

Enhancing Visual Perception in Interactive Direct Volume Rendering of Medical Images

by

AmirAli Sharifi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

© AmirAli Sharifi, 2016

Abstract

Accurate information is the foundation of correct clinical diagnoses. Physicians are increasingly relying on new devices and tools to improve the quality of information used in their decision making. Wrong, misleading, or hard to interpret data can prevent patients from receiving proper treatment, hence putting their lives in danger or negatively impacting their quality of life.

Medical imaging is one of the most widely used techniques in clinical diagnosis. It enables physicians to view the underlying anatomy of the patients. The information attained using medical imaging provides a non-invasive approach to detection of abnormalities and defects. Many imaging modalities are widely used today such as Computed Tomography (CT), and Magnetic Resonance Imaging (MRI). These imaging techniques in particular provide physicians with volumetric data representing the segment of interest from the internal anatomy of the patient.

An effective and accurate visualization of the volumetric data is one of the bottlenecks among many, that prevents physicians to take advantage of the full potential of the available information. Many visualization techniques such as creating a set of 2D slides, surface rendering, direct volume rendering (DVR), etc. has been used to make the volumetric information available for the consumption of the physicians. However, in the past 30 years, lack of proper algorithms as well as limitations of hardware have prevented us from using the full potential of methods such as DVR.

DVR creates 3D rendered images based on the volumetric data. Human brain is naturally trained to understand the spatial arrangement of objects using our visual system. A 3D rendered image simulates our vision. In contrast, understanding of the spatial arrangement, sizes, and shapes of objects using

2D slices is far less intuitive and requires many years of training. Moreover, DVR uses all of the volumetric data, while methods such as surface rendering only use a fraction of the data while discarding most of it.

Despite great advantages of DVR, its usage remained limited due to some of its shortcomings. DVR simulates the physical process by which the image of a 3D scene is formed on a 2D surface. However, it is not computationally viable to simulate all or even nearly enough rays of light to create naturally occurring phenomena such as depth of field (DoF). These natural phenomena are used by our brain alongside each other to provide us with visual cues to help us interpret the images we see.

In the first phase of this research we have created an algorithm that enables any portion of the volume to be rendered with the desired degree of blur in real-time using DVR. This algorithm is then utilized to tackle the lack of DoF problem in DVR. We have taken advantage of parallel programming and a novel sampling scheme to minimize the rendering time of DoF-enabled DVR.

In the second phase of this research we have used our blur algorithm to introduce two new synthetic visual cues to help physicians: Transfer Function Based Blurring (TFBB), and Focus-Based Color Coding (FBCC). The former technique augments transfer functions to control the degree of blur based on the material. This enables the physicians to create semantic depth of field, keeping the material of interest in focus while other blurred material provide context. FBCC provides a way to direct viewer's attention to the object of interest while clearly distinguishing objects of interest. The combination of depth of field, TFBB, and FBCC has also been examined and shown great potential to remove clutter from the scene.

The third phase of this research puts our previous two contributions through a rigorous usability study, where physicians are presented by images rendered by our methods and their interpretation of these images as well as many parameters of their interaction are recorded and evaluated. Our usability study showed that our proposed methods provide physicians with an enhanced visualization that has the potential to significantly improve correct clinical decision making.

Preface

This thesis is an original work by AmirAli Sharifi. The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Project Name “Assessment of newly developed synthetic visual cues on visual perception in CT”, No. Pro00054433, February 23rd, 2015.

To my mother, Sedigheh

There are no incurable diseases - only the lack of will.

– Avicenna

Acknowledgements

I am very grateful to my supervisor Dr. Pierre Boulanger. His continuous support and knowledge granted me the opportunity to work in this very interesting area of research. His wisdom and guidance throughout these years paved the way to overcome many great challenges.

I would like to express my gratitude to Dr. Michelle Noga. Without her expertise and support from the beginning until the very end of this work, it would have been next to impossible to finish this dissertation.

I am very grateful to my supervisory committee Dr. Kumaradevan Punithakumar, Dr. Irene Cheng, and Dr. Michelle Noga for their support, guidance, and feedback along the way.

I am hugely indebted to my family specially my mother. They felt all the challenges as I was facing them and supported me during each single step of the way. I dedicate this thesis to my mother, Sedigheh, to whom I owe everything in my life.

I also need to thank my friend and colleague Dr. Levi Lelis whose support and care was a big help during the work on this dissertation.

Table of Contents

1	Introduction	1
2	Background	5
2.1	Volume Rendering	5
2.2	Image-Order Methods	6
2.2.1	Ray Casting	6
2.2.2	Speed Optimization for Ray Casting	8
2.2.3	Hardware Acceleration for Ray Casting	12
2.3	Object-Order Methods	14
2.3.1	Volume Splatting	14
2.4	Other Volume Rendering Methods	15
2.5	Visual Perception in Volume Rendering	16
2.5.1	Directing Viewers' Attention	17
2.5.2	Depth Perception in Volume Rendering	18
2.5.2.1	Depth-of-Field	19
2.5.3	Volume Illumination	24
2.5.4	Volumetric Halos	27
2.6	Quality Enhancement	29
2.7	Transfer Functions	31
2.7.1	Pre-Integrated Classification	32
3	Depth-of-Field Using Stochastic Sampling	33
3.1	Simulating Out of Focus Blur in 3D	34
3.1.1	Using 2D Kernels in DVR	34
3.1.1.1	Kernel Sampling	35
3.1.2	Neighborhood Sampling	36
3.1.3	Permutation-based Selection of a Single $k(i, j, t)$	37
3.1.4	Central Reflection Sampling	39
3.1.5	Calculating Sampling Locations	40
3.2	Fast Sampling of the Permutation Space	42
3.2.1	Seed Permutations	42
3.2.2	Set of Prime Numbers	43
3.2.3	Generating Permutations	44
3.2.4	Transition Between Different Blur Levels	45
3.2.5	Varying Blur Levels for Simulating Depth-of-Field	46
3.3	Blurred Edge Highlighting	48
3.4	Discussion of the Proposed Model	49
3.5	Implementation	50
3.6	Results	51
3.7	Conclusion	59
4	Directed Visual Perception	61
4.1	Method	61
4.1.1	Transfer Function Based Blurring	62

4.1.2	Focus-Based Chroma-coding	64
4.1.3	Combinations	65
4.2	Implementation	66
4.2.1	Implementation of TFBB	67
4.2.2	Implementation of FBCC	67
4.3	Results	68
4.4	Conclusion	75
5	Usability Study	78
5.1	Evaluation Targets	78
5.2	Study Design	80
5.2.1	Data Extraction	81
5.2.1.1	Primary & Secondary Questions	81
5.2.1.2	Abnormality Marking and Error Threshold	82
5.2.1.3	Time	84
5.2.2	Usability Study Steps	84
5.2.3	Volunteers	89
5.2.4	Factors Against Our Usability Study Results	90
5.2.4.1	Lack of Motion Parallax	90
5.2.4.2	No Spatial Orientation	90
5.2.4.3	Concurrent Availability of Methods	91
5.2.4.4	Lack of Previous Training	91
5.3	Results	91
5.3.1	Abnormality Assessment	94
5.3.1.1	Correct Assessment	96
5.3.1.2	False Positive Assessment	97
5.3.1.3	False Negative Assessment	99
5.3.2	Correctness of Marking	101
5.3.2.1	Correct Marking	103
5.3.2.2	Correct Assessment - Wrong Marking	106
5.3.2.3	Combined Missed Abnormalities	108
5.3.3	Distance of Marked Location	110
5.3.4	Time	114
5.3.4.1	Times To Answer Primary Questions	115
5.3.4.2	Times to Answer Primary Questions Correctly	116
5.3.5	Discussion	116
5.3.5.1	$\langle TFBB, NCC \rangle$	118
5.3.5.2	$\langle TFBB, RFBO+ \rangle$	120
5.3.5.3	$\langle TFBB, YFCO \rangle$	120
5.3.5.4	$\langle TFBB, DoF, NCC \rangle$	122
5.3.5.5	$\langle TFBB, DoF, RFBO+ \rangle$	124
5.3.5.6	$\langle TFBB, DoF, YFCO \rangle$	129
5.3.5.7	Overall Comparison	129
5.4	Conclusion	130
6	Conclusion	131
	Bibliography	133
	A Usability Study Consent Form	139
	B Test Taker Instructions	143

List of Tables

5.1	Participating Methods in the Usability Study	79
5.2	Our visualization methods compared to the reference $\langle FF, NCC \rangle$	119

List of Figures

2.1	Three states of leap-buffer for empty space skipping using coherence between neighboring rays.	12
2.2	An example of growing boxes	13
2.3	Simulating depth-of-field effect for slice-based direct volume rendering.	24
2.4	Illustration of circles of confusion.	25
2.5	Single-pass approach to create shadows for volume splatting.	26
2.6	A comparison between a volume rendered with halos and without halos.	28
2.7	Results of virtual sampling.	30
2.8	Slab-based rendering with quality enhancement.	31
3.1	Choices for sampling locations for kernel stretching.	36
3.2	Sections of a ray for permutations-based sampling.	38
3.3	Illustration of central inversion sampling.	40
3.4	Vector model for calculating actual coordinates of sampling locations.	41
3.5	Storing n different randomized permutation seeds in an array of size n_{max}^2	43
3.6	Transition between different blur levels.	46
3.7	The thin lens model	47
3.8	Comparison of permutation based sampling with pseudo-random sampling.	53
3.9	Results from our depth of field method: Cardiovascular System	54
3.10	Results from our depth of field method: Cardiovascular System	55
3.11	Results from our depth of field method: Knees	56
3.12	Comparison of quality and speed	57
3.13	Magnification of quality and speed comparison	58
4.1	Results of using TFBB, DoF, and FBCC simultaneously	63
4.2	Results of using TFBB, DoF, and FBCC simultaneously	69
4.3	Results of using TFBB, DoF, and FBCC simultaneously	72
4.4	Results of using TFBB, DoF, and FBCC simultaneously	74
4.5	Results of using TFBB, DoF, and FBCC simultaneously	76
5.1	Example of an arrow pointing to the artery of interest.	80
5.2	Example of an image with an abnormality wider than a single pixel.	83
5.3	Example of setting an error threshold for an abnormal image in image editor.	85
5.4	Evaluation: Presenting the tester with primary question	86
5.5	Evaluation: Presenting the tester with the image while the timer is running	87
5.6	Evaluation: Presenting the tester with the secondary question and marking the abnormality.	88

