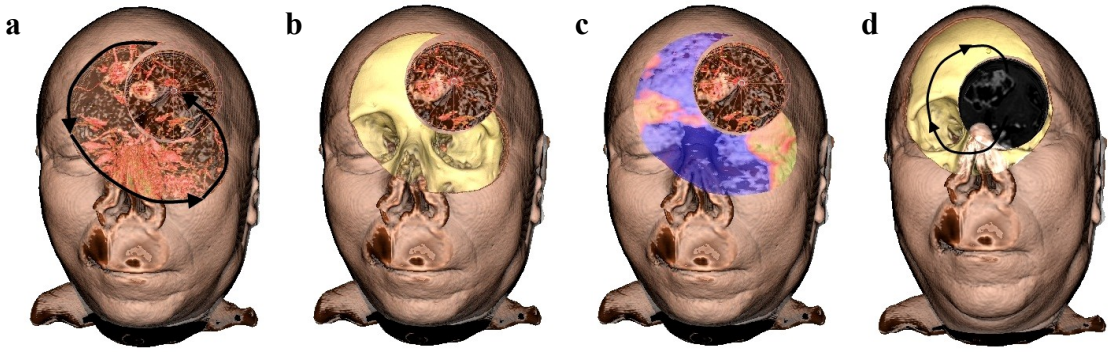

Figure 3-8: Focus-driven context in different exploratory scenarios. The focus area can display the same dataset as the focus-driven context area (**a**), the focus-driven context can display different datasets – for example a CT scan (**b**) or a PET scan (**c**). An MRI slice is visible in the lens, while focus-driven context displays 3D rendering of a CT scan (**d**). The arrows show the path of the lens.

A further examination of the spatial relationship cues provided by either of the context areas shows that radiologists can use the focus-driven context as an explicit short-

term memory that provides information about how the explored target objects are related to areas far from it but important for the placement of accurate diagnosis. Surgeons can use this visualization to ensure clear visibility of the target region and also visualize structures extracted from different dataset modalities lying along the incision path.

3.5 Arbitrary focus region definition

3.5.1 Overview of technique

Any of the current visualization paradigms that can be explained using the proposed generalized framework utilizes one 3D focus geometry. Usually this is a predefined shape that is used to probe the datasets in order to get a better understanding of the dataset composition. However, none of these techniques considers visualizations where the focus is defined by multiple lens positions in the temporal domain, with the possibility of having modifiable concave focus shapes at runtime. Therefore, we explored time-dependent framework-based techniques that can be used to define an arbitrary-shaped focus region made up of the lens geometry as it is moved in space over time.

Thus, we introduce a focus + context visualization which uses a focus region of constantly changing shape that allows us to sculpt the context dataset. The focus region shape gets bigger and more complex with every newly saved 3D lens position. For example, using a focus + context visualization with a cubic lens, we can save four positions of the lens as time passes by. The result of rendering all saved lens positions together is the definition of an arbitrary-shaped concave focus region shown as a red outline in Figure 3-9a. The volume inside the newly defined focus region can be rendered using any of the described techniques - 2D, 3D or 3D sub-volume.

Utilizing the visualization paradigm of defining an arbitrary shaped focus region, we can adjust the visualization parameters to cause no focus rendering. This can be done by either traversing the rays inside the focus and setting the transparency at the end of each dashed segment from Figure 3-9b equal to 0, or skipping the dashed portion of the rays and starting the accumulation of color and opacity after those segments. Therefore no rendering is executed inside the focus, and only context is visible. Thus, the described paradigm can be viewed as a way of carving out material in real time as if sculpting the context dataset to explore optimal surgical intervention trajectories.

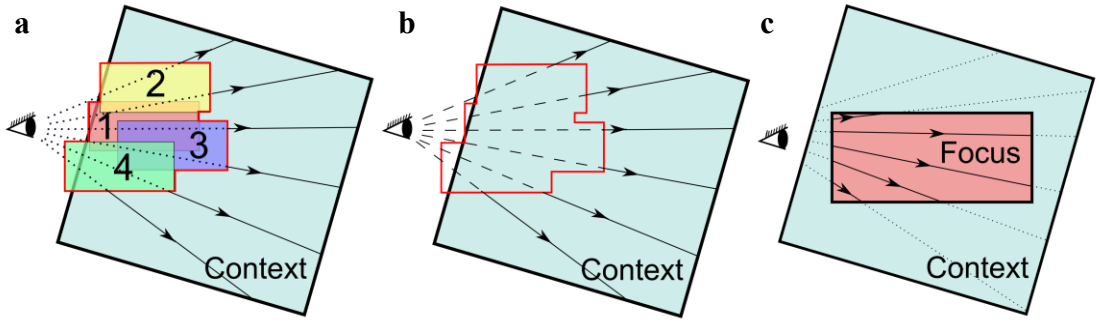

Figure 3-9. A diagram showing the portion of the rays considered during each of the surgical procedure visualizations. (**a**) Multiple lens positions can be saved and used as focus. (**b**) When volume sculpting, the rays are skipped inside the lens and are traversed when passing through context. (**c**) Inner sub-volume exploration is achieved by traversing only those rays that fall inside the focus.

3.5.2 Volume sculpting for surgical path planning applications

Significant research has been devoted to improving tissue removal visualizations and simulations in order to achieve accuracy and realism. Surgical simulators are often used in pre-operative planning so that the surgeon can practice the planned procedure on patient-specific datasets and explore possible incision paths [71-73]. However, there are few systems that allow users to virtually sculpt the dataset and explore the result of the performed simulated procedure. Even though current drilling simulators support such tasks, they are limited to very specific procedures and expect prior detailed knowledge of

the patient anatomy. The two most often used tools - a spherical drill and a suction tool do not provide any information about objects in front of them, therefore the user needs to anticipate what objects are at the tip of the drill.

To address the shortcomings of current dataset sculpting techniques, we applied our arbitrary-shaped focus to volume sculpting tasks. The described visualization of defining an arbitrary-shaped focus region can be successfully used to modify the underlying anatomy when surgeons need to plan incision trajectories. Drilling simulations and sculpting visualization use three-dimensional solid drills as in Figure 3-10a. In contrast, the fact that our sculpting tool is a lens from the introduced generalized focus + context framework lets us assign any rendering modality to it. If we allow a 3D volume rendering inside the lens, the surgeon can visualize what parts will be removed if the tool is placed at the given position. The surgeon sees the exact volume that will be carved out and can thus adjust the position and orientation of the tool so that no vital organs are damaged.

The 3D rendering inside the lens might not give any detailed information since the transfer function might not capture the whole range of important intensity values. For instance, some blood vessels might not be rendered, or they might be occluded by 3D structures visible in front of the blood vessels. Therefore, we can enable the 2D slice visualization introduced in Chapter 3.2. Once the position of the lens is determined (Figure 3-10b), the surgeons can switch to the 2D slice rendering (Figure 3-10c) and explore the slices that fall within the lens region to ensure that all data is safe to remove. If desired, the sculpting tool position can be adjusted with precision equal to the dataset spacing. Each position of the sculpting tool is saved by the user and is used to create an arbitrary-shaped focus region with no volume rendering inside it. For example, the result

from sculpting the context dataset by removing the region visible in Figure 3-10b and c is shown in Figure 3-10d. This combination of focus + context techniques guarantees very precise path planning with sufficient visible detail.

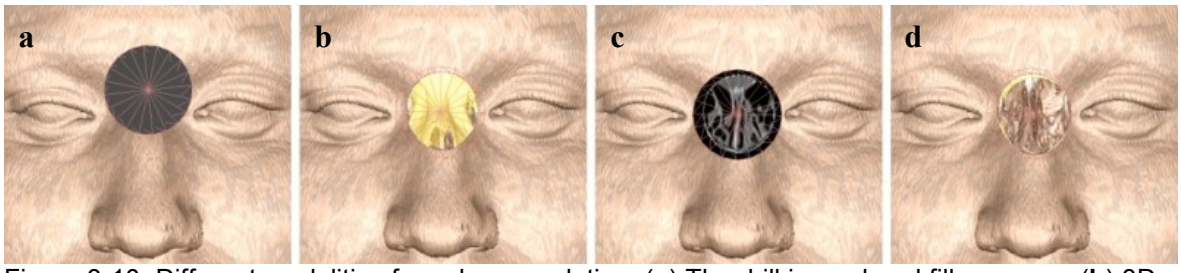

Figure 3-10: Different modalities for volume sculpting. (**a**) The drill is rendered filly opaque. (**b**) 3D volume inside the drill is displayed. (**c**) The back-most slice at the current drill position is shown. (**d**) The result from removing the section selected in (b).

The advantage of path planning and volume sculpting using the proposed technique is that those parts that have been removed with the arbitrary lens geometry are not lost. As the removal of structures is a result of skipping any rendering inside a large focus region, the information inside this focus shape is preserved. If the surgeon would like to see the 3D structure that would be removed after the sculpting procedure is completed (Figure 3-11b), he can enable 3D volume rendering inside the multiple saved lens positions that were used (Figure 3-11a). Due to the definition of the removed regions as a framework-based paradigm, we support the interactive exploration of these regions by enabling rendering inside the removed parts and using the magic lens to probe the specified volume to see 2D, 3D or 3D sub-volume rendering.

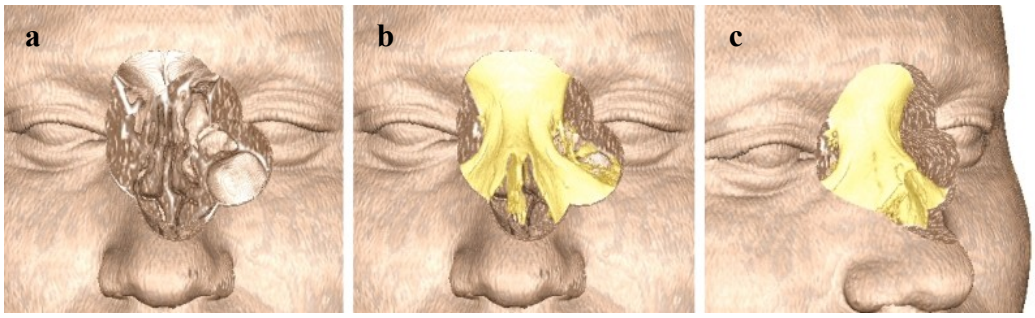

Figure 3-11: Visualizing sculpted-out areas. (**a**) A large portion of the sinuses have been drilled out. (**b**) Rendering the removed areas from the same viewing angle. (**c**) Rotation of the model allows better exploration of the removed volume.

3.6. Inner sub-volume exploration

3.6.1 Overview of technique

While any of the described techniques that fit within the presented framework have context rendered as a 2D slice or a 3D volume, none of the techniques address the possibility of not displaying context and what should be seen as a result. Therefore we explore a new visualization within the framework domain where no context is visible. The focus region is a 3D geometry with size smaller than the focus dataset. In this visualization the volume is rendered by not traversing rays outside the lens – i.e. dotted rays that pass through context as seen in Figure 3-9c. The lens position can be anywhere inside the focus dataset and no context would occlude the volume within the lens. The rendering inside the focus region can be switched between 2D, 3D and 3D sub-volume so that the focus region can be explored in detail.

The visualization of inner sub-volume exploration defined by the generalized focus + context framework can be used to achieve visualizations similar to virtual endoscopy [74, 75]. In virtual endoscopy the position of the endoscope is considered to be the virtual camera position, and the endoscope is advanced through a hollow organ. The rendering algorithm traverses rays from the camera to the walls of the object through which the

endoscope is moved. Nothing beyond the opaque walls is rendered, and any structures right behind are not visible. The final image is similar to the image obtained during real endoscopic procedures that could be uncomfortable for patients. However, such a technique would not be sufficient if a physician is looking for abnormal structures along either side of a hollow structure's wall. A visualization similar to virtual endoscopy with the added advantage of providing visibility of structures beyond the organ's walls can be achieved using the described visualization of not rendering context and moving the lens in 6 DOF.

3.6.2 Surgical planning application

Unlike virtual endoscopy where the whole hollow anatomy in front of the camera is visible, the proposed focus + context visualization for inner volume exploration shows only those parts of the volume that fall within the lens. Therefore, this visualization method for inner volume exploration allows rendering of the volume along a path that is often chosen for endoscopic procedures and makes sure that the tool is not advanced too far in the organ. This is best illustrated in sinus surgery where the surgeon inserts an endoscope in the nasal cavity (Figure 3-12a) and has to navigate around the sinus structures. When using an endoscope, the surgeon is not aware of what lies outside the nasal cavity and the sinuses. When using the lens as a navigation tool, the surgeon can traverse the lens along the nasal cavity while also seeing structures beyond the nasal walls (Figure 3-12b). When advanced further in, the lens can display structures within the patient's sinuses as well as the structure of the sinus cells (Figure 3-12 c and d).

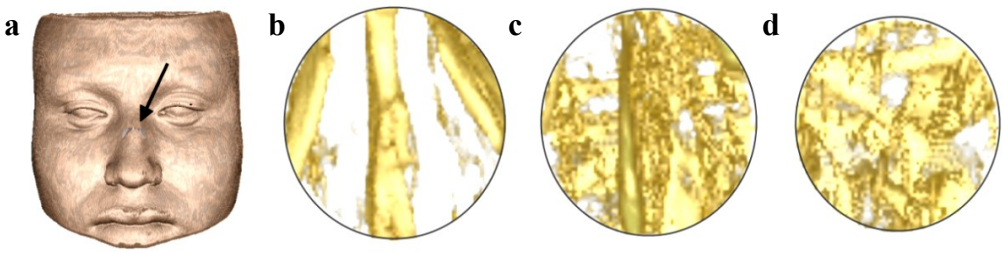

Figure 3-12: Visualizing areas inside the volume. The lens is inside the sinuses (shown by an arrow in **a**). Moving the lens along the nasal cavity path reveals different volumetric structures – **b**, **c**, **d**.

In our inner volume exploration visualization the lens is moved in 6DOF using an EM tracker and the volume that falls inside it can be seen. If the lens is positioned in a hollow organ, for example the colon, and the diameter of the lens is at least equal to the average diameter of the colon, the walls of the colon that fall inside the lens geometry will be visible. In cases when the ray opacity at the wall's surface is less than one, the traversal will continue beyond the colon walls, allowing the surgeon to also see anatomies that are usually occluded by the colon. If necessary, the rendering modality can be switched to 2D or 3D sub-volume in order to support a detailed investigation of regions of interest. Thus, abnormalities or cancerous structures on the outside wall of the colon can be detected and investigated further, which is not possible in standard virtual colonoscopy visualizations.

3.7 Collaborative meeting application

The typical medical team meeting environment has been presented and a proposed improvement has been described in the work of Olwal et al. [76]. A radiologist sits in front of a computer workstation where he can manipulate the slices that constitute the dataset. The slices of interest are projected on large screens, and surgeons view the screens from their seats positioned far from the screen. In this conference-style meeting

setup the radiologist is the one who explains his findings and then surgeons discuss the treatment plan with limited control over what is shown on the screen. They would point to a specific area with their hand or a laser pointer, which is very inaccurate, and other discussion participants might infer wrong positions of interest. Olwal et al. proposed an interaction system where medical specialists scroll through slices and point at regions of interest on their tablets, and the result is shown on the projection screen. However, this may cause users to lose concentration while they are manipulating the dataset on their device. It has also been pointed out that displaying and manipulating 3D dataset renderings can be beneficial in such conference-style meetings [77]. Furthermore, in such multidisciplinary meetings the radiologist has to pay attention to what surgeons suggest and has to provide his feedback if there are any organs that have to be avoided and the surgical team is not aware of. Groth et al. [77] gives insight on the disadvantages of such a verbal-based planning scheme. Lack of sufficient information, indistinct words or even the slightest distraction can cause inefficient meetings and confusion among the team members. As presented, while surgeons were thinking out-loud about possible ways of getting to the target object, the radiologist had to repeat several times that there were areas near the target that had to be avoided. The surgeons did not pay attention to this information for a while since they were occupied mentally reconstructing the dataset from different viewing angles and had to rely on limited images and a verbal discussion.

Realizing that while radiologists are used to 2D slice visualizations, surgeons often require a different visualization modality and the option of independent interaction with the dataset, we investigated the unification of the presented new approaches for dataset and surgical path explorations. Being paradigms within the same temporal focus +

context framework, the numerous possible ways of combining them can serve both radiologists and surgeons in their multidisciplinary discussions. Therefore, a system implemented using the introduced focus + context framework can accommodate multiple users, each of whom has a framework-based visualization with the flexibility to display any desired visualization. Our collaborative framework-based environment has been implemented using a single workstation and currently serves as a proof-of-concept that multiple magic lenses that show all of the already proposed extensions – 2D, 3D, 3D-subvolume, focus-driven context and surgical path visualizations – can be combined so that a more efficient and precise discussion can be achieved.

Conceptually the combination of techniques can be viewed as having multiple dependent or independent visualizations from the generalized temporal focus + context framework on the same screen. Each lens visualization is presented in its own window, and all current focus positions are also visible in a birds-eye view window that also employs a generalized framework visualization. To help with the spatial relation of multiple different focus regions, we combine these techniques with defined authority parameters so that one user can have partial control over other users' focus regions while using the birds-eye view.

Being aware of the authority of radiologists while presenting cases and findings, we gave radiologists more control of the visualization and the position of the tools that surgeons use. In our example (Figure 3-13) a radiologist and a surgeon each have a magic lens that displays the focus dataset in a way that they prefer. For example, the radiologist uses a slice rendering inside the lens and shows the surgeon the target slice from the optimal position on a birds-eye view screen that renders the whole dataset as in Figure 3-

13a: Radiologist View. Now the surgeon has to be able to explore the dataset, point at areas of interest or investigate possible incision paths to validate his idea. To do that, the surgeon starts off with a magic lens with the same rendering modality as the radiologist and at the same position (Figure 3-13a: Surgeon View). The lens can display 2D, 3D or 3D-sub volume so that the user can explore the target in a way that is optimal for him: in Figure 3-13b the radiologist's lens shows a 2D slice, while the surgeon's lens is a 3D rendering of the same region.

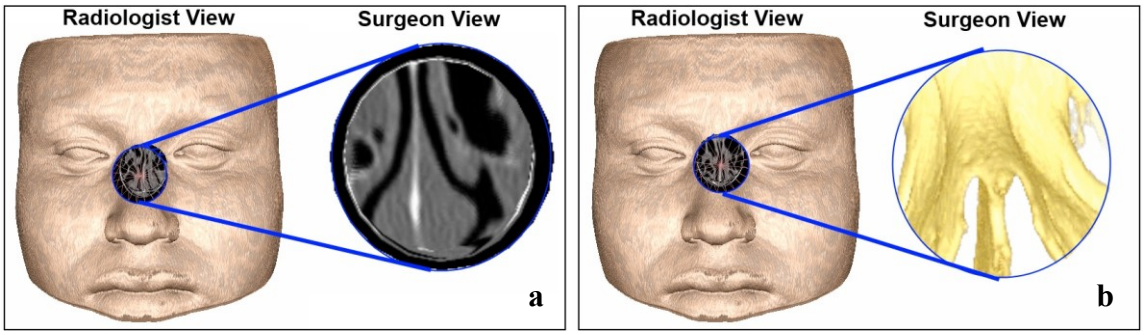

Figure 3-13: Radiologist versus surgeon view. Each view can be assigned different rendering parameters. (**a**) CT slice visualization inside both views. (**b**) CT slice seen inside the radiologist' lens, 3D volume rendering inside the surgeon's lens.

The radiologist and the surgeon lenses are moved interactively using an EM tracker. The 6 DOF movement of the focus lenses can be limited to a fewer number of DOF if the user wants to explore a certain area, or can be completely independent of the other user's lens. For example, while the surgeon is exploring different incision paths, he can see a 3D volume or a 2D slice rendering of the area in a separate window where he can explore the volumetric structural composition. The radiologist's overview window of the whole dataset allows him to see a birds-eye view of the surgeon's lens and if there are any position corrections to be made.

Passing through the inside structures, the surgeon can also sculpt the dataset. This will be respectively shown in the radiologist's window. Once the sculpting tool positions

are saved, the removed volume can be rendered so that both team members can evaluate the appropriateness of the proposed treatment plan in a visualization modality that best suits their dataset understanding.

The unification of techniques can be successfully implemented in a conference room setting, or when only a few physicians use the system on a computer screen. However, depending on the room size and the number of participants, there is a need for using different types of input devices for the control of the focus regions. EM trackers are connected to the computer using wires, and they have a limited range within which the electromagnetic transmitter detects position and orientation. An optimal interaction device for large rooms with multiple people would be a depth camera like Microsoft Kinect®. Kinect can track up to six users at the same time, and the sensor's range is much larger than that of an EM tracker. Currently our system successfully employs tracking using Kinect where the position of the user's left hand corresponds to the position of the lens. Movement of the user's right hand in the 2D or 3D sub-volume rendering modalities changes the position of the visible slice. Further work is necessary to develop a 6DOF orientation tracking based on a single hand motion.

Chapter 4: GPU rendering of proposed approaches

Our focus + context visualization uses the raycasting method described in Kirmizibayrak [10]. It has been extended to use the depth peeling algorithm originally introduced by Everitt [78] so that the proposed framework can support volume sculpting. The frame rate of the used raycasting techniques is compared to the work presented in Chapter 2 and summarized in Table 2-1.