5.7	The information presented by each box in a box plot for each visualization category.	94
5.8	ANOVA: comparison of correct primary question answers	95
5.9	ANOVA Table: comparison of correct primary question answers	95
5.10	Multiple Comparisons: mean number of correct answers to primary question.	96
5.11	ANOVA: mean number of false positive primary question answers	97
5.12	ANOVA Table: mean number of false positive primary question answers	97
5.13	Multiple Comparisons: mean number of false positive answers to primary question $\langle TFBB, DoF, NCC \rangle$	98
5.14	Multiple Comparisons: mean number of false positive answers to primary question $\langle TFBB, DoF, RFBO+ \rangle$	99
5.15	ANOVA: false negative primary question answers	100
5.16	ANOVA Table: false negative primary question answers	100
5.17	Multiple Comparisons: mean number of false negative answers to primary question $\langle FF, NCC \rangle$	101
5.18	ANOVA: mean number of correct markings - secondary question	102
5.19	ANOVA Table: mean number of correct markings - secondary question	102
5.20	Multiple Comparisons: mean number of correct markings or correct answers to secondary question $\langle FF, NCC \rangle$	104
5.21	ANOVA: mean number of correct assessments with wrong marking - secondary question	105
5.22	ANOVA Table: mean number of correct assessments with wrong marking - secondary question	105
5.23	Multiple Comparisons: mean number of correct assessments with wrong markings or wrong answers to secondary question while correctly answering the primary question.	106
5.24	ANOVA: combined number of false negatives	108
5.25	ANOVA Table: combined number of false negatives	108
5.26	Multiple Comparisons: mean number of combined wrong markings.	109
5.27	ANOVA: mean normalized distance to the center of the defect.	110
5.28	ANOVA Table: mean normalized distance to the center of the defect.	110
5.29	Multiple Comparisons: mean normalized distance to the center of the defect.	111
5.30	ANOVA: mean time to answer the primary question	112
5.31	ANOVA Table: mean time to answer the primary question . .	112
5.32	Multiple Comparisons: mean time to answer the primary question	113
5.33	ANOVA: mean time to answer the primary question correctly	114
5.34	ANOVA Table: mean time to answer the primary question correctly	114
5.35	Multiple Comparisons: mean time to answer the primary question correctly	115
5.36	Coronary stenosis revealed by $\langle TFBB, NCC \rangle$	117
5.37	Pulmonary embolism revealed by $\langle TFBB, RFBO+ \rangle$	121
5.38	Coronary stenosis revealed by $\langle TFBB, YFCO \rangle$	122
5.39	Pulmonary abnormality revealed by $\langle TFBB, DoF, NCC \rangle$. . .	123
5.40	Pulmonary embolism revealed by $\langle TFBB, DoF, RFBO+ \rangle$. .	125
5.41	Large thrombus revealed by $\langle TFBB, DoF, RFBO+ \rangle$	126
5.42	Pulmonary embolism revealed by $\langle TFBB, DoF, YFCO \rangle$. . .	127
5.43	Aortic dissection revealed by $\langle TFBB, DoF, YFCO \rangle$	128

Chapter 1

Introduction

Correct and accurate clinical diagnosis has a significant role in many aspects of human life. It is the key to clinical planning and directly impacts patients' quality of life as well as length and effectiveness of the healing process. Clinical diagnosis, similar to any other rational decision making process, relies on information. This information needs to be accurate, correct, and practical. For each new tool that is proposed these properties must be rigorously and objectively measured to validate its performance. In modern medicine many such tools have been used and new ones are being developed and tested every day. One of these tools in clinical diagnosis, which has been proven to provide reliable and accurate information for more than half a century, is tomographic medical imaging.

In medical imaging different imaging modalities such as Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) are utilized to assess different types of abnormalities. The acquired data from these imaging devices are visualized in order to facilitate the examination of the internal anatomy of the patient in order to locate abnormalities.

Traditionally, data from CT or MRI are examined as a set of 2D slices. To make a correct assessment, a physician is required to mentally visualize the spatial arrangement of anatomy structures. This process requires long training

time and is prone to errors. Direct Volume Rendering (DVR) has been used as both a complementary and an alternative approach to visualizing medical volumetric data. However, due to many factors such as high computational cost, lack of efficient algorithms, etc. full capacity of DVR methods has not yet been made commercially available.

DVR methods are often inspired by the physical process in which a 2D image of a 3D object is projected on a surface. In nature this process takes advantage of almost infinite rays of light that are reflected by the 3D object onto the 2D surface of the camera sensor or the eye. However, to fully simulate such process is not computationally viable. Consequently, DVR methods take advantage of algorithms that simulate a simplified version of what happens in nature. For example, in traditional ray casting the number of light rays for each pixel in the final image is reduced to 1. This simplification greatly improves performance in terms of rendering speed, however, it results in an image of a 3D object that is in focus throughout its depth.

Human eye is an optical system. Our brain uses various natural optical phenomena to interpret the image captured by our eyes. For example, the depth of field that exists in the image formed on our retina is used with other phenomena such as motion parallax and stereo vision to help us understand depth and the ordering of our surrounding objects. Removal of any of these natural visual cues will weaken our ability to correctly judge the spatial relationship of objects. The correctness of this judgment becomes very important when a physician is examining results from imaging a patient. Any misjudgment can result in wrong diagnosis, wrong surgical planning, and can be detrimental to the patient. As a result, this thesis focuses on new algorithms to simulate depth of field in DVR in real-time as it applies to medical imaging.

Although natural visual cues are helping human vision a lot, there are more ways to improve the accuracy and correctness of image-based clinical decisions.

One of these ways is to create synthetic visual cues which are not necessarily seen in nature. It has been shown, as will be discussed in Chapter 2, that synthetic cues can be used to enhance or alter perception and direct viewers' attention to the objects of interest. The creation of these synthetic visual cues are based on the existing knowledge of human visual system. Many studies that assess various properties of our visual system are discussed in Chapter 2. Elements such as blur, opacity, size, color, etc. have great impact on our perception. The work presented in this thesis show some of the possible ways of using these elements to improve visual perception for clinical diagnosis.

Unique properties of the blurring method that we developed as part of the depth of field algorithm enabled us to take advantage of semantic depth of field and focus-context blurring in real-time direct volume rendering. Moreover, we were inspired by chroma-coding and its effects on visual perception. As a result, we have created multiple visual cues using combinations of material-based blurring, focus-based chroma-coding, depth of field and discussed them in Chapter 4.

Our hypothesis is that using DVR with our new methods can significantly improve medical diagnosis and reduce chances of false-negatives. In order to validate this hypothesis, we have designed and conducted a usability study. In this study we have shown that our visual cues can have a significant positive impact on clinical decision making. This study is discussed in detail in Chapter 5. Finally, our overview and conclusion is presented in Chapter 6.

In summary, the contributions of this thesis are:

- Interactive blurring and Depth-of-Field algorithm for DVR.
- Transfer Function Based Blurring (TFBB).
- Focus Based Color Coding (FBCC).

- Synthetic real-time visual cues for medical diagnosis using DVR.
- A usability study to validate our hypothesis.

Chapter 2

Background

2.1 Volume Rendering

Volume rendering refers to the process of visualizing volumetric data in 3D. Prior to the invention of volume rendering, it was only possible to look at the 2D slices of volumetric data. Looking at slices provides the viewer with some information; however, complete understanding and mental models of the spatial relation between different parts of volumetric data is a very challenging task. In many cases a specialized and long-term training is required before the viewer is allowed to make decisions based on such visualizations.

Many imaging modalities produce volumetric data that needs to be visualized. Computed Tomography (CT), Magnetic Resonance Imaging (MRI), echocardiography, finite element simulations, in addition to geographical and meteorological measurements are just a few of many ways to acquire this type of data [9].

Early works on volumetric data almost always involved extracting an iso-surface and then rendering using the polygons extracted. This results in losing a huge portion of information contained in the volumetric data that can contribute to the final image. Direct Volume Rendering (DVR) was created to tackle this problem. DVR provides us with a way to fuse the whole portion of the volumetric data to render potentially more informative images. This

can result in a better understanding of the data at the expense of more computation and more complex methods [16]. Direct Volume Rendering refers to methods that render volumetric data without having to generate any geometrical primitives as an intermediate step. There are three categories of DVR methods: *image-order*, *object-order*, and *domain-based*.

In image-order methods a backward mapping happens where a group of rays are cast one by one (in a sequential process) or simultaneously (in a parallelized process) through the volume in order to determine the color of each pixel in the final image. Ray casting is one of the most important techniques in this category. In object-order methods a forward mapping happens where the volumetric data is mapped on the 2D image plane of the virtual camera. The iteration in these algorithms happens on the data voxels in 3D space as opposed to pixels of the final image in 2D space. Splatting [16, 61] is one of the object-order techniques. In domain-based volume rendering volumetric data is transformed into other domains such as frequency domain or compression domain and then the rendering is performed from that domain directly. There are also hybrid methods such as shear-warp which uses a combination of different categories of DVR methods [16].

2.2 Image-Order Methods

Image-order methods are one of the subcategories of DVR techniques. In these methods color and opacity of the pixels in the final image are calculated one by one based on volumetric data. One of the most important techniques for image-order methods is ray casting.

2.2.1 Ray Casting

Ray Casting is one of the common operations among image-order methods. In this method, for each pixel in the final image a ray is cast through the volume

while discrete samples are taken from the volume along the ray. Data points in a volumetric data represented in a regular grid are located on grid points. In ray casting sampling might happen at arbitrary locations between grid points. Various interpolation methods such as trilinear or tricubic methods are used in order to approximate the value at the exact sampling location. There are numerous methods described in the literature about the ways to obtain and use these values called volume rendering modes.

It is possible to simply add the values obtained for each ray (x-ray rendering) or to only use the sample with the maximum value (Maximum Intensity Projection). Another possibility is to process each sample considering its material and transparency along the ray to calculate the color of each pixel. This method is called full volume rendering [16] and is described mathematically by:

$$I_\lambda(x, r) = \int_0^L C_\lambda(s)\mu(s)e^{-\int_0^s \mu(t)dt} ds, \quad (2.1)$$

where c and α are calculated from the following recursive equations:

$$c = C(i\Delta s)\alpha(i\Delta s)(1 - \alpha) + c, \quad (2.2)$$

$$\alpha = \alpha(i\Delta s)(1 - \alpha) + \alpha. \quad (2.3)$$

In order to calculate the value of each pixel using full volume rendering, one must consider light attenuation, reflection $R_\lambda(s)$, transparency $T_\lambda(s)$, and emission $E_\lambda(s)$. It is possible to incorporate these values while summing the acquired samples for each ray using the ray casting integral (see Equ. 2.1). In this equation, $I_\lambda(x, r)$ is the amount of light with wavelength λ received at point x on the image plane from direction vectorial r . The parameter μ represents the mass density of the particles in the volume and $exp(-\int_0^s \mu(t)dt)$ accounts for light attenuated by going through particles with density μ . The parameter L is the length of the ray and s is the location of the particle in space.

$C_\lambda(s) = R_\lambda(s) + T_\lambda(s) + E_\lambda(s)$ is the contribution of different light sources to a ray. Often the integral of Equ. 2.1 is approximated by the Riemann sum in a low-albedo model (see recursive Equ. 2.2 and Equ. 2.3). In low-albedo model the secondary scattering and reflection is neglected. In this way only the direct ray of light and its attenuation before reaching the voxel is considered. We should note that this is a simplified model of light transfer in a propagating media. The simplifications and assumptions as well as detailed analysis of volume rendering mathematics are presented in Hadwiger and Hansen [14, 16].

It is possible to sample volumetric data in a front-to-back or a back-to-front traversal schemes. The front-to-back method is more popular than the back-to-front method (painter’s method) since it enables the algorithm to take advantage of early ray termination (ERT). Early ray termination is one of the most common acceleration methods for ray casting. It stops a ray from sampling the volume when the accumulative opacity of the corresponding pixel reaches a predefined value, or when a step close to a predefined threshold occurs in the value of the samples being read from the data.

2.2.2 Speed Optimization for Ray Casting

Many applications that use volume rendering require interactive frame rates. For example, smooth navigation inside arteries in virtual endoscopy software can help physicians to make correct clinical decisions compared to waiting many seconds or even minutes for each frame to render. With interactive frame rates viewers can take advantage of phenomena such as motion parallax to obtain more information from the rendered images.

In the past 25 years many methods have been proposed to enhance the performance of ray casting. Some of these methods reduce the computation required by renderers, some take advantage of parallelization, and some combine both. In this section, we will present different ideas and techniques found

in the literature to tackle this problem. Researchers have often tried to reduce the computation required by ray casting by using techniques such as empty space skipping (ESS) or ERT.

Levoy proposed an ESS technique [28] using hierarchical space abstraction. In his method the lowest level in an abstraction hierarchy, level m , is the actual volumetric data, and at the top of abstraction hierarchy, level 0, is a cube of size $(1, 1, 1)$. For any given n where $0 < n \leq m$, level $n - 1$ is a volume in which each node represents 8 neighboring voxels (forming a super-cube) from level n . The value of a node at level $n - 1$ is 1 if any of those 8 voxels at level n has any value other than 0, otherwise the value of the corresponding node at level $(n - 1)$ is 0.

By doing this the number of nodes at each abstraction level $n - 1$ equals to approximately $\frac{1}{8}$ of the number of nodes at its lower level n . Since the first 2 top levels and 2 bottom levels are only going to add extra computational overhead and not much performance gain, one can discard them and start casting rays from level 3 with a step size according to granularity of that level. Each time a location with a value other than 0 is sampled, the algorithm switches to a more finely divided abstraction level and also changes the step size in accordance to the Nyquist rate. Consequently, the rendering algorithm can skip clusters of empty nodes while not missing any data points. Skipping traversal of the empty space using this method will result in less interactions with the data, and consequently improved performance.

This method of empty space leaping depends on the amount of empty space as well as its distribution in space. Levoy [28] also proposed a technique for ERT. He called this method adaptive termination in which the process of casting a ray stops when the opacity of a voxel along the ray differs from the previous ones by more than a user-defined threshold.

Yagel *et al.* [66] used coherence between nearly similar views to skip the

empty space inside volumetric data. A *Coordinates-Buffer* was used on which the coordinates of the surface of the volume is projected. A change in viewing parameters will then result in transformation of the buffer and the ability to discard hidden surfaces. The values in this buffer are used as estimators of approximate location inside the volume to start sampling. Wan *et al.* [59] proposed a technique for skipping empty space by classifying the voxels into three different categories: *inside*, *outside*, and *boundary* voxels. Then a technique resembling the coordinates-buffer of Yagel [66] projects the location of boundary voxels onto a buffer. These stored locations are then used as approximations of starting positions for casting rays.

Meißner *et al.* [36] proposed a change in the way the rays are cast. He proposed a memory-efficient coding that will help skipping the empty space. Normally after initializing the starting location and direction of a ray, the ray casting process starts moving away from the origin of the ray by taking samples along each ray on equidistance locations called steps. Meißner [36], however, instead of advancing the sampling location by a fixed step size, advances the sampler to the next sample point of interest in one step regardless of how far that point is. The encoding process divides the volume of size $(p \times q \times r)$ into non-overlapping sub-cubes of size $(i \times j \times k)$ where $\{(0 < i \leq p) \wedge (0 < j \leq q) \wedge (0 < k \leq r)\}$. Each of these sub-cubes is then mapped to a bit in a space-occupancy map. At the time of ray traversal the bits in this space-occupancy map are checked and if they have a value of 0 the distance to the next sampling point of interest is calculated. Then the ray is advanced to that point. If the value in the space-occupancy map is 1, the algorithm will proceed with regular sampling until it reaches the next sub-cube.

Kruger *et al.* [23] used an octree data structure to divide space. In an octree each node has exactly 8 children. Each level of the octree contains a coarse abstraction of the original data with a different level of granularity.

For each node of the abstraction, maximum and minimum intensity values are stored. The traversing happens on the coarse data and if the transfer function indicates that a value within the cell (between min and max) is marked as visible, the traversal of ray through that node will happen on the volumetric data with the original granularity. Kruger *et al.* [23] also proposed a different method for ERT. Instead of only considering opacity increments bigger than a user-defined ε , the accumulative opacity is also considered. That is, at any point during traversal of the ray if the accumulative opacity of a pixel reaches a certain threshold τ or an opacity increment bigger than user-defined ε is observed, the ray traversal is terminated.

Lakare *et al.* [25] used coherence between neighbouring rays in order to enable a majority of the rays to skip the empty space. They divide rays into *detector rays* and *space-leap rays*. Detector rays are distributed uniformly among space leap rays. In the first pass, detector rays are cast and the number of steps required for each ray to intersect with the first non-empty voxel is stored in the *Leap Buffer*. Color and accumulated opacity calculated by a detector ray is also stored into the image buffer like normal ray casting. Leap buffer and image buffer are of the same size. In the next step the empty spaces around each detector pixel in the leap buffer is populated with the values of the detector pixels. In the final pass the space-leap rays are cast. Moreover, each of these space-leap rays is advanced forward with one giant step in order to skip the empty space in front of them. The size of this big step is calculated by multiplying the value stored for each ray in leap buffer and the step size used by detector rays. Figure 2.1 shows different states of leap buffer during the three stages of this rendering algorithm.

Li *et al.* [30, 29] proposed a method using growing boxes to skip empty spaces. A cost function is calculated for each box and the box tends to grow as long as the cost function remains low. They used three types of boxes in

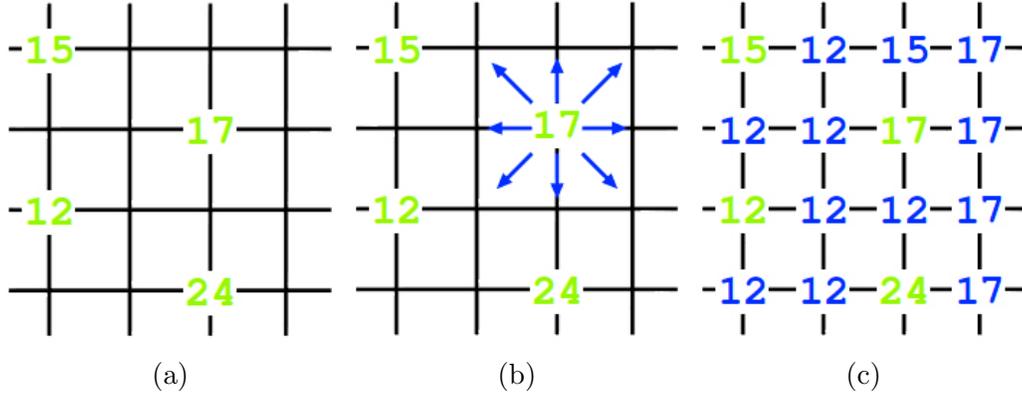


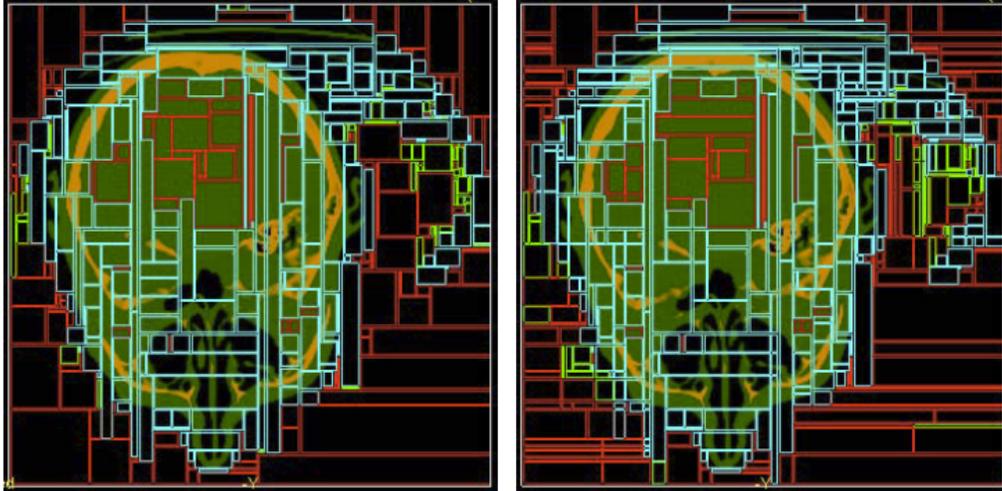
Figure 2.1: Three states of leap-buffer for empty space skipping using coherence between neighboring rays [25]. Fig. 2.1a shows how the number of steps taken by detector rays to reach the first data points is stored in leap buffer. Fig. 2.1b shows how the neighbors of a detector ray are going to be affected by a detected value. Fig. 2.1c shows the populated leap buffer [25]. © 2004 IEEE.

case of illuminated volume rendering: *uniform*, *gradient*, and *other*. *Uniform* boxes are used for the uniform areas. *Gradient* boxes are used for the places where a surface exits (non-zero gradients). The remaining spaces are grouped into *other*. They have used an orthogonal binary space partitioning (BSP) tree as the underlying data structure for this abstraction. The main benefit of using an object-aligned BSP tree¹ is that the order in which the boxes are traversed can be forced to be either front-to-back or back-to-front. By merging and splitting the boxes in BSP tree, it is possible to reduce the number of boxes as well as making the boxes better fit the data. Figure 2.2 shows one slice of a brain image partitioned by growing boxes.

2.2.3 Hardware Acceleration for Ray Casting

Hardware accelerated ray casting has been widely researched in the past 15 years. Many researchers focused their research on developing dedicated hardware for volume rendering. Several architectures were proposed which resulted in designing VolumePro hardware [43] based on EM-Cube [39] architecture.

¹It seems that by object-aligned they mean axis-aligned.



(a) After conversion to BSP tree (b) Before conversion to BSP tree

Figure 2.2: An example of growing boxes [30]. © 2003 IEEE.