4.1 Three-pass GPU raycasting

The underlying GPU rendering technique described in [10] requires two types of preprocessing steps. The first one is loading all necessary datasets and extracting the gradient which is used to compute the Phong shading of the volumes, as shown on the left-most side of Figure 4-1. The second preprocessing step renders the context and the lens meshes to depth buffers shown in Figure 4-1 (second column) that provide information about the beginning and the end of the traversed rays for each volume. This information, together with the datasets and the gradient information, is sent to the GPU where the raycasting is executed. Based on the input information, the fragment shader computes three areas - before the lens, within the lens, and after the lens for an optimized ray traversal, and accumulates color and opacity within each traversed region.

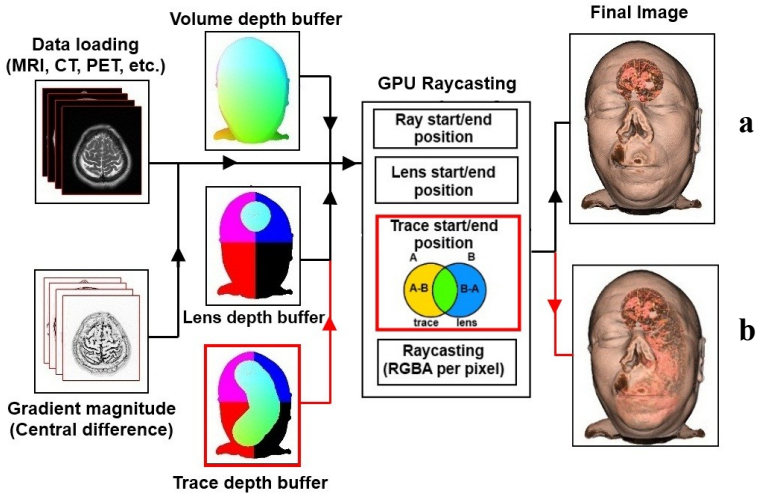

Figure 4-1: Rendering pipeline used for the focus + context visualization is shown by the black squares and arrows. (**a**) Final image for the focus + context visualization. (**b**) Final image for the focus-driven context visualization. The red squares and arrows show the additional steps in the rendering algorithm.

To render the slice corresponding to the position defined by the user, we modified the ray traversal algorithm applied to the rays that fall within the lens geometry. Instead of traversing the ray from the start of the lens, we compute the point along the ray where the ray intersects the selected slice position. Therefore, only one volume sample is required to extract the correct color data, while the corresponding pixel is rendered at full opacity.

To render the volume between the oldest and the most current slice positions, we extend the slice-based focus + context GPU raycasting technique described above. The slice positions are saved in an array of size N, where N depends on the user-defined time span. Our raycasting algorithm uses the oldest and newest slice positions at every frame (designated red and blue on Figure 4-2), and thus we can compute the position where each ray passes through the two slices. The ray is traversed and color and opacity are accumulated only between the two computed points. As shown on Figure 4-2, the ray is skipped (dotted line) until it hits the front (red) slice, after which we accumulate color and opacity until it hits the back (blue) slice. When the ray intersects the slice furthest

away from the viewer (slice 2), the grayscale color value of the slice at that position is added to the current RGB value to display the slice right behind the rendered sub-volume. Using this technique, we do not need to traverse the ray before the front slice and behind the back slice, therefore we can further speed up the volume rendering by volume skipping. This ensures that our visualization always remains at interactive rates even when multiple large datasets are imported, as observed in Table 4-2.

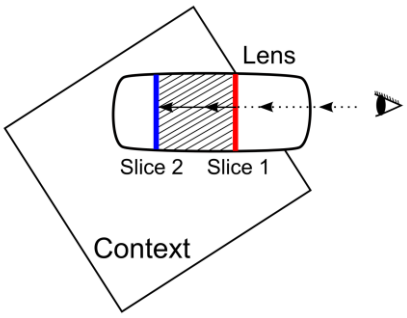

Figure 4-2: Ray traversal for 3D sub-volume rendering. The ray that falls within the lens is traversed only between the front-most (red) and back-most (blue) slice positions in order to speed up color accumulation calculations.

A modification of the raycasting algorithm has to be done when rendering the focus-driven context introduced in Chapter 3.4. We extend the algorithm presented above to correctly render the area defined by the past *n* saved lens positions. As shown in Figure 4-1 (second column), a third depth buffer called Trace Depth Buffer is created during the pre-processing step and generates all focus-driven context textures necessary for the GPU rendering. It holds depth renderings of the 3D lens volume at all positions that define the trace area. Once the rendering to the depth buffer is completed, the newly created depth textures are also sent to the GPU.

The GPU rendering of the focus-driven context starts with finding whether a pixel falls within the focus-driven context area by computing the distance between the trace start position and the volume start position, as well as the distance between the trace and

the lens. The most-recently stored position of the lens as a trace area (area A in Figure 4-1) would most likely have an overlap with the current position of the lens (area B). In order to avoid raytracing two times in this overlapping area, we find the overlap (A∩B) using the computed distance between the trace and the lens. Since the lens volume has higher importance than the focus-driven context, the focus volume will be displayed in the overlapped area. For every ray that passes through the focus-driven context (area A-B shaded in yellow in Figure 4-1), we follow the ray and accumulate context color and opacity. The same ray is also traversed to compute the focus-driven context color and opacity based on the assigned dataset. For this computation, we split the ray into three regions – in front of the trace area, within the trace area, and after the trace area, similar to how we render the lens volume. This will ensure that the trace region displays the same volume that has been displayed as focus a few seconds ago. Once the RGBA values for trace and context are computed, we perform a weighted RGBA blending of the two areas using equation 4.1, where w is the weight blending factor.

$$\alpha_{final} = (1\text{-}w) * \alpha_{context} + w*\alpha_{trace} \quad (4.1)$$

$$w = w_{user}*(1.0 - w_{distance}) \quad (4.2)$$

As shown in equation 4.2, w is computed using two weights: w_{user} - a user-defined alpha blending and $w_{distance}$ - a distance-dependent weight computed by the raycasting algorithm which can be enabled or disabled. W_{user} is a user-defined alpha-blending weight which is allowed to vary between 0.0 and 1.0. $W_{distance}$ is computed per pixel and determines the distance between the current magic lens position and the current focus-driven context pixel. Therefore, pixels that are near the magic lens will be displayed by

merging context and trace with the weight w nearly equal or equal to w_{user} since $w_{distance}$ is close to or equal to zero. The merging factor w will get smaller the further we get from the lens, allowing the display of more context while the focus-driven context is assigned reduced visibility.

An illustration of the described weighted RGBA blending is presented in Figure 4-3. Figure 4-3a shows focus-driven context at the same opacity as the focus region. This focus-driven context, however, does not provide any additional cues that a magic lens with a size and shape equal to the currently displayed CT dataset would provide. Therefore, in Figure 4-3b we show the weighted RGBA-blending between context and focus-driven context regions so that we can see not only the areas where the lens has been, but also important context features like skin and eyelids. Figures 3-4 c and d display the same interaction time step, however we have reduced w_{user} to 0.3 and 0.1 respectively to reveal more context.

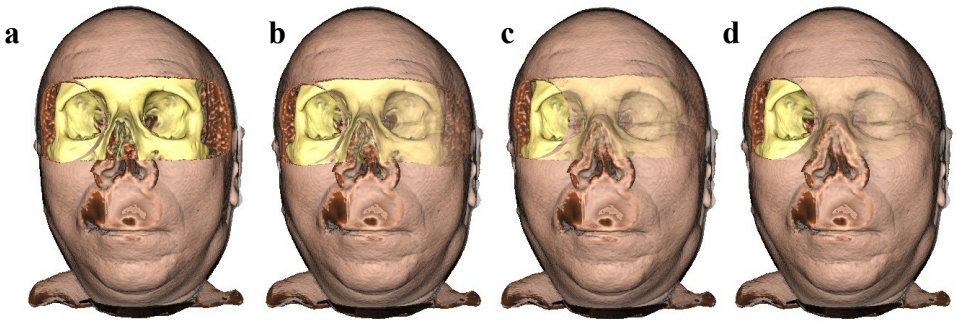

Figure 4-3: Time-dependent focus-driven context with different weighted RGBA blending. Magic lens is positioned on the left side of the volume. Context displays MRI dataset, focus and focus-driven context display CT dataset. (**a**) Focus-driven context not blended with context. Focus-driven context alpha-blended with the context using distance-dependent weight and (**b**) w_{user} = 0.5. (**c**) w_{user} = 0.3. (**d**) w_{user} = 0.1.

4.2 GPU rendering using depth-peeling and a new parity computation

The utilized raycasting approach described in Chapter 4.1 however gives artifacts when used for the volume sculpting visualization of Chapter 3.5. This is due to the fact

that the arbitrary-shaped focus region constitutes multiple lenses which might have insignificant overlap. Therefore, when using the front-most (red in Figure 4-4a) and back-most faces (blue in Figure 4-4a) among all saved lens geometry meshes, wrong ray traversal occurs in regions where the ray passes through both sculpted areas and through context as shown by the top-most ray in Figure 4-4a.

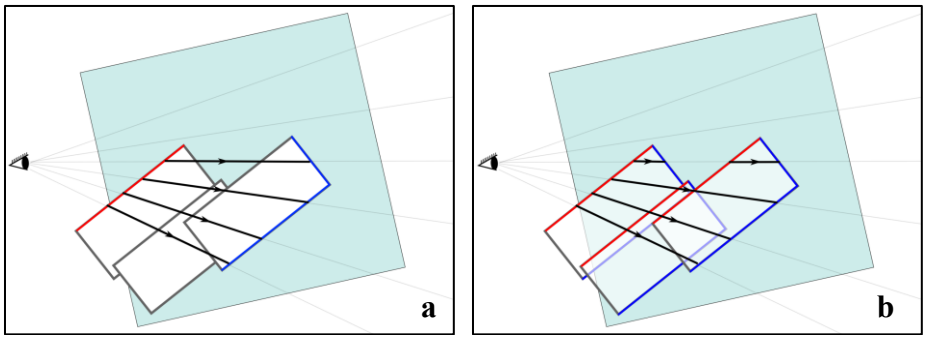

Figure 4-4: Two different ways of ray position computation. (**a**) The ray segment inside the drilled-out region is computed by the front and back-most faces of the volume. (**b**) Depth peeling can help for the correct computation of the ray inside the drilled regions.