However, as a result of recent advances in the design and development of graphics hardware and GPUs, a dedicated volume rendering hardware is no longer required in order to achieve interactive frame rates. Another path for research in hardware accelerated volume rendering is to develop parallelized algorithms and methods as well as utilizing tools such as GPUs using CUDA, GLSL, CG, etc. to achieve interactive frame rates and high image quality.

Fortunately most of volume rendering algorithms are highly parallelizable. For example, in ray casting for each pixel in the final image a ray is cast through the volume. It is possible to assume these rays and their computations are completely independent of each other; as a result, a thread could be assigned to each ray on a node in a multi-node parallel system. In this section we will briefly look at some of the literature that proposed methods utilizing programmable graphics hardware (GPU).

Kruger *et al.* [23] used 3D textures and fragment shaders to exploit the capabilities of graphics hardware to accelerate rendering speed. In their method, an empty cuboid is created containing the volume using geometrical primitives. Front and back faces of the cuboid are rendered to two different texture

buffers. When the view is changed, the cuboid will be transformed and translated accordingly and rendered to the texture buffers. By simply subtracting the points with same coordinates from the two texture buffers, the direction of the ray is calculated. In an enhanced version of this method many small cubes that contain sub-volumes are rendered to the textures. In this way more empty space is skipped.

Gobbetti *et al.* [12] divide the data into coarse bricks and organize them into an octree structure stored on CPU. For each frame a cut of the octree that contains relevant nodes to the current view is made and transferred to GPU memory. The generated octree nodes are augmented by linking their neighbours making front-to-back rendering possible. The structure of the octree is encoded and stored in a texture on the GPU which is then used by fragment shader. The encoding procedure is fast since the octree resolution is coarse and there are only a few thousand nodes that should be encoded. The coarse octree structure enables the use of adaptive sampling for this method and consequently enhances the skipping of the empty space.

2.3 Object-Order Methods

As mentioned before, in object-order methods the volume rendering iteration happens on basis elements or basis functions inside data instead of pixels in the final image. In this section, we will talk about Volume Splatting which is one of the main techniques of object-order volume rendering.

2.3.1 Volume Splatting

Kernel splatting or simply splatting is the process of projecting basis functions inside data onto an image. The reason for calling it kernel splatting is that in most cases an interpolating kernel function is initially put on each voxel in data. Then the footprint of data points on the final image is calculated using

that kernel function.

Composite-only, axis-aligned sheet based, and image-aligned sheet based splatting are the three types of splatting [16]. Composite only method for splatting was proposed by Westover [61]. In this approach, each voxel is assigned a color and an opacity value according to the transfer function. These voxels are then traversed either in front-to-back or back-to-front fashion. During this process the footprint of each voxel is calculated on the final image as a contributing splat. Although splatting was faster than other volume rendering algorithms at the time, lack of quality in the rendered images created problems such as color bleeding and sparkling artifacts [16]. As a result, sheet-based splatting was introduced. We will talk about axis-aligned sheet-based and image-aligned sheet-based splatting in Section 2.6 when we talk about enhancing the quality of rendering.

2.4 Other Volume Rendering Methods

Most of the research on direct volume rendering has been concentrated on image-order and object-order methods. However, there are hybrid methods and domain-based methods that were proposed for DVR as well. Hybrid methods such as shear-warp algorithm [24] combine the benefits of both object-order and image-order methods to achieve high performance.

In shear-warp algorithm, one of the 3 possible axis aligned planes of a volume is chosen that makes the smallest angle with the current viewing plane. The volume is then sheared and projected onto this plane as an intermediate step. The image on the axis-aligned plane is then warped to create the final image. Before projection, the volume is encoded using run-length encoding (RLE). This reduces the amount of required space and efficiently skips the empty space [24, 16]. This method in its initial form had visible aliasing artifacts; however, this issue was addressed in later papers.

Another set of volume renderers are domain-based volume renderers. In these methods the volumetric data is transformed into other domains such as frequency, wavelet, or compression domains and then rendered. Based on Fourier projection-slice theorem the result of inverse Fourier transform of a 2D slice taken from the Fourier transform of the 3D volumetric data is equivalent to projecting the volumetric data onto a 2D plane. However, the frequency domain cannot deal with occlusion. Some researchers have proposed methods to address these issues related to frequency domain rendering [57, 16]. Compression, wavelet and multi-resolution transforms were also used in volume rendering to reduce the amount of memory requirement and also speed up rendering by reducing computation. For example, it is possible to transform uniform regions of a volume to coarse resolution regions using wavelet [16]. In recent years the focus of research on compression and wavelet domains has shifted from normal volume rendering to very large datasets.

2.5 Visual Perception in Volume Rendering

There has been a lot of work on improving visual perception in direct volume rendering using both natural and synthetic cues. Both image-order and object-order methods [1] have been proposed to enhance perception for various rendering techniques. However, DVR has received less attention in spite of its importance. Many visual enhancement methods from other disciplines do not trivially extend to DVR due to their differences in nature and applications. In this section, we will review various techniques to enhance visual perception as well as methods created specifically for DVR. Our emphasis will be on creating synthetic visual aids based on either natural cues or artificially created cues and their combinations to direct viewer’s attention, highlight regions of interest, and to enhance visual perception in DVR.

2.5.1 Directing Viewers' Attention

Directing viewers' attention to the objects of interest as well as reducing clutter and noise in visualization has been researched in many different contexts. In this section, we will briefly review works that their ideas can potentially enhance visual perception in DVR.

Cockburn [5] created an informative survey of different approaches to enhance visual perception and user experience. The surveyed methods mostly dealt with either visualization of complex information or complicated interfaces, some of which are discussed later. Kosara *et al.* [18, 19] proposed and tested the idea of semantic depth of field for directing viewer's attention to the object of interest in a crowded environment. This idea was previously explored in a different context by Wixson [65], who proposed non-linear focus to reduce the clutter in the image, however, there was no mechanism to extend the idea to DVR. Many techniques for emphasizing region of interest in visualization were proposed such as a system called the macroscope by Lieberman [31]. An area of interest on a map is expanded and kept in focus while providing context by blurring the less detailed background.

Viola *et al.* [58] proposed a system for importance-driven focus of attention. In this system, the volumetric data is pre-processed and the user could choose from a set of pre-defined features. The best view points are chosen automatically using view point mutual information system. Moreover, many researchers have worked on the effect of blur on human perception. Ciuffreda *et al.* [4] have empirically evaluated the effect of blur and their study suggests that while blur can help understanding of relative depth of objects it certainly directs viewer's attention to the object of interest. Distinguishing a cup on a crowded table is one of the examples they use from everyday life.

Chroma-Coding for depth was proposed by Steenblik [54] and was extended

into chromadepth, and later pseudo chromadepth. Ropinski *et al.* [45] have shown that DoF blurring is useful when accompanied by pseudo chromadepth in angiography. Additionally, other methods such as volumetric halos were used by many researchers [3, 55] to emphasize objects of interest and direct viewer’s attention. We will discuss some of these methods later in more detail.

2.5.2 Depth Perception in Volume Rendering

Compared to other aspects of DVR, less attention has been paid to depth perception. However in recent years many new studies and methods are being published which are focusing on depth perception in DVR. Depth cues can potentially improve the information gain of a rendered image. This can be crucial in medical understanding of the rendered images such as 3D ultrasound where it is especially hard to perceive depth. Furthermore, performance plays an important role here and existing methods should be modified to fit the performance requirements of real-time medical systems.

Commercial software that are currently available do not take advantage of all possible depth enhancing cues. For example, pseudo chromadepth was used in the Philips machines at University of Alberta hospital as the only depth enhancement cue in addition to perspective. Since this project is focused on enhancing visual perception for DVR, we are going to continue looking at various types of visual cues and methods that improve the spatial understanding of a DVR-rendered image.

Similar to photography, in volume rendering we are projecting a 3D object onto a 2D screen. Normally our brain observes objects from two slightly different angles (stereo vision) and that is one of the ways we perceive depth in real life. Nevertheless, there are other visual cues that help our brain to perceive depth even in the absence of stereo such as in 2D images or on a screen. Some of the visual cues that provide a sense of depth for 2D images are occlusion,

depth of field, perspective, ambient light shadows, perspective, motion parallax, etc. Although there has been a lot of research on depth perception in computer generated images, less attention has been paid to this problem in DVR. In this section, we will look at some of the work on improving depth perception using such depth cues.

Wanger *et al.* [60] designed a few number of experiments to evaluate the effect of various depth cues on depth perception for computer generated images. They have studied effects of perspective projection, motion, shadows, and texture on three types of tasks: positioning, orienting, and scaling. Their results show that the effectiveness of each of the tested depth cues can depend on the task as well as presence of other depth cues simultaneously. In positioning tasks shadow and perspective projection improved the perception of depth significantly. In the orienting task shadows and motion significantly improved results while perspective projection negatively affected viewers' depth perception. In the scaling task all four tested cues improved the results significantly.

2.5.2.1 Depth-of-Field

Although there has been a huge amount of work for simulating DoF for rendering geometry, there has not been many methods proposed to tackle this in DVR. Many methods has been proposed for DoF simulation in ray-tracing; however, due to the differences between ray-tracing and ray casting and their applications, it is not trivial to extend those methods to DVR as well. Both image-space and object-space methods as described by Barsky [1] can potentially be used in DVR.

Mather *et al.* [35] experimented with region and border blur as depth cues. They discovered that region blur is more effective for depth perception than border blur. Also, they showed that although blur is an effective depth cue, the limitations of human perception prevents us from distinguishing various

levels of blur to perform depth ordering.

Held *et al.* [15] showed the fact that blur was previously shown as a weak depth cue is not correct. Blur can be used to impact the perception of size and depth ordering of objects in a scene. They have shown that a combination of blur and other depth cues can strongly increase the viewer's ability to perceive the absolute or relative distance between objects. An algorithm was proposed by Held *et al.* [15] to create the illusion of different scale models of the same polygonal scene. In order to do so the tilt-shift effect was used. Here the term tilt-shift generally refers to the exaggerated blur effect created by tilting the lens. In this photography technique, the image plane and the lens plane are not parallel, which causes the focus plane to tilt as well. This result is a different depth of field effect compared to having a large aperture opening.

In Ciuffreda [4], an empirical model of human blur perception is presented. This study evaluates blur distinction, discrimination and depth ordering based on blur. The proposed model suggests that it is possible to clearly distinguish an object of interest surrounded by a lot of other objects if DoF is employed. Ciuffreda [4] claims that viewing objects on a monitor screen will also follow the same model. Boucheny *et al.* [2] performed various experiments to evaluate the quality of depth perception for DVR. They identify the problem of depth perception in DVR by arguing that human perception is trained mainly with opaque surfaces rather than with semitransparent ones. The latter, which is mostly the case in DVR, can mislead human perception in terms of depth in absence of other assisting depth cues. They have performed user studies with motion parallax, exaggerated perspective, and light propagation models and found shortcomings in each one of them.

In another study Ropinski *et al.* [45] have evaluated the effect of various depth cues for the interpretation of angiographic images. Based on their studies, they conclude that angiography benefits from DoF effect combined

with pseudo chromadepth. Ropinski *et al.* [45] performed a usability study to evaluate different methods for depth perception. However, they did not provide any technical details about what methods were used to create these effects. Their study evaluates standard rendering, stereoscopic rendering, chromadepth, pseudo chromadepth, overlaid edges, blended edges, perspective edges, edge shading, depth of field, and pseudo chromadepth combined with depth of field. What they describe as the difference between chromadepth and pseudo chromadepth is that in chromadepth each hue is mapped directly to a depth value of a voxel based on the fact that different color wavelengths of light get refracted in different angles by the lens of the eye; however, in pseudo chromadepth the number of hues are reduced to red and blue. Based on their usability study they also suggest that angiography can benefit from chroma-coding and pseudo chromadepth. However, it is not clear if the users have any experience with angiography or if they are medical professionals. They also note that some users had bias for some of the techniques because of their previous experience, however, this was not considered in the results.

In a study by Grosset *et al.* [13] the effect of DoF on depth perception in DVR has been evaluated. They have shown that DoF effect when accompanied by other depth cues can improve the perception of ordinal depth in many cases but not all. Grosset [13] states that although DoF can be beneficial to depth perception, in some cases blurring of the objects which are farther from the focus plane might confuse the viewer's ordinal perception. Barsky [1] presents an informative survey of various techniques to produce DoF effect in different rendering methods. Many techniques have been proposed in two main categories - object space and image space methods. These methods are mainly used for techniques other than direct volume rendering. Light field methods are also introduced as part of image order methods. These methods are mostly used for changing the focus and depth of field of a two dimensional

image after it has been created. A list of conditions for correct generation of depth of field is also proposed that requires using a Point Spread Function (PSF), controlling per-pixel blur level, lacking intensity leakage, lacking depth discontinuity artifacts, simulating of partial occlusion, and high performance. Although some of the ideas presented in this chapter such as image-space approaches can be extended to DVR as well, there are no specialized or easily extendable to DVR methods presented to create DoF.

One of the first methods to simulate DoF was proposed by Cook [7] for ray-tracing and called Distributed Ray-Tracing. To determine the color of a point in this method many rays are traced for each point on the image plane and through various points on the lens. The returned value of these rays are then blended together to simulate an integral of rays with a common starting point over the surface of the lens. Although this methods can produce good results for ray-tracing, it is computationally very expensive.

Yu *et al.* [67] devised a method to generate DoF in ray-tracing using synthetic light field. Kraus *et al.* [20] proposed an image-space approach using sub-images and depth map to create depth of field through post processing of the rendered image. Although this is an image-space technique, it is not trivial to extended to ray casting with transparencies. Another post processing technique was proposed by Lin *et al.* [32]. The proposed method is mostly concerned with the correct simulation of camera model to synthesize a photo-realistic DoF effect. It relies on image segmentation to have different blur levels.

Lei *et al.* [27] extended the idea of Distributed Ray-Tracing. Normally in distributed ray tracing many rays are cast from a common starting point through different points of the lens to calculate the color of the point on the image plane. Calculating such an integral over the surface of the lens is very expensive. However, in their proposed method [27] the value obtained from

each single ray in a pinhole camera model is reused. Each of these values correspond to a pixel on the image plane. Additionally, to compensate for partial occlusion a few additional see-through rays are traced. In this way, the DoF effect is approximated for ray-tracing with much lower cost. Demeres [8] surveys different techniques for creating depth of field effect while rendering scenes using ray-tracing. These techniques are divided into techniques for high quality results and real-time results. None of these categories involve techniques for real-time DVR, however, they present some ideas that can be expanded to be incorporated by DVR techniques. In one of the closer techniques to DVR the scene is rendered from multiple points of view in the image and then blended together into accumulation-buffer. The more images we have to blend, the closer the effect is to the real depth of field.

Surprisingly there has been very little work on creating DoF effect for DVR. Krivanek [22] presents an algorithm to produce DoF effect for surface splatting. The idea of under-sampling in blurred areas from this paper, which was also emphasized by Barsky [1] has inspired us for the work presented in Chapter 3. Schott *et al.* [46] proposed a method to generate DoF for volumetric data in slice-based direct volume rendering. The idea of incremental blur originally proposed by Rokita [44] was used with slice-based approach to DVR to create this effect. In this approach, the volume is divided into thin slices parallel to the image plane. It is then divided into front, back, and in-focus slides. Each are processed separately into three different buffers and then blended together to create the final image. Figure 2.4 illustrates how the circles of confusion are created using a single lens model. Figure 2.3 shows the results generated by the method proposed by Schott [46].

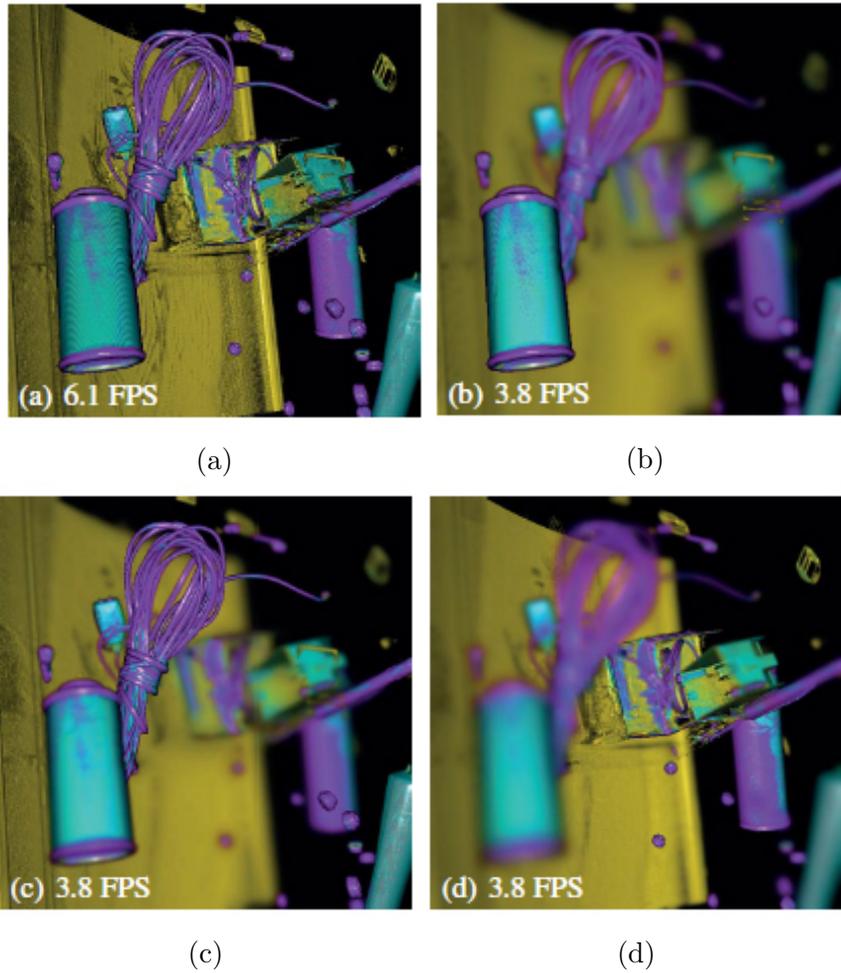


Figure 2.3: Simulating depth-of-field-effect for direct volume rendering. Fig. 2.3a is without this effect and the rest are with depth of field effect while focusing the camera on different objects. It can be seen that understanding of the spatial relations between objects is greatly improved [46]. © 2011 IEEE.

2.5.3 Volume Illumination

Illuminated volumes with proper shadows and shading can almost always convey more information to the viewer compared to plain rendering. Lightness and shadows can contribute to depth perception as well as object recognition. In this section, we look at few papers on illumination and shadows for both image-order and object-order volume rendering.

In surface rendering there is often a surface consisting of geometrical primitives. Each element on a surface is on a plane and has a normal vector.

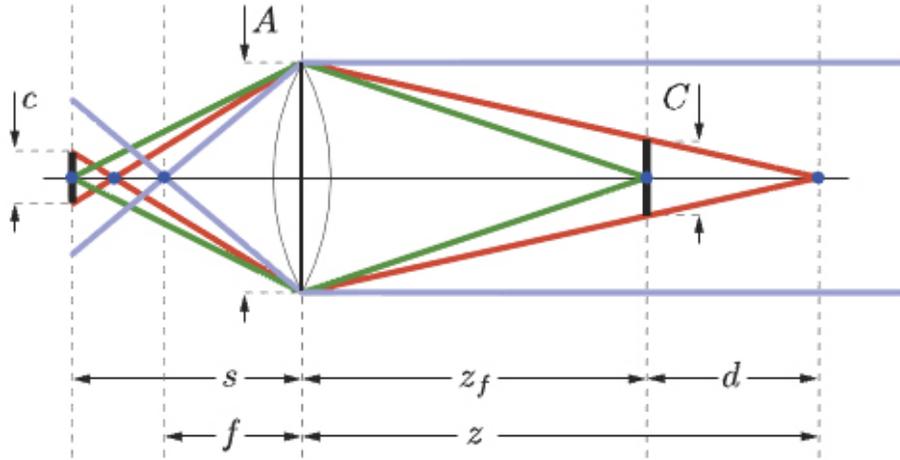


Figure 2.4: In-focus objects and out of focus objects have different effect on the final image plane. Rays from the former hit only one point. However, rays from the latter make a circle of confusion on the final image [46]. © 2011 IEEE.

These normal vectors are used to calculate the illumination of a surface. In the simplest case a unit vector representing direction of light, a normal vector representing surface normal, light intensity, light color, and surface color are used to calculate perceived color of a surface. However, in direct volume rendering there are no geometrical primitives representing surfaces and consequently there are no normal vectors to be used for shading calculations. As a result, gradient vector for each voxel is used as a replacement for surface normal vectors. Stronger gradients mean stronger surfaces that for example can induce greater reflections. Gradient of a voxel can be calculated in many different ways (for details see Hansen [16]). One of the widely used techniques that produces acceptable results in most cases is central differencing, which is shown in Equ. 2.4. This method is very popular for its speed. There are other more accurate but slower gradient calculation methods such as Sobel [53], Zucker-Hummel [70], etc. These methods look at more samples in the neighborhood of the current voxel to calculate a more accurate gradient. By using gradients, existing methods for surface illumination can be used in DVR.