To correct for this, the start and end positions of the rays for each unique region have to be defined more accurately. One way is to render the front (red) and back (blue) faces of each lens position separately as shown in Figure 4-4b and then compute the overlap between the lenses based on the start and end positions of each lens' mesh. If multiple overlapping lens positions are used for volume sculpting, this technique becomes highly inefficient and can slow down the rendering speed to under-real-time.

As presented in [79], we achieve accurate rendering of the complex concave geometrical region by combining depth peeling techniques [78] and performing a new way for parity computation on the extracted layers. Similarly to [57], the first and front-most layer is extracted by rendering all saved lens geometries to the depth buffer. All consequent layers are computed by rendering the saved geometries without face culling

and discarding fragments with depth less than or equal to the previously extracted depth layer. Unlike [57] where the sequence of depth layers is saved by means of bitwise operations and the number of saved layers is limited to 32, we incorporated a parity computation similar to the one presented in [80] and therefore we can support an unlimited number of tool positions. The parity computation method described in [80] is done as a preprocessing step and it cannot accommodate the visualization of regions that change their shape and size dynamically at run-time. Furthermore, the parity values can be only 0 or 1 for each depth layer, making it applicable to only CSG volumes.

To make the parity computation method applicable to the rending of our complex sculpted region, we utilize a constant *c* for the parity at every depth layer. We add the constant *c* to the computed parity for the previous depth layer when we have a front-facing depth layer *i* and subtract the same constant *c* in cases of back-facing layers as presented in equation 4.1 and graphically shown in Figure 4-5. The value of the constant *c* depends on the number of saved geometries that make up the sculpted region, with $c = 1/ number\_of\_saved\_lenses$.

$$\begin{aligned} Parity_i &= Parity_{i-1} + c \quad \text{if layer i is a front face} \\ Parity_i &= Parity_{i-1} - c \quad \text{if layer i is a back face} \end{aligned} \tag{4.3}$$

To identify whether we need to add or subtract *c* to the previously computed parity value, the algorithm executes two calls to the GPU rather than only one call as in the work of [57, 80]. Every layer following the initial front-most depth layer is computed as follows. During the first GPU call, the next depth layer *i* is extracted by disabling face culling and applying depth comparisons. In the second call, the algorithm extracts the next front-face depth layer and compares this depth value to the depth value saved from

the previous GPU call. If the values are equal, the initially extracted depth layer corresponds to a front face and *c* is added to the parity, otherwise *c* is subtracted. The depth and parity values for layers *i* and *i-1* are saved in two textures that are alternated. Color is computed depending on the parity saved in each layer. If the parity of layer *i-1* is 0 and the parity of layer *i* is equal to *c*, then the region between the two depth layers lies within the context region. In any other case the traversed ray passes through the complex focus region that has been sculpted out.

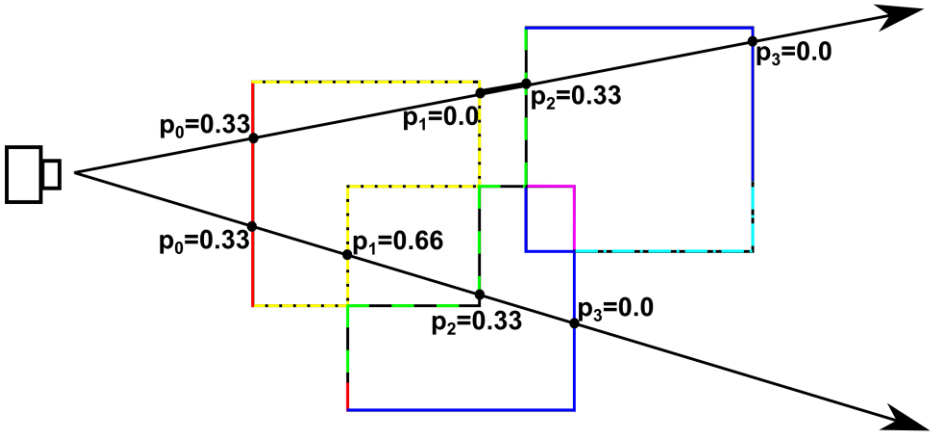

Figure 4-5: Depth peeling and parity computation for two rays traversing three square tool regions. The depth layers are represented by different colors: layer 0: red, layer 1: yellow, layer 2: green; layer 3: blue; layer 4: purple, layer 5: cyan. Parity for each volume intersection is shown.

To speed up the visualization, depth peeling and parity computations are done only for the arbitrary-shaped focus region. The RGBA values for this region are computed during the depth peeling computations and saved in a texture. This texture becomes input to the raycasting algorithm used for the whole visualization. If the ray is found to be traversing an arbitrary-shaped focus region defined by its front-most and its back-most depths, a lookup of RGBA in the saved texture is performed and the values are blended with the RGBA values accumulated for the ray.

This algorithm is strongly dependent on the number of lens geometries saved. The more geometries we need to depth peel, the slower the algorithm becomes, which is a known drawback of the depth peeling algorithm.

4.3 Interactivity evaluation

A key factor in the used visualization technique is the interactivity of the system so that no lag in the lens position and trace movement is introduced. We quantitatively compared the rendering speeds in frames per second (fps) using a medical case and looked at all proposed visualizations. The case is a multimodal dataset (a MRI scan with dimensions 176x224x256 and a CT scan with dimensions 512x512x174) registered using the ICP registration algorithm [81]. Two different transfer functions are used for the rendering of skin/soft tissue and for the rendering of bone. All computations are done on a Dell Precision T7600 workstation. The GPU card used is an NVIDIA Quadro 6000 graphics card with 6GB of graphics memory. The results from the comparison are presented in Table 4-1.

The combination of 2D and 3D volume rendering does not create any significant performance and memory overheads compared to the standard focus + context visualization paradigm. When only 2D slices are displayed, no changes in the rendering speed were observed. In the case of 3D sub-volume rendering, the number of frames per second depends on the size of the rendered sub-volume. The maximum number of fps occurs when the rendered sub-volume is between two consecutive slices, while the minimum number of fps is seen when the whole volume inside the lens has to be

rendered. Any other sub-volume size will generate speeds between the minimum and maximum observed fps.

For rendering focus-driven context we used the modified GPU raycasting technique described in Chapter 4.1. It introduces small reductions in the frame rate. Without using the focus-driven context, the rendering speed is around 59 frames per second depending on the areas over which the lens is positioned. The observed frame rate is significantly higher than previous work as presented in Table 2-1 where the average rate is often 20 fps or less, and in rare occasions higher than 30 fps. The size of the cylindrical lens for the speed comparison is fixed to dimensions 0.3x0.3x0.45 of the original volume. When we introduce the focus-driven context, we observe a reduction of rendered frames per second explained by the fact that we have to trace more rays and do more color and alpha computations. Thus, the speed is inversely proportional to the size of the lens and the number of lens geometries that define the trace. However, no speeds under 28 fps were observed under any dataset combinations even if the focus-driven context took up the whole context area.

Table 4-1. Comparison between the various combinations of rendering techniques - Focus (F), Context (C), Focus-driven context (FDC), 3D sub-volume (3DS). The observed fps have been recorded for different combinations of datasets displayed in the various regions.

VISUALIZATION	FOCUS DATA-SET	CONTEXT DATASET	TRACE. DATA-SET	FPS (AVERAGE OVER MULTIPLE LENS POSITIONS)
Standard F + C	MRI	MRI	-	59
Standard F + C	CT	MRI	-	59
F + C +FDC	MRI	MRI	MRI	47 - 59
F + C +FDC	MRI	MRI	CT	50-59
Focus is 2D slice	CT	MRI	-	59
Focus is 2D slice + FDC	CT	MRI	MRI	28 - 57
Focus is 3DS	CT	MRI	-	58-59
Focus is 3DS + FDC	MRI with MRI slice	MRI	MRI	28: all context is trace, subvolume is whole lens 47: small trace and subvolume

Table 4-1 (cont.) Comparison between the various combinations of rendering techniques.

Focus is 3DS + FDC	CT with MRI slice	MRI	CT	29: all context is trace, subvolume is whole lens 49: small trace and subvolume
Surgeon + Radiologist visualization (both lenses with 3D volume rendering)	MRI	MRI		59

The applied depth peeling algorithm has been evaluated by comparing the average observed speed performance when changing the size of the sculpting tool and the number of cut-out regions – Figure 4-6. As shown, no significant performance changes were observed when depth peeling was performed on less than 10 saved lenses. If more saved geometries were to be depth-peeled, the visualization showed a decrease in performance. However, even with 40 saved geometries the algorithm maintained real-time speeds. The performance is not dependent on the size of the sculpting tool, showing that the main reason for the decreasing performance is the non-optimized depth peeling method. There was no observed dependence between the size of the sculpting tool and the overall visualization performance.

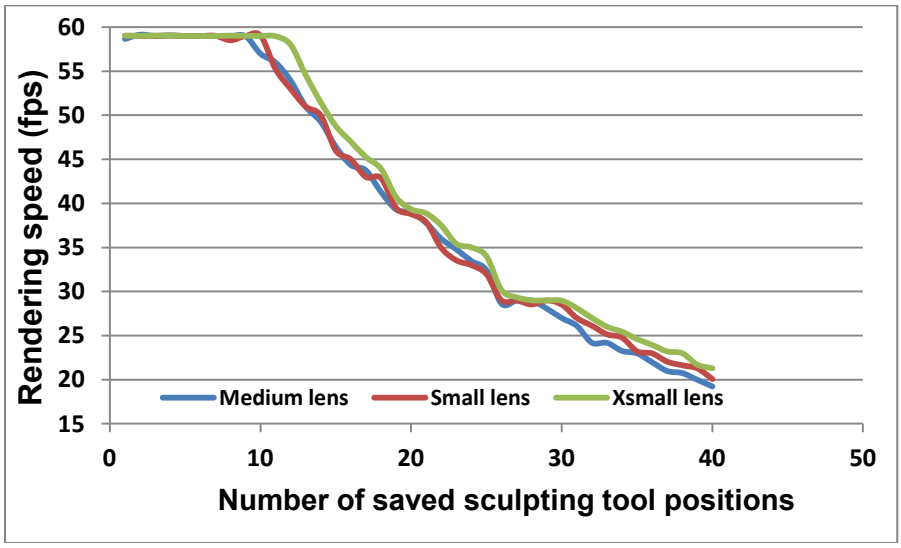

Figure 4-6: Rendering performance when a sculpting tool is used. The performance is independent of the size of the tool used.

Chapter 5: User study for the validation of the proposed approach

To validate the efficiency and usefulness of the proposed generalized framework, we executed a user study that compared three of the novel visualizations to already existing visualizations. The study consisted of three experiments, completed in two separate successive blocks. The order of the tasks was randomized within each block. The study was approved by our institution's ethics committee and all participants gave their informed consent. The study was performed by 18 subjects (9 male and 9 female) with median age of 28 years. To exclude factors dependent on previous medical knowledge and experience we recruited college-educated individuals with no medical background.