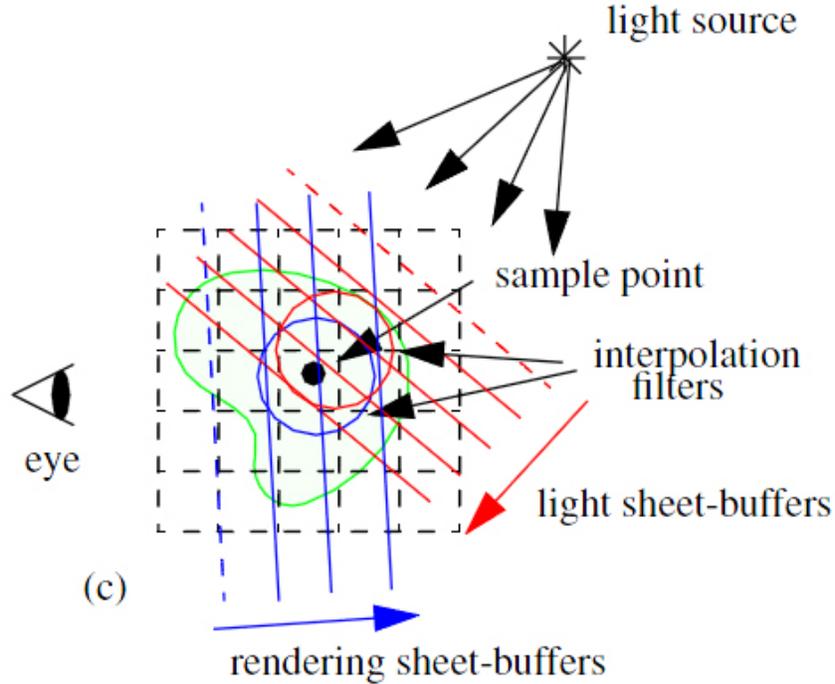


Figure 2.5: Single-pass approach to create shadows for volume splatting [68].
 © 2002 IEEE.

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} V_{(x-1,y,z)} \\ V_{(x,y-1,z)} \\ V_{(x,y,z-1)} \end{bmatrix} - \begin{bmatrix} V_{(x+1,y,z)} \\ V_{(x,y+1,z)} \\ V_{(x,y,z+1)} \end{bmatrix} \quad (2.4)$$

Zhang *et al.* [68] stated that calculating shadows in splatting is the same as rendering itself. One of the first algorithms to project shadows for splatting was introduced by Nulkar *et al.* [38] and required two render passes: one for volume and one for shadows. First splatting pass was from the light source point of view to determine visibility of the voxels and store this information in a shadow buffer. Then splatting from eye point of view should take place and visibility information in the shadow buffer is used to determine illumination. The method proposed by Zhang *et al.* [68] reduced the required render passes to one. For each splat during the rendering pass, contribution of a voxel to the

sheet buffer is calculated, and at the same time its contribution to the shadow buffer is also calculated by transforming the voxel from eye point of view to light source point of view. Figure 2.5 illustrates the single-pass approach for volume splatting.

2.5.4 Volumetric Halos

In computer graphics two types of visual cues are used. First type of cues are the ones directly derived from a real physical phenomena such as depth of field or shadows. The second type of visual cues for perception are either used in illustration or devised specifically to help visual perception in rendering. Volumetric halos are somewhere in between these two categories of visual cues. These halos exist in real life in form of faint shadows created by ambient light. However, a more exaggerated version of these halos are used in illustrations. In order to generate such halos in DVR some methods have been proposed such as the one by Tao *et al.* [56]. In their method a halo volume is generated and rendered with the normal volume. This halo volume is based on the halo seeds placed around the edges and contours in the opacity volume. The subtraction of the smoothed opacity volume from the normal opacity volume provides the location for the halo seeds for halo generation method.

Bruckner *et al.* [3] proposed a method to generate volumetric halos in order to enhance depth perception. Halos are used by artists and illustrators to emphasize objects and occlusion. View aligned slices in front-to-back order are used for generating halos. Halo contribution of each slice is calculated and then composited with the rest of slices at the end. Initially in the three step process of generating halos, the halo region is selected and then intensity values for the halo are generated. Finally, these intensity values are mapped to opacity and color values and combined with volumetric data to be used for rendering. Figure 2.6 compares the result of volume rendering with and

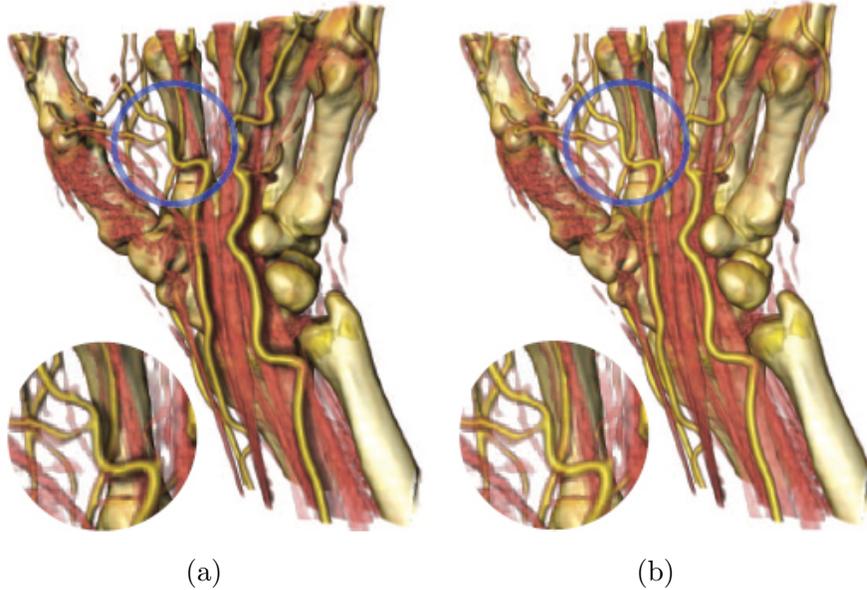


Figure 2.6: A comparison between a volume rendered with halos in Fig. 2.6a and one without halos in Fig. 2.6b. We can see how much better we can separate arteries and bones from each other in the haloed one [3]. © 2007 IEEE.

without halos using the method proposed by Bruckner [3].

Schott *et al.* [47] proposed a method to simulate ambient occlusion for volume rendering. Instead of using a sphere to calculate ambient occlusion, he used “a backward-peaked cone phase function of user-specified aperture angle”.

Moreover, color, contrast, and luminosity can play an important role in depth perception. Some researchers such as Zheng *et al.* [69] have proposed methods to enhance depth ordering by using differences between luminance of different overlapping translucent layers. They have used optimization of an energy function that they have also proposed as a quantitative way of evaluating and enhancing perception of depth. However, this method is still very slow, requires some segmentation of the image (this is an image-based approach), and can cause incoherency in animations.

2.6 Quality Enhancement

There have been many methods proposed in literature to enhance the results of volume rendering. There are problems like moiré and other visually detectable artifacts such as color bleeding, jagged edges, lack of sharpness, etc. We will briefly discuss tackling the lack of sharpness in Section 2.7. Here we present few methods to enhance volume rendering results.

Lee *et al.* [26] used virtual sampling in order to enhance the quality of volume ray casting without adding noticeable computational overhead and even speeding up the process compared to true sampling. For each ray segment during ray casting, the beginning and ending points of the segment are used to interpolate a curve between them using Catmull-Rom spline curve. The reason for choosing this method of interpolation is that this curve is C^1 and does not require the whole sample set (unlike C^2 splines) to be generated. This will enable the algorithm to also benefit from early ray termination. After the spline is generated for the ray segment, more sampling will take place on the interpolated curve instead of the real data. This type of sampling is referred to as virtual sampling.

Min-Max blocks are also used [26] in order to incorporate empty space skipping in this algorithm. These blocks contain a set of adjacent voxels. Maximum and minimum intensity values of the inner voxels are also stored for each block. When a ray hits one of these blocks, if the range between min and max values is visible according to the transfer function, the algorithm will continue sampling inside the block but with the sampling rate required for the original data. Otherwise, the ray skips the current block and processes the next block. In virtual sampling if the values generated by interpolation curve fall outside the $[min, max]$ range, we do not consider them for visibility test with transfer function. This form of abstraction is more elaborated in Sec-

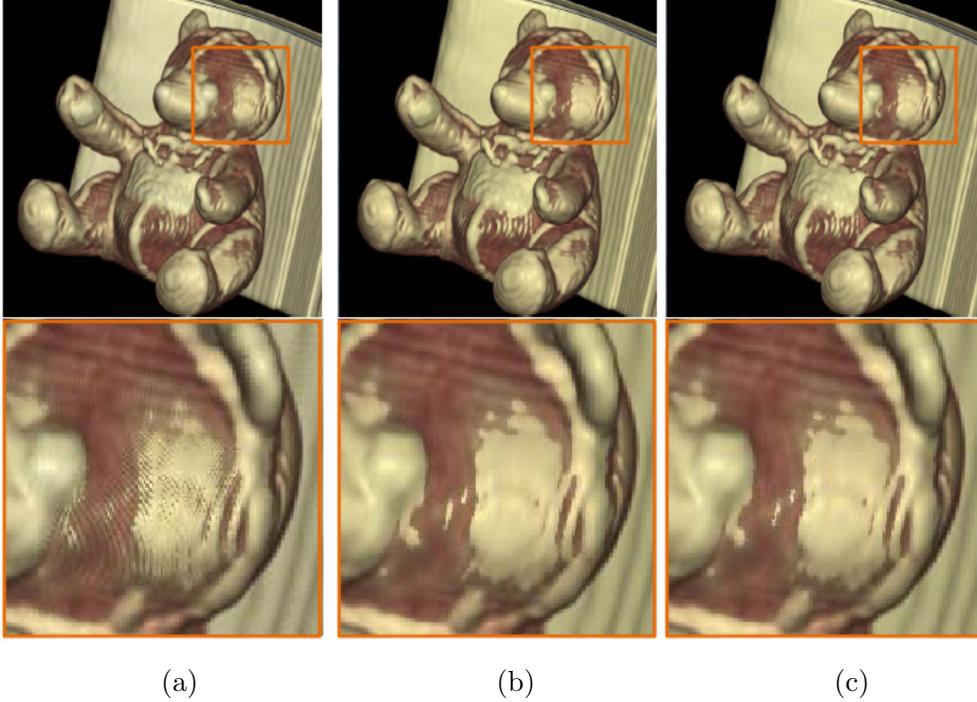


Figure 2.7: Bear with dimensions $(128 \times 128 \times 62)$. Fig. 2.7a Cubic \times 1. Fig 2.7b Cubic \times 4. Fig. 2.7c virtual sampling method. We can see that the results of 2.7b and 2.7c are very similar except the latter is missing some details [26]. © 2010 IEEE.

tion 2.2.2. Figure 2.7 compares visual quality of the results from interpolation using Cubic \times 1, Cubic \times 4, and virtual sampling.