Upon arrival users were informed about the purpose of the study and that response time and accuracy between novel and existing techniques would be compared. We did not specify which techniques were new. We provided a brief explanation of each experiment, supplemented by graphical depictions. Each subject performed a practice run using a different dataset (the Stanford Bunny) than the study proper (the Piggy bank). Each visualization technique was tested only with one target position so that users could get acquainted with how to control the different visualizations and how to proceed once a target has been localized. Once comfortable with the interaction, users ran the user study. The first block was conducted to provide users some familiarity with the visualization methods before engaging in the experimental trials in the second block; data from these practice trials are not included in the analyses described below. A paper survey regarding the study was conducted at the end.

5.1 Experiment 1

The first experiment compared a focus + context visualization that uses a front-clipping plane within the lens with a visualization that uses two clipping planes to define sub-regions of interest, which is a simplified version of the 3D sub-volume visualization introduced in Chapter 3.3. The task required users to find which pair of cubes from the focus dataset was separated in depth by a gap. The focus dataset consisted of two pairs of cubes – two cubes had a gap in-between with gap size of 25 voxels - Figure 5-1a, and the other two had no distance between them – Figure 5-1b. The experiment was tested with four randomly positioned target objects during each block.

The focus + context visualization used a cylindrical focus region with diameter slightly wider than the width of the target cubes. The first technique allowed users to adjust the front clipping plane (FCP) of the focus region using the up and down keyboard keys (Figure 5-1c). In the second technique users were allowed to adjust both the front and the back clipping planes (FBCP) in order to explore sub-sections of the focus region - Figure 5-1d. The front clipping plane was adjusted using the up and down keys, while the back clipping plane was adjusted using the up and down keys from the numeric pad. Accuracy was scored as True or False, based on the distance between the saved mouse position and the projected correct target position. In both techniques we provided the depth visualization tool described in Section 3.3.3 so the depth position and the depth of the rendered volume could be explicitly seen.

We hypothesized that having two clipping planes that also provide slice visibility would improve structural understanding of anatomies from multimodal datasets. Therefore, we expected to observe accuracy improvement when using FBCP. We also

anticipated instances in which users would not be able to find the target object using FCP. We hypothesized that FCP would result in significantly faster response time than FBCP because it is a simpler technique to interact with.

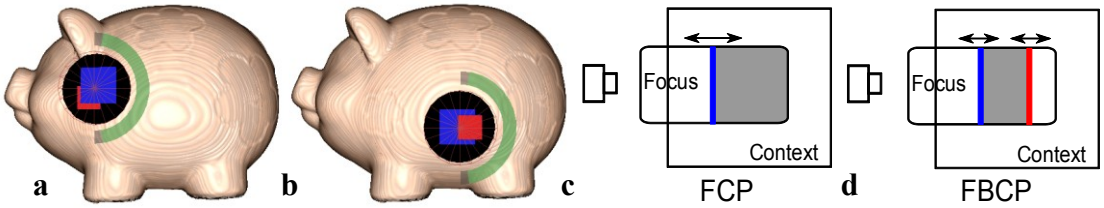

Figure 5-1: The techniques used in experiment 1. The two sets of cubes as seen by users (**a, b**). Graphical representation of the two compared techniques – FCP (**c**) and FBCP (**d**).

For each user in experiment 1 we calculated the average response time across the 4 trials. The response time performance analysis is graphically shown in Figure 5-2. The average response time for FCP was 26.54 +/- 2.74s while for FBCP it was 28.73 +/- 2.57s. Pairwise post-hoc comparison (two-tailed t-tests) between the two averages did not result in significant differences between the two (p=0.43).

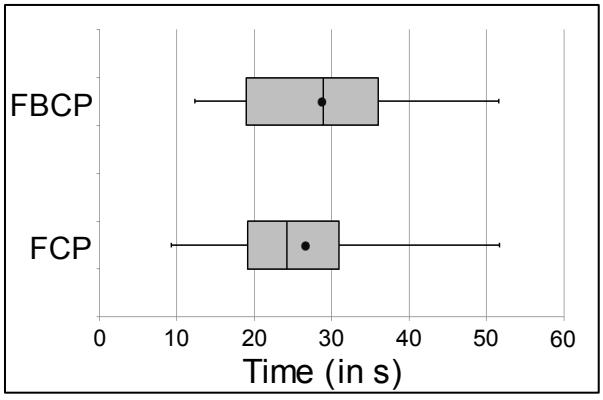

Figure 5-2: Box and whisker plot of response time for FCP and FBCP. Average time is represented by the dots.

The accuracy measurement was represented as either 0 (not correct), or 1 (correct). The average accuracy scores across 4 trials per technique for FCP and FBCP were 86% +/- 4% vs. 96% +/- 2% respectively. A paired-sample t-test on the accuracy data was

marginally significant ($p = 0.069$). This hints at a trend toward improved understanding of depth relationships among occluding objects when two adjustable clipping planes are available.

Three questions were asked in the paper survey. Users were asked to identify which visualization was easiest to use and which one gave them a better understanding of the position of the cubes. The last question asked whether the depth visualization tool was helpful in completing the task. 33% of users indicated that FCP was easiest to use, 61% found that FBCP was easiest to use, and 5% indicated that both were equally easy. 83% said that FBCP gave them the best understanding about how the cubes were positioned, 5% indicated FCP, 5% said that both gave them equally good understanding, and 5% indicated "neither". 89% said that the depth visualization tool was helpful, although some users mentioned informally that without it they wouldn't be able to localize the two cubes with a gap in-between. 11% answered that the tool was not helpful.

5.2 Experiment 2

In the second experiment we compared different techniques that could help users deduce the distance relationship between structures visible on multimodal datasets. The focus dataset consisted of three large cubes, three large spheres and multiple smaller spheres and cubes of randomized color (red or blue) positioned at the same depth. Users were asked to identify the largest cube closest to both the largest sphere and one of the flowers visible on the context dataset. Smaller objects and colors were used as distracting factors, while users were expected to concentrate on the largest objects only. This experiment was repeated with four randomly positioned targets within each block.

Four visualization techniques were compared. The first technique utilized a focus + context visualization with a small focus region (SL) – Figure 5-3a. The second technique used a focus + context visualization with a large focus region (LL) - Figure 5-3b. The other two techniques employed a small lens with additional focus-driven context visualization. In the first such technique context and focus-driven context were merged with a weight *w* of 0.35 so that focus objects were clearly seen (Tr1) – Figure 5-3c. The fourth technique used a weight *w* of 0.05 which resulted in less visible focus targets (Tr2) – Figure 5-3d. In both Tr1 and Tr2 the trace was visible for around 4 seconds. The accuracy for each target position was evaluated as correct or incorrect based on the distance between the saved mouse position and the projected correct target position.

We hypothesized that the focus-driven context techniques would be faster and more accurate in target finding. We expected to see performance differences between Tr1 and Tr2 given their different weights *w*. Furthermore, differences between SL and LL were anticipated since SL has very limited visible focus region. Of particular interest was the comparison between the focus-driven context and large lens conditions; both visualizations can show large focus regions, but the shape of these regions vary between the two techniques.

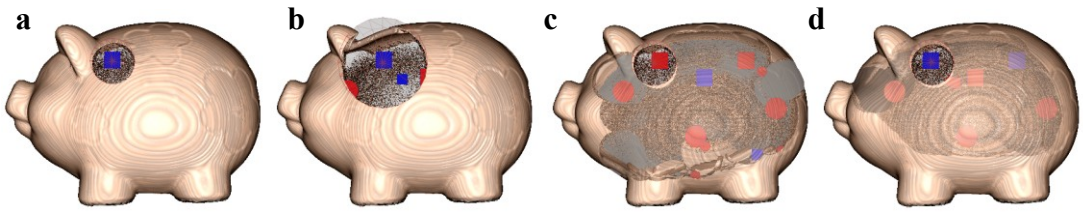

Figure 5-3: The visualization techniques compared in experiment 2. (**a**) Small lens (SL); (**b**) Large lens (LL); (**c**) Focus-driven context (w=0.35): Tr1; (**d**) Focus-driven context (w=0.05): Tr2.

The average response time for each user was calculated across the 4 trials per task. A one-way repeated measures analysis of variance (ANOVA) performed on these average

values showed a significant difference in response time ($F(3,51)=6.18$, $p=0.001$). The average response times for each technique as seen on Figure 5-4 are LL =12.93s; Tr1 =14.52s; Tr2 =16.46s; SL =17.32s. Pairwise post-hoc comparisons (two-tailed t-tests) showed that this effect was driven by significant differences between the SL and LL conditions ($t(17) = 3.49$, $p=0.003$), between SL and Tr1 conditions ($t(17)= 2.51$, $p=0.023$), and between the LL and Tr2 conditions ($t(17)=-2.88$, $p=0.010$). A one-way ANOVA performed on the average accuracy scores across 4 trials per technique showed no main effect of task, ($F(3, 51)= 0.14$; $p=0.94$). The average accuracy per technique was: LL =93.6% +/- 3%; Tr1 = 93.6 % +/- 4%; SL =91.7% +/- 3%; Tr2 = 91.2 % +/- 4%.

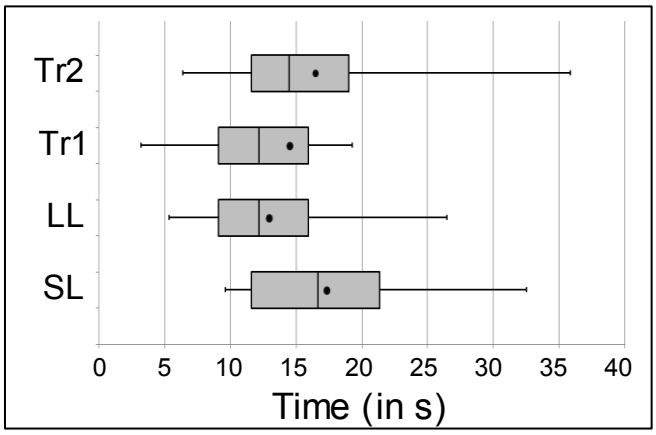

Figure 5-4: Box and whisker plot for response time for all four methods. Average time is represented by the dots.

Three questions were asked regarding this experiment in the paper survey. First, users were asked to choose between SL, LL and Trace (Tr1 and Tr2 were treated together) in terms of which technique gave them the best understanding of how all focus objects were positioned. 5% of users indicated that SL gave them the best understanding about the position of objects. 67% indicated LL and 39% indicated Trace. 61% of users said that LL was easiest to use versus 33% preference for Trace and 5% preference for SL. A

strong preference for a long visibility (4 seconds or more) of the trace (78%) was observed when compared to 22% preference for shorter visibility.