Axis-aligned sheet-buffer splatting introduced by Westover [63, 62] is a method to reduce the background color bleeding of composite-only splatting. In this method, the face of the volume most parallel to the image plane is used as a parallel reference for a set of sheets. Splats are added to these sheets and then the sheets are composited in back-to-front order. Although this technique enhances the final image, a visually disturbing popping artifact as described by Mueller [37] can be seen in the animated rendered images. Mueller *et al.* [37] enhanced the sheet-buffer splatting to eliminate the popping artifact. This problem happens mainly when the face of the volume most parallel to the viewing plane changes due to transformation of the volume. This problem is

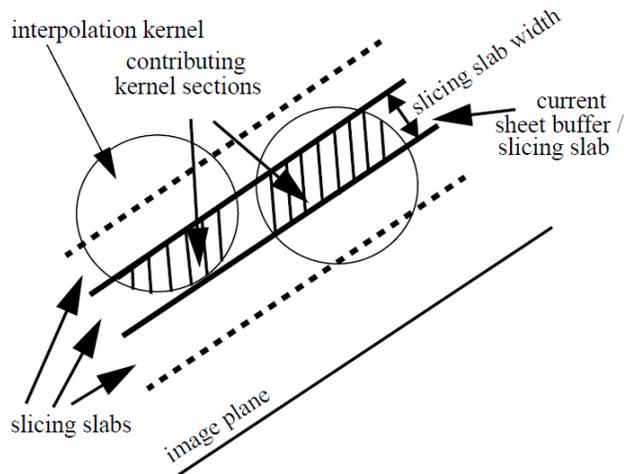


Figure 2.8: Slicing slab and sheet buffers are selected to be parallel to the image plane in order to tackle the color bleeding and popping artifacts [37]. © 1998 IEEE.

tackled by making view-aligned sheet-buffers by slicing slabs parallel to the image plane. The final image which was composited in front-to-back order can also benefit from occlusion map for culling. Figure 2.8 illustrates the way this method uses slabs and sheet buffers in order to perform splatting on volumetric data.

2.7 Transfer Functions

Often the volumetric data contains raw values. These values can represent different material properties, vectors, flow, etc. For example in the case of CT images they represent the x-ray absorption of the material at each point in space. There is usually some information in data that do not need to be visualized or need to be separated from the rest of the data. As a result this raw data needs to be classified to produce a more meaningful visualization.

Transfer functions are responsible for classifying raw data. These functions can vary but they essentially dismiss certain ranges of density (in case of CT) and map other ranges to visible colors and opacities. The topic of transfer functions has some overlap with segmentation. Various techniques have been

used in multidimensional transfer functions such as using histogram of first and second derivatives [17], statistical classifiers [50], Laplacian-weighted density histograms [42], region growing [21], Hessian matrix (A matrix of second partial derivatives) [16], etc. However, these techniques are not in the scope of this research.

Transfer functions can be applied to data in different ways. First approach is the pre-classification technique in which the colors and transparencies of voxels are determined prior to casting rays. This will ensure smooth edges and transitions between different material [64]. However, post-classification technique, which means applying a transfer function after interpolation, makes the rendered image sharper and more detailed by avoiding the interpolation to be applied to colors and opacities while sampling the volume [16].

2.7.1 Pre-Integrated Classification

Sampling the volume with a sampling rate higher than Nyquist rate guarantees that the signal can later be reconstructed correctly and prevents aliasing effect. However, when a transfer functions is being applied one cannot solely rely on the maximum frequency from the raw volumetric data. The minimum required sampling rate for correct evaluation of ray integral may increase by the non-linear properties of the transfer functions [16, 48]. Disregarding this problem can result in missing details or artifacts to be present in the final rendered image.

Pre-integrated classification is one of the ways to tackle this problem. By replacing the non-linear transfer function by a piece-wise continuous scalar function one can avoid the high frequency problem and consequently no under sampling will happen (For details see Hansen [16] pages 214-215). In this method, for each ray segment, a 1D table lookup retrieves the corresponding value and the integral will be calculated based on these values.

Chapter 3

Depth-of-Field Using Stochastic Sampling

In this chapter, we present the algorithm that we devised to create DoF effect. Krivanek [22] uses the idea that a high sampling rate of low resolution part of the input signal is not required as long as the Nyquist rate is preserved. Our approach uses this idea to reduce the amount of sampling required for blurring. This minimizes the rendering overhead and boosts the performance of our method. This will enable us to have real-time performance and minimizes the rendering overhead. Real-time performance ($\geq 10Hz$) can be achieved if DVR optimizations are used in conjunction with our method. However, here we present our results without taking advantage of such optimizations.

Traditional ray casting has an important property that our method exploits. The fact that many equidistant samples are taken along a ray very close to each other is used to produce blur effect. Our method maintains the 6 vital properties that are often seen as the criteria for an acceptable method for simulating DoF [1]. These 6 properties are:

- Choice of point spread function (PSF);
- Per-pixel blur level control;
- Lack of intensity leakage;
- Lack of depth discontinuity artifacts;

- Proper simulation of partial occlusion;
- High computational performance.

In the following sections, we will discuss our method in detail.

3.1 Simulating Out of Focus Blur in 3D

Convolution using averaging kernel (box blur), Gaussian kernels, etc. is one of the popular methods to produce a blurred image. In case of box blur, a square kernel is placed over each pixel of an image. The value of each pixel is then replaced by the average intensity of all the pixels in the kernel to produce the blurred image.

3.1.1 Using 2D Kernels in DVR

There are different approaches to simulate out of focus blur in 3D. One way is to place a 2D kernel, parallel to the viewing plane, and centered over each sampling location along each ray. Doing so simulates the physical process by which the light rays from the neighboring out of focus points contribute to the final color of the central point on the kernel. The average intensity of all the samples taken on the kernel plane and inside the kernel is calculated and returned as the intensity of the central sample location. The returned intensity is then used by the ray casting algorithm to determine pixel colors from samples.

Normally in ray casting one sample is taken on each sampling location along a ray. Instead, one can use a square kernel of size $n \times n$ to sample average kernel intensity at each sampling location. This is computationally very expensive and makes this approach impractical. However, the question is that do we need all these samples when our output is blurred? In this chapter, we show that an approximation with much fewer number of samples can be

used to create blur. In fact, our method requires only 2 samples per each kernel of size n^2 instead of n^2 samples to produce blur.

3.1.1.1 Kernel Sampling

Let us first define a few terms and symbols that will assist us in explaining the process of sampling a kernel.

Notation 1. Let $\vec{r}(x, y, t)$ be the current ray cast towards the volume and started from location $(x, y, 0)$ on the image plane.

Notation 2. Let t denote time which starts at 0 and each time we move to the next sampling location on $\vec{r}(x, y, t)$ it is increased by 1. Here t and $t + 1$ refer to two consecutive time-steps.

Notation 3. Neighboring rays of $\vec{r}(x, y, t)$ denoted as $\mathbf{R}_n(x, y, t)$ refers to all the rays that intersect the kernel that is placed on the ray $\vec{r}(x, y, t)$ at time t . This kernel is parallel to the image plane and centered on $\vec{r}(x, y, t)$.

Notation 4. Let $\mathbf{K}(x, y, t)$ be the set of all the n^2 intersection points of $\vec{r}(x, y, t)$ and its neighbors, $\mathbf{R}_n(x, y, t)$, with the kernel at time t . Each one of these intersections on the kernel is denoted as $k(i, j, t) \in \mathbf{K}(x, y, t)$ where \mathbf{K} is defined by:

$$\mathbf{K}(x, y, t) = \begin{pmatrix} k(0, 0, t) & k(0, 1, t) & \cdots & k(0, n-1, t) \\ k(1, 0, t) & k(1, 1, t) & \cdots & k(1, n-1, t) \\ \vdots & \vdots & \ddots & \vdots \\ k(n-1, 0, t) & k(n-1, 1, t) & \cdots & k(n-1, n-1, t) \end{pmatrix} \quad (3.1)$$

A neighboring ray of $\vec{r}(x, y, t)$, denoted by $\vec{r}_n \in \mathbf{R}_n(x, y, t)$, that intersects the kernel of size n^2 at $k(i, j, t) \in \mathbf{K}(x, y, t)$ can be obtained from $\vec{r}_n(x + i -$

$\frac{n-1}{2}, y + j - \frac{n-1}{2}, t)$. Also, $\vec{r}(x, y, t)$ intersects the kernel at the kernel's central point $k(\frac{n-1}{2}, \frac{n-1}{2}, t)$.

3.1.2 Neighborhood Sampling

In our method while following $\vec{r}(x, y, t)$ at each t only a single $k(i, j, t)$ is chosen among all n^2 possibilities. The volume is then sampled at the location of the chosen $k(i, j, t)$ instead of sampling all n^2 locations. Figure 3.1 illustrates this neighborhood sampling scheme with an example for $n = 3$.

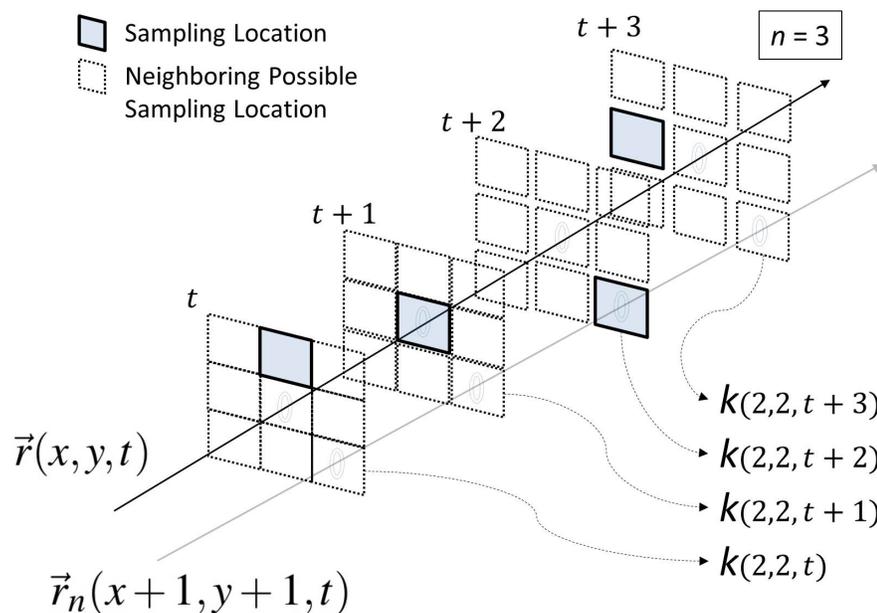


Figure 3.1: A different sampling location is chosen within a kernel on each time-step. Solid squares show the chosen sampling location on each time-step. Dotted squares show the potential sampling locations within a kernel. The sampled value at these sampling locations are treated and accumulated by the ray casting algorithm as if they are being read normally on $\vec{r}(x, y, t)$ ray. Divergence of the rays is to demonstrate perspective projection.

The distances between the consecutive sampling locations along $\vec{r}(x, y, t)$ are very small, and a different $k(i, j, t)$ is chosen for each time-step as shown in Figure 3.1. This results in an approximation of completely calculating the average of n^2 kernel averages (in case of box blur). It is possible to put weights on the different $k(i, j, t)$ in order to approximate different point spread

functions using other kernels such as a Gaussian kernel. Also, the shape of the kernel can change from square to other shapes to produce a desired effect.

3.1.3 Permutation-based Selection of a Single $k(i, j, t)$

Empirically, we found that complete randomness or pseudo-randomness in choosing each single $k(i, j, t)$ does not necessarily produce the desired results. In fact, it produces aliasing and there is a possibility of missing details of the volume. Repeating selection patterns are also not favorable for the same reasons. As a result, we used random permutations of sampling locations inside kernels to solve these two issues. Let us define a few symbols and terms to help us explain the permutation-based sampling scheme.

Notation 5. Let $\mathbf{Z}(t_i, n)$ refer to a window on $\vec{r}(x, y, t)$ consisting of n^2 consecutive time-steps associated with a kernel of size n^2 and starting at time t_i . There are exactly n^2 sampling locations in $\mathbf{Z}(t_i, n)$. By “associated” we mean a kernel of size n^2 is placed on each of the n^2 different sampling locations within $\mathbf{Z}(t_i, n)$. All these kernels are parallel to the image plane and centered on $\vec{r}(x, y, t)$.

Notation 6. Let $\mathbf{Z}^*(t_i, n)$ be a window on $\vec{r}(x, y, t)$ associated with a kernel of size n^2 . It is similar to $\mathbf{Z}(t_i, n)$ except that it consists of less than n^2 time-steps. For brevity, we use \mathbf{Z}_i notation when we refer to both $\mathbf{Z}(t_i, n)$ and $\mathbf{Z}^*(t_i, n)$ at the same time. Also, t_i and t_{i+1} refer to two different starting times and not two consecutive time-steps. $\mathbf{Z}(t_i, n)$ and $\mathbf{Z}^*(t_i, n)$ are mutually exclusive.

Notation 7. Let Ψ_n be the set of all the permutations of n^2 with $(n^2)!$ distinct elements. Let $\omega_n \in \Psi_n$ be a single permutation of n^2 distinct elements.

Notation 8. Let $g : \omega_n \rightarrow \mathbf{K}(x, y, t)$ be an injective map that takes a single element from ω_n and outputs a sampling location $k(i, j, t)$. The map g is defined when ω_n and $\mathbf{K}(x, y, t)$ have the same size.

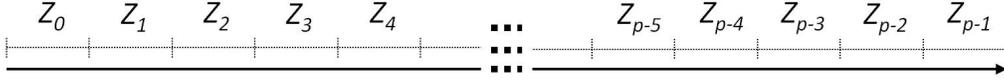


Figure 3.2: A ray is divided into windows associated with kernels of size n^2 . We should note that n can change during traversal of the ray in order to produce different levels of blur. Each \mathbf{Z}_i can have a different n associated with it.

Figure 3.2 shows a ray that has been divided into p sections called \mathbf{Z}_0 to \mathbf{Z}_{p-1} . Each different \mathbf{Z}_i may have a different associated n based on the desired blur level and consequently a different length. A new permutation $\omega_n \in \Psi_n$ is chosen pseudo-randomly every time we enter a new \mathbf{Z}_i associated with a kernel of size n^2 .

On each consecutive time-step in \mathbf{Z}_i the next unused element in ω_n is used with the mapping g to select a unique $k(i, j, t)$. This permutation-based selected $k(i, j, t)$ is the location on which the volume will be sampled on that time-step. This guarantees that in $\mathbf{Z}(t_i, n)$ all the n^2 distinct sampling locations on the kernel are chosen exactly once. It also guarantees that in $\mathbf{Z}^*(t_i, n)$ all the chosen $k(i, j, t)$ are different. After entering \mathbf{Z}_{i+1} a new permutation ω_n is generated and the algorithm continues in a similar manner.

Our initial single samples on each kernel produce a certain amount of noise in low frequency areas. Although Cook [6] states that human adapts well to noise, we added an additional step to further eliminate the noise.

3.1.4 Central Reflection Sampling

In full kernel sampling if we take n^2 samples for a kernel of size n^2 and m is the number of samples taken from the volume in \mathbf{Z}_i , then m equals n^4 in $\mathbf{Z}(t_i, n)$ and $n^2 \leq m < n^4$ in $\mathbf{Z}^*(t_i, n)$. In our method by replacing the average of a kernel with a single permutation-based sample on each time-step, we under-sample the kernel by only taking $\frac{m}{n^2}$ samples instead of taking m samples in \mathbf{Z}_i .

A kernel can cover a variety of values such as different materials as well as empty space. Using a value as the representative of the kernel’s average can contribute to the noisiness of the image if the value is very different from the actual average. It is shown here that using central reflection sampling can tackle this problem and considerably reduce the amount of noise in the final image.

On each time-step, t , the reflection of the chosen sampling location through the central point of the kernel is found and sampled as well. Then the average of the value sampled at the location of the original sample $k(i, j, t)$ and the value sampled at the point reflection of the sample at $k'(i', j', t)$ is calculated and processed as the current sample taken from the volume. This method has shown to greatly reduce the amount of noise.

Figure 3.3 illustrates the idea behind central reflection sampling by expanding on our previous example of Figure 3.1. Solid squares represent our normal stochastic sampling locations chosen based on the current permutation. Diagonally hatched squares show the sampling location, which is the point reflection of the permutation-based sampling location across the central point of the kernel.

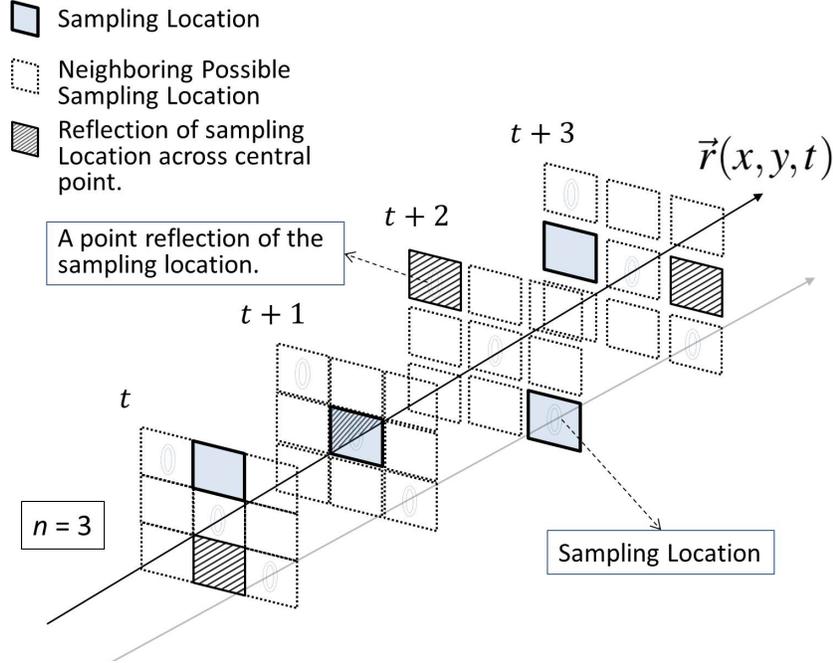


Figure 3.3: A second sampling location is calculated by finding the point reflection of the stochastic sampling location across the central point of the kernel. Solid squares are the permutation-based sampling locations. Hatched squares are the secondary sampling locations found by central reflection. The final value that will be used by the ray casting algorithm on each time-step is the average of the two values read on these two locations on the corresponding time-step.

3.1.5 Calculating Sampling Locations

We need to find the locations of the stochastic sample $k(i, j, t)$ and its point reflection $k'(i', j', t)$ at each time-step. Having variable kernel sizes make it infeasible to keep all the neighboring sampling locations at all time. As a result, we need to calculate the locations of $k(i, j, t)$ and $k'(i', j', t)$ on each time-step.

By using vector calculations as shown in Figure 3.4, locations of $\vec{C}' = k(i, j, t)$ and $\vec{E}' = k'(i', j', t)$ are found. Here \vec{AB} is the ray that is being traversed. Kernel plane K intersects \vec{AB} at \vec{A}' and $K \parallel \text{Image Plane}$. The normal vector of the plane K , denoted as \vec{h} , is also the same as the normal vector of the image plane.

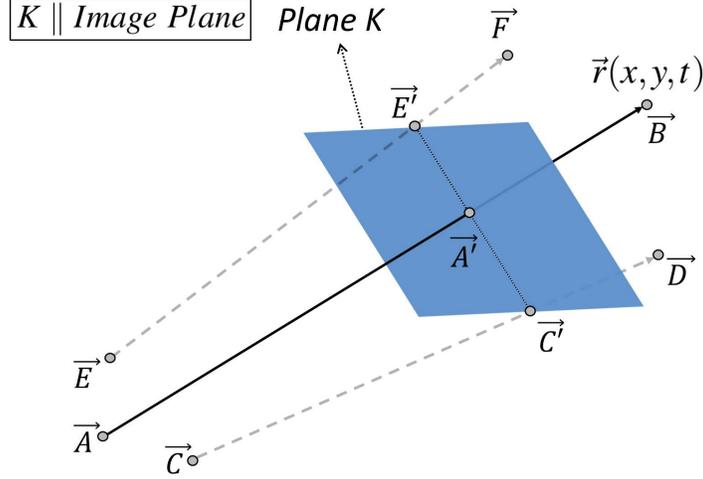


Figure 3.4: Finding the permutation-based sampling location for \overrightarrow{AB} on its neighboring ray \overrightarrow{CD} . \vec{C}' is the stochastically chosen sampling location and \vec{E}' is its point reflection through the central point, \vec{A}' .

The plane equation of K is:

$$\text{Plane } K: (\overrightarrow{A'C'}) \cdot \vec{h} = 0 \quad (3.2)$$

The line equation of \overrightarrow{CD} is:

$$\overrightarrow{CD}: \vec{C}' = \alpha \cdot \vec{l} + \vec{C} \quad \text{where} \quad \vec{l} = \frac{\overrightarrow{CD}}{|CD|} \quad (3.3)$$

Intersecting plane K and ray \overrightarrow{CD} gives us the α corresponding to our stochastic sampling location $\vec{C}' = k(i, j, t)$ as shown in:

$$\vec{C}' = \frac{(\overrightarrow{CA'}) \cdot \vec{h}}{\vec{l} \cdot \vec{h}} \cdot \vec{l} + \vec{C} \quad (3.4)$$

The location of $\vec{E}' = k'(i', j', t)$ can now be found simply by using symmetry. \vec{E}' is the point reflection of \vec{C}' through \vec{A}' .

We should note that although $\vec{E}' = k'(i', j', t)$ is guaranteed to be inside the kernel, it might not exactly fall on one of the $n^2 - 1$ neighboring rays