5.3 Experiment 3

In the third experiment we compared current techniques used in volume sculpting to the proposed technique of creating an arbitrary-shaped focus region using the focus lens. Users were asked to remove structures starting from a small white cube visible on the surface of the context dataset (Figure 5-5a) until they reached a larger white cube which was the target. Users were told to stop removing structures as soon as they saw the target object so that they don't cut through it. The experiment was repeated with three randomly positioned targets within each block.

Two types of sculpting tools were compared. The first one was an opaque tool that would not allow users to see what lies ahead of it (OD) - Figure 5-5b. The other was a lens-based sculpting tool that allows users to see what lies beyond the tip of the tool (TD) - Figure 5-5c. The tool position was controlled with the mouse, and cutting out material was executed by a key press. Accuracy computations were performed by computing the distance between the saved depth position of the front face of the target cube and the depth position of the user-controlled sculpting tool. Negative distance corresponded to the tool position being in front of the target. Positive distances represented how much of the target object had been cut out.

We hypothesized that there would be a significant response time and accuracy improvement of TD over OD. We believe that having the capability of seeing what objects lie ahead of the tool's position would improve not only the accuracy of the

method, but also the response time, since the transparent tool does not have to be moved away in order to validate the position of the target object.

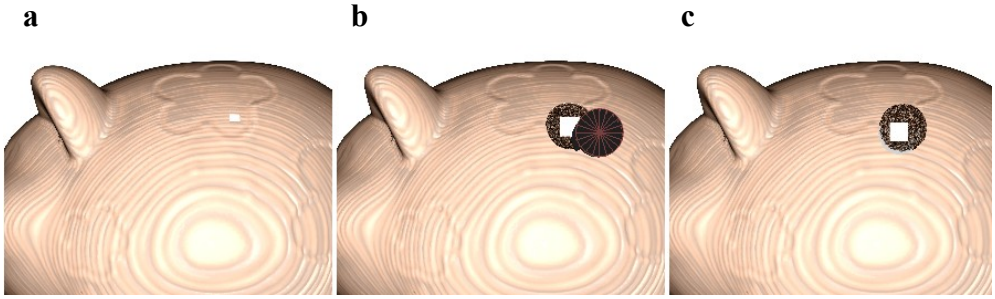

Figure 5-5: Volume sculpting starting at the white cube in (**a**), until the target is clearly seen using an opaque tool (**b**) and a transparent tool (**c**).

The depth position of the target objects was varied throughout the experiment and the blocks. Targets that are close to the surface will take less execution time than targets deeper inside the context dataset. When considering the second repetition, OD had two targets positioned near the surface of the context dataset and one target placed deep, while TD had one target positioned near the surface and two targets placed deep inside the dataset. This different ratio of deep vs. shallow targets would impact the response time comparison. Therefore we randomly excluded response time and accuracy data corresponding to one shallow target from OD and one deep target from FD per user.

When comparing response time by means of a two-tailed t-test, the transparent drill yielded significantly faster average responses (t_{avg}=16.1s) than the opaque drill (t_{avg}=19.6s) as seen in Figure 5-6a ($t(17)=2.352$, $p=0.031$). This showed that TD resulted in significantly faster performance than the method currently used in ablation visualizations.

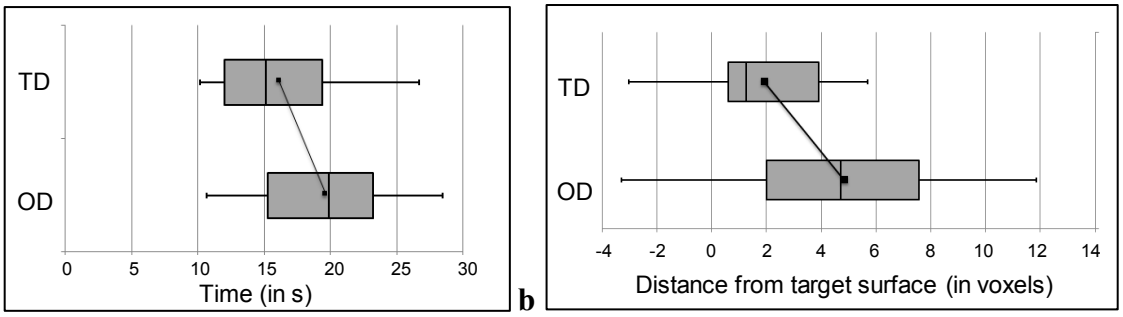

Figure 5-6: Box and whisker plot for response time (**a**) and accuracy (**b**) for TD and OD. Average time is represented by the dots

The depth of the target cube was 20 voxels, thus any accuracy error distance greater than 20 voxels corresponded to the complete removal of the target structure. When using OD, two users advanced the tool too fast, resulting in drilling through almost the whole target cube (error > 15 voxels), resulting in a large positive error for OD - Figure 5-6b.

On average TD resulted in more accurate performance than OD. The average error for TD was 1.92 voxels, which corresponded to barely cutting through the surface of the target. OD, in contrast, gave an average distance error of 4.85 voxels, which means that approximately one-fourth of the target cube was cut out. A paired-sample t-test with respect to accuracy confirmed that performance using TD was significantly better than OD ($t(17)=2.844$, $p=0.011$).

Two questions regarding the drilling interface were asked, both of them having two possible choices – OD or TD. Users were asked which technique was easiest to use and which technique allowed them to be more accurate. 94% of users said that TD was easiest to use, and 6% indicated OD. 89% said that TD allowed them to be more accurate, versus 11% preference for OD.

5.4 Discussion

Experiment 1 showed that FCP resulted in slightly faster performance (though not reliably so) than a technique that requires the adjustment of two clipping planes. The unexpected small response time difference between the two techniques and the marginally higher accuracy of FBCP suggests an advantage of FBCP over FCP. While most users spent a large portion of their time adjusting the two clipping planes from the FBCP to accurately find the cubes with a gap, more time than we expected was required to correctly deduce the depth relationship between the cubes when using the FCP technique. This suggests that FCP requires a lot of concentration and provides limited spatial and structural information, as supported by the lower average accuracy results per user. In contrast, FBCP allowed users to explore smaller sections of the dataset in order to correctly understand the structural dependencies between objects without requiring high concentration.

Focus-driven context outperformed the small lens with respect to time, showing that medical cases where the lens is expected to be small can benefit from focus-driven context. A large lens and focus-driven context provided comparable accuracy. However, performance using focus-driven context was slower, perhaps because the visibility of context features inside the focus-driven context area was reduced and the time it took until this new region disappears was relatively long. More complex focus datasets and the requirement for the visibility of more than two datasets should also be tested in the future to evaluate whether focus-driven context would show significant advantage over a large lens. Moreover, the observed significant difference in accuracy between SL and LL indicates that focus+ context visualizations should provide user control over the lens size.

We would like to point out the significant time and accuracy improvements in the tissue removal task when using a magic lens as a sculpting tool. The measured variables, together with the survey and the observed interaction with the system, show that a transparent sculpting tool can greatly contribute to a faster and more accurate incision path trajectory planning. This would be especially beneficial for novices who are not comfortable with the anatomy being removed. In this case, clear visibility of the structures in front of the tool should improve performance.

Chapter 6: Conclusions and future work

Most visualization techniques for the study of medical data in two dimensions and three dimensions do not address the demands for different types of visualizations tailored to certain medical specialists. Radiologists are trained to look at series of 2D slices and infer the volumetric structure of the datasets. They also have very strong spatial perception cognition and can deduce distances between anatomical objects visible on different slices. Radiologists resort to 3D volumetric visualizations when they have to work with multimodal datasets or when the target object is of a highly complex nature in order to place an accurate diagnosis. On the other hand, surgeons prefer to examine 2D slices oriented along the path of a possible planned incision. When using 3D visualizations, they choose to view the target volume from specific perspective positions that resemble the anatomical appearance in the operating room. Such visualizations are often used for surgical planning and procedure training. However, no existing system addresses the difference in used visualizations, the lack of a visualization technique that can be employed by all physicians, and how to unify existing techniques into such a visualization.

To address this problem we examined 2D, 3D and focus + context techniques used in medical data visualizations to create a generalized temporal focus + context framework that can be used to classify existing visualizations and to develop new ones. The framework provides a novel unique and easy to use way of supporting visualizations for dataset exploration and procedure planning even when multiple users interact with the system simultaneously. Most of the existing and employed by physicians techniques can be described and implemented by the framework so that the user can quickly choose and

adapt the technique most suitable for the task in mind. New techniques that can combine the advantages of 2D and 3D visualizations can be developed within the unifying framework definition. The creation of new visualization techniques based on the focus + context paradigm require additional improvements of the used raycasting rendering technique so that focus regions of complex shapes can be correctly displayed. We combine depth-peeling with a novel improved method for parity computation in order to correctly define the regions traversed by the rays and maintain interactive rendering speeds.

Our framework has been used to create a visualization technique where the user can scroll through slices that fall within the lens geometry. Enabling a 3D sub-volume rendering inside the lens that renders only the volume defined by an interactively controlled number of slices provides more inner structure information of the target anatomy. We also apply the described generalized framework to improve the explicit spatial perception relationship between the current and past focus areas. This creates a new hierarchy of context regions that can be used for the display of different modality datasets, while the focus region displays any rendering modality. These new framework-based visualizations can be used by both radiologists and surgeons in dataset exploration and volume visualization in axis-aligned or arbitrary angles while maintaining explicit spatial relationship between different modality datasets and visualization techniques.

The temporal focus + context framework is also used to identify two new ways of helping surgeons in the visualization of surgical procedures when they explore optimal resection paths. We define an arbitrary-shaped focus that can be used to show focus structures of concave shape and to visualize volume sculpting. We also provide a

technique that is used as a virtual endoscope, but with the added advantage of showing what lies beyond the hollow structure used to traverse the camera. Thus, the new and existing visualizations can be combined and used simultaneously by surgeons and radiologists during medical team meetings to present and examine the datasets in 2D and 3D, as well as to accurately identify and evaluate possible intervention paths.

The applicability of the generalized framework in accurate target finding has been evaluated by performing a user study. The study showed significant accuracy and response time improvements of our arbitrary-shaped focus region visualization when compared to currently used techniques for volume sculpting. Furthermore, a comparison of four visualizations showed that focus-driven context helps users correctly infer distances between objects by providing explicit spatial relationship information when compared to a small lens. Moreover, the study asserted that visualizations should be able to support focus lenses of varying size. Accuracy in understanding the structural composition of complex objects was shown to be higher when users explore sub-section of the whole target object.

The described framework and the combination of different visualization techniques within the focus region can thus lead to a more precise view of the abnormalities involved, augment multidisciplinary communication, and help to further define the nature and extent of pathology or the results of therapy. This may also be useful for the surgical planning and postoperative localization of medical implants and devices. In addition, this approach may have valuable applications beyond the clinical realm, including improved visualizations for scientists in more distant fields, medico-legal illustration modes, and teaching anatomy and pathology.

Future work involves the design and execution of a second user study aimed at clinicians. Dataset exploration and surgical planning tasks using medical datasets (MRI and CT) will be provided in order to evaluate the execution time and accuracy when using the described framework techniques. The framework definition can also be used for the development of novel techniques that explore the use of multiple dependent lenses that filter data or display time-dependent datasets.

References

[1] Hoppenrath, M., BS, ELS; 2004. "Applied radiology focus: understanding multimodal fusion imaging". *Journal of Applied Radiology*, **33**(6).

[2] Kainz, B., Portugaller, RH., Seider, D., Moche, M., Stiegler, P., and Schmalstieg, D.; 2011. "Volume visualization in the clinical practice". *In Proceedings of the 6th international conference on Augmented Environments for Computer-Assisted Interventions* (AE-CAI'11), Springer-Verlag, Berlin, Heidelberg, pp. 74-84, doi: 10.1007/978-3-642-32630-1_8.

[3] Lorensen, WE., and Cline, HE.; 1987. "Marching cubes: a high resolution 3D surface construction algorithm". *SIGGRAPH Computer Graphics*, **21**(4), pp. 163-169, doi: 10.1145/37402.37422.

[4] Slicer3D. Multiplatform, free and open source software package for visualization and medical image computing. Available at: www.slicer.org/. Accessed on Oct 09, 2013.

[5] Tory, M., and Swindells, C.; 2002. "Exovis: an overview and detail technique for volumes". Technical Report SFU-CMPTTR2002-05, Computing Science Dept., Simon Fraser University.

[6] Bruckner, S. and Groller, ME.; 2006. "Exploded views for volume data". *IEEE Transactions on Visualization and Computer Graphics,* **12**(5), pp. 1077-1084, doi: 10.1109/TVCG.2006.140.

[7] Weiskopf, D., Engel, K., and Ertl, T.; 2003. "Interactive clipping techniques for texture-based volume visualization and volume shading". *IEEE Transactions on Visualization and Computer Graphics*, **9**(3), pp. 298-312, doi: 10.1109/TVCG.2003.1207438.

[8] Correa, C., Silver, D., and Chen, M.; 2006. "Feature aligned volume manipulation for illustration and visualization". *IEEE Transactions on Visualization and Computer Graphics* September 2006; **12**(5), pp. 1069-1076, doi: 10.1109/TVCG.2006.144.

[9] Viega, J., Conway, MJ., Williams, G., and Pausch, R.; 1996. "3D magic lenses". *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pp. 51-58, doi:10.1145/237091.237098.

[10] Kirmizibayrak C: "Interactive volume visualization and editing methods for surgical applications". Ph.D. Dissertation, Department of Computer Science, The George Washington University, Washington, DC, 2011.

[11] Kirmizibayrak, C., Yim, Y., Wakid, M., and Hahn, J.; 2012. “Interactive visualization and analysis of multimodal datasets for surgical applications”. *Journal of Digital Imaging,* **25**(6), pp. 792-801, doi: 10.1007/s10278-012-9461-y.

[12] Hornak, J.P., “The Basics of MRI”. 1996-2011, http://www.cis.rit.edu/htbooks/mri/index.html, Site accessed: 02/23/2013.

[13] Santesoft, Medical imaging software, The orthogonal MPR window, http://www.santesoft.com/howto/mpr.html, Site accessed: 02/12/2013.

[14] Chernyaev, EV.; 1995. “Marching cubes 33: construction of topologically correct isosurfaces”. Technical Report CN/95-17.

[15] Gueziec, A., and Hummel, R.; 1995. "Exploiting triangulated surface extraction using tetrahedral decomposition". *IEEE Transactions on Visualization and Computer Graphics,* **1**(4), pp. 328-342, doi: 10.1109/2945.485620.

[16] Johansson, G. and Carr, H., 2006. “Accelerating marching cubes with graphics hardware”. *In Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research* (CASCON '06). IBM Corp., Riverton, NJ, USA, Article 39, doi: 10.1145/1188966.1189018.

[17] Goetz, F., Junklewitz, T., and Domik, G., 2005. “Real-time marching cubes on the vertex shader”. *In Eurographics* 2005, pp. 5-8.

[18] Jing, G. and Song, W.; 2008. “An octree ray casting algorithm based on multi-core CPUs”. *In Proceedings of the 2008 International Symposium on Computer Science and Computational Technology - Volume 02* (ISCSCT '08). IEEE Computer Society, Washington, DC, USA, pp. 783-787, doi: 10.1109/ISCSCT.2008.207.

[19] Levoy, M.; 1988. “Display of surface from volume data”. *IEEE Computer Graphics and Applications*, **8**(3), pp. 29-37, doi: 10.1109/38.511.

[20] Haidacher, M., Bruckner, S., Groller, E.; 2011. "Volume analysis using multimodal surface similarity”. *IEEE Transactions on Visualization and Computer Graphics,* **17**(12), pp. 1969-1978, doi: 10.1109/TVCG.2011.258.

[21] Bramon, R., Ruiz, M., Bardera, A., Boada, I., Feixas, M., Sbert, M.; 2013. "Information theory-based automatic multimodal transfer function design". *IEEE Journal of Biomedical and Health Informatics,* **17**(4), pp. 870-880, doi: 10.1109/JBHI.2013.2263227.

[22] Jung, Y., Kim, J., Eberl, S., Fulham, M., & Feng, DD.; 2013. “Visibility-driven PET-CT visualisation with region of interest (ROI) segmentation”. *Vis. Comput.,* **29**(6-8), pp. 805-815. doi: 10.1007/s00371-013-0833-1.

[23] GEHC MicroView, Parallax Innovations, Last updated on Feb 22, 2012, http://microview.sourceforge.net/, Site accessed: Nov 5, 2012.

[24] Zhao, Y., Clapworthy, G.J.; Tao, Y., Dong, F., Wei, H., and Wang, T.; 2011. "CCVis: a software plugin for unified visualisation in ContraCancrum based on VTK extensions”. *15th International Conference on Information Visualisation (IV), 2011*, pp. 610-615, doi: 10.1109/IV.2011.80.

[25] Neuroimage Analysis Center**:** "understanding the human brain through imaging”, NAC as a Critical Link, http://nac.spl.harvard.edu/pages/Overview, Site accessed: 02/12/2013.

[26] Kohlmann, P., Bruckner, S., Kanitsar, A., and Groller, ME.; 2008. “LiveSync++: enhancements of an interaction metaphor”. *In Proceedings of Graphics Interface 2008*, pp. 81–88.

[27] Wiebel, A., Vos, FM., Foerster, D., and Hege, H.; 2012. "WYSIWYP: What You See Is What You Pick”. *IEEE Transactions on Visualization and Computer Graphics*, **18**(12), pp. 2236-2244, doi: 10.1109/TVCG.2012.292.

[28] Tietjen, C., Meyer, B., Schlechtweg, S., Preim, B., Hertel, I., and Strauß, G.; 2006. "Enhancing slice-based visualizations of medical volume data”. *In Proceedings of EuroVis*, 2006, pp. 123-130, doi: 10.2312/VisSym/EuroVis06/123-130.

[29] König, A., Doleisch, H., and Gröller, ME.; 1999. “Multiple views and magic mirrors - fMRI visualization of the human brain”. TR-186-2-99-08.

[30] Bier, EA., Stone, MC., Pier, K., Buxton,W., and DeRose, TD.; 1993. “Toolglass and magic lenses: the see-through interface”. Proceedings of SIGGRAPH '93, doi: 10.1145/166117.166126.

[31] Hurter, C., Telea, A., and Ersoy. O.; 2011. “MoleView: an attribute and structure-based semantic lens for large element-based plots”. *IEEE Transactions on Visualization and Computer Graphics*, **17**(12), pp. 2600-2609, doi: 10.1109/TVCG.2011.223.

[32] Sarkar, M., and Brown, MH.; 1992. “Graphical fisheye views of graphs”. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '92), Penny Bauersfeld, John Bennett, and Gene Lynch (Eds.). ACM, New York, NY, USA, pp. 83-91, doi: 10.1145/142750.142763.

[33] Pindat, C., Pietriga, E., Chapuis, O., and Puech, C.; 2012. “JellyLens: content-aware adaptive lenses”. *In Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, pp. 261-270, doi: 10.1145/2380116.2380150.

[34] Keahey, TA.; 1998. “The generalized detail-in-context problem”. *Proc. of the IEEE Symposium on Information Visualization*, pp. 44-51, doi: 10.1109/INFVIS.1998.729558.

[35] Carpendale, MST., Cowperthwaite, DJ., and Fracchia, FD.; 1995. “3-dimensional pliable surfaces: for the effective presentation of visual information”. *Proc. of UIST*, pp. 217 – 226, doi: 10.1145/215585.215978.

[36] Keahey, T., and Robertson, E.; 1996. “Techniques for nonlinear magnification transformations”. *Proc. of the IEEE Symposium on Information Visualization*, pp. 38-45, doi: 10.1109/INFVIS.1996.559214.

[37] Robertson, GG., and Mackinlay, JD.; 1993. “The document lens”. *Proc. of the ACM Symposium on User Interface Software and Technology (UIST)*, pp. 101-108, doi: 10.1145/168642.168652.

[38] Svakhine, N., Ebert, DS., and Stredney, D.; 2005. “Illustration motifs for effective medical volume illustration”. *IEEE Comput. Graph. Appl.* **25**(3), pp. 31-39. doi: 10.1109/MCG.2005.60

[39] Kruger, J., Schneider, J., and Westermann, R.; 2006. "ClearView: an interactive context preserving hotspot visualization technique". *IEEE Transactions on Visualization and Computer Graphics*, **12**(5), pp. 941-948, doi: 10.1109/TVCG.2006.124.

[40] Yang, Y., Chen, JX., and Beheshti, M.; 2005. “Nonlinear perspective projections and magic lenses: 3D view deformation”. *IEEE Comput. Graph. Appl.*, **25**(1), pp. 76-84, doi: 10.1109/MCG.2005.29.

[41] LaMar, E., Hamann, B., and Joy, KI.; 2001. “A magnification lens for interactive volume visualization”. *Proc. 9th Pacific Conf. Computer Graphics and Applications* (PG 01), pp. 223-233, doi: 10.1109/PCCGA.2001.962877.

[42] Wang, L., Zhao, Y., Mueller, K., Kaufman, A.; 2005. "The magic volume lens: an interactive focus+context technique for volume rendering”. *Visualization, 2005. IEEE VIS 05*, pp. 367-374, doi: 10.1109/VISUAL.2005.1532818.

[43] Zhao, X., Zeng, W., Gu, D., Kaufman, A., Xu, W., and Mueller, K.; 2012. "Conformal magnifier: a focus + context technique with local shape preservation". *IEEE Transactions on Visualization and Computer Graphics* 2012; vol. 99, no. PrePrints, doi: 10.1109/TVCG.2012.70.

[44] Li, B., Zhao, X. and Qin, H.; 2012. "Four-dimensional geometry lens: a novel volumetric magnification approach". *Computer Graphics Forum*. **31**(7), pp. 1-12, doi: 10.1111/cgf.12166

[45] Cowperthwaite, DJ.; 2000. "Occlusion resolution operators for three-dimensional detail-in-context". *Ph.D thesis, Simon Fraser University*.

[46] Viola, I., Kanitsar, A., Groller, ME.; 2004. "Importance-driven volume rendering". *In Proceedings of the conference on Visualization '04 (VIS '04)*. IEEE Computer Society, Washington, DC, USA, 2004, pp. 139-146, doi: 10.1109/VISUAL.2004.48.

[47] Hauser, H., Mroz, L., Italo Bischi, G., Groller, ME.; 2001. „Two-level volume rendering". *IEEE Trans. Vis. Comput. Graphics*, **7**(3), pp. 242-252, doi: 10.1109/2945.942692.

[48] Luo, Y.; 2012. "Distance-based focus + context models for exploring large volumetric medical datasets". *Comput Sci Eng*, **14**(5), pp. 63-71, doi: 10.1109/MCSE.2011.114.

[49] Sikachev, P., Rautek, P., Bruckner, S., Gröller, ME.; 2010. "Dynamic focus+context for volume rendering". *In Proceedings of Vision, Modeling and Visualization*, 2010, pp. 331-338.

[50] Ropinski, T., Viola, I., Biermann, M., Hauser, H., and Hinrichs, K.; 2009. "Multimodal visualization with interactive closeups". *In Proceedings of EG UK Theory and Practice of Computer Graphics*, Cardiff University, United Kingdom, 2009, pp. 17-24.

[51] Diepenbrock, S., Praßni, JS., Lindemann, F., Bothe, HW., and Ropinski, T.; 2011. "Interactive visualization techniques for neurosurgery planning". EUROGRAPHICS, 2011.

[52] Rieder, Ch., Ritter, F., Raspe, M., and Peitgen, H-O.; 2008. "Interactive visualization of multimodal volume data for neurosurgical tumor treatment". *Eurographics/ IEEE-VGTC Symposium on Visualization*, **27**(3), pp. 1055-1062, doi: 10.1111/j.1467-8659.2008.01242.x.

[53] Burns, M., Haidacher, M., Wein, W., Viola, I., and Gröller, ME.; 2007. "Feature emphasis and contextual cutaways for multimodal medical visualization". *In Proceedings of the 9th Joint Eurographics / IEEE VGTC conference on Visualization* (EUROVIS'07). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 275-282, doi: 10.2312/VisSym/EuroVis07/275-282.

[54] Luo, Y., Jose Antonio, Guitián, I., Gobbetti, E., and Marton, F.; 2009. "Context preserving focal probes for exploration of volumetric medical datasets". *In Proceedings of the 2009 international conference on Modelling the Physiological Human* (3DPH'09), Springer-Verlag, Berlin, Heidelberg, pp. 187-198, doi: 10.1007/978-3-642-10470-1_16.

[55] Fuhrmann, A., and Groeller, E.; 1998. "Real-time techniques for 3D flow visualization". *In Proceedings of the conference on Visualization '98 (VIS '98)*. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 305-312, doi: 10.1109/VISUAL.1998.745317.

[56] Gasteiger, R., Neugebauer, M., Beuing, O., and Preim, B.; 2011. "The FLOWLENS: a focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms". *IEEE Transactions on Visualization and Computer Graphics,* **17**(12), pp. 2183-2192, doi: 10.1109/TVCG.2011.243.

[57] Rossler, F.; Botchen, RP.; Ertl, T.; 2008. "Dynamic shader generation for GPU-based multi-volume ray casting". *IEEE Computer Graphics and Applications,* **28**(5), pp. 66-77, doi: 10.1109/MCG.2008.96.

[58] Plate, J.; 2007. "A flexible multi-volume shader framework for arbitrarily intersecting multi-resolution datasets". *IEEE Transactions on Visualization and Computer Graphics*, **13** (6), pp. 1584 – 1591, doi: 10.1109/TVCG.2007.70534.

[59] Borst, CW.; Tiesel, JP.; Best, CM.; 2010. "Real-time rendering method and performance evaluation of composable 3D lenses for interactive VR". *IEEE Transactions on Visualization and Computer Graphics*, **16**(3), pp. 394-410, doi: 10.1109/TVCG.2009.89.

[60] Larson, AM., and Loschky, LC.; 2009. "The contributions of central versus peripheral vision to scene gist recognition". *Journal of Vision*, **9**(10): 6, doi: 10.1167/9.10.6.

[61] van Asselen, M., and Castelo-Branco, M.; 2009. "The role of peripheral vision in implicit contextual cuing". *Attention, Perception, & Psychophysics*, **71**(1), pp. 76-81, doi:10.3758/APP.71.1.76.

[62] Schott, M., Grosset, AVP., Martin, T., Pegoraro, V., Smith, ST., and Hansen, CD.; 2011. “Depth of field effects for interactive direct volume rendering”. *In Proceedings of the 13th Eurographics / IEEE - VGTC conference on Visualization* (EuroVis'11), Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 941-950. DOI: 10.1111/j.1467-8659.2011.01943.x

[63] Grosset, P., Schott, M., Bonneau, GP., and Charles, H.; 2013. “Evaluation of depth of field for depth perception in DVR”. *IEEE Pacific Visualization 2013, pp.* 81-88, doi: 10.1109/PacificVis.2013.6596131.

[64] Gooch, A., Gooch, B., Shirley, PS., and Cohen, E.; 1998. “A non-photorealistic lighting model for automatic technical illustration”. *In Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pp. 447-452, doi: 10.1145/280814.280950.

[65] Lum, EB., and Ma, KL.; 2002. “Hardware-accelerated parallel non-photorealistic volume rendering”. *In Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (NPAR '02). ACM, New York, NY, USA, pp. 67-ff. doi: 10.1145/508530.508542

[66] Svakhine, NA., Ebert, DS., and Andrews, WM.; 2009. "Illustration-inspired depth enhanced volumetric medical visualization". *IEEE Transactions on Visualization and Computer Graphics*, **15**(1), pp. 77-86, doi: 10.1109/TVCG.2008.56

[67] Bruckner, S.; and Groller, ME.; 2007. "Enhancing depth-perception with flexible volumetric halos". *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), pp. 1344-1351, doi: 10.1109/TVCG.2007.70555

[68] Tao, Y., Lin, H., Dong, F., and Clapworthy, G.; 2011. “Opacity volume based halo generation for enhancing depth perception”. *In Proceedings of the 2011 12th International Conference on Computer-Aided Design and Computer Graphics* (CADGRAPHICS '11). IEEE Computer Society, Washington, DC, USA, pp. 418-422. doi: 10.1109/CAD/Graphics.2011.81

[69] Ropinski, T.; Steinicke, F. and Hinrichs, KH.; 2006. “Visually supporting depth perception in angiography imaging”. *In Proceedings of the 6th International Symposium on Smart Graphics (SG06)*, Springer-Verlag, Vancouver , pp. 93-104, doi: 10.1007/11795018_9.

[70] Radeva N., Levy L., and Hahn J.; 2014. ”Generalized temporal focus + context framework for improved medical data exploration”. *Accepted for publication in Journal of Digital Imaging*.

[71] Petersik, A., Pflesser, B., Tiede, U., Höhne, KH. and Leuwer, R.; 2003. "Realistic haptic interaction in volume sculpting for surgery simulation". *In Proceedings of the 2003 international conference on Surgery simulation and soft tissue modeling* (IS4TM'03), Springer-Verlag, Berlin, Heidelberg, pp. 194-202.

[72] Morris, D., Sewell, C., Barbagli, F., Blevins, N., Girod, S., and Salisbury, K.; 2006. "Visuohaptic simulation of bone surgery for training and evaluation". *IEEE Computer Graphics and Applications*. **26**(4), pp. 48-57, doi: 10.1109/MCG.2006.140.

[73] Sorensen, MS., Mosegaard, J., and Trier, P.; 2009. "The visible ear simulator: a public PC application for GPU-accelerated haptic 3D simulation of ear surgery based on the visible ear data". *Otol Neurotol.,* **30**(4), pp. 484-487, doi: 10.1097/MAO.0b013e3181a5299b.

[74] Deligianni, F., Chung, AJ., and Yang, GZ.; 2006. "Nonrigid 2-D/3-D registration for patient specific bronchoscopy simulation with statistical shape modeling: phantom validation". *IEEE Transactions on Medical Imaging*, **25**(11), pp. 1462-1471, doi: 10.1109/TMI.2006.883452

[75] Lee, TY., Lin, PH., Lin, CH., Sun, YN., and Lin, XZ.; 1999. "Interactive 3-D virtual colonoscopy system". *IEEE Transactions on Information Technology in Biomedicine*, **3**(2), pp. 139-150, doi: 10.1109/4233.767089.

[76] Olwal, A., Frykholm, O., Groth, K., and Moll, J.; 2011. "Design and evaluation of interaction technology for medical team meetings". *In Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part I* (INTERACT'11), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, pp. 505-522.

[77] Groth, K., Frykholm, O., Segersvard, R., Isaksson, B., and Permert, J.; 2009. "Efficiency in treatment discussions: A field study of time related aspects in multi-disciplinary team meetings". *22nd IEEE International Symposium on Computer-Based Medical Systems 2009, pp.* 1-8, doi: 10.1109/CBMS.2009.5255318.

[78] Everitt, C.; 2001. "Interactive order-independent transparency". Technical report, NVIDIA Corporation

[79] Radeva N., and Hahn J.; 2014. "Visualization of tissue removal using focus + context techniques". *Accepted for presentation at NextMed/MMVR21*, 20-22 February 2014.

[80] Trapp, M., and Döllner, J.; 2008. "Efficient representation of layered depth images for real-time volumetric tests". In *TPCG, pp. 9-16, doi: 10.2312/LocalChapterEvents/TPCG/TPCG08/009-01.*

[81] Levoy M., and Rusinkiewicz S.; 2001. “Efficient variants of the ICP algorithm”. *In Third Int. Conf. on 3D Digital Imaging and Modeling (3DIM 2001)*, IEEE Comp. Soc., pp. 145–152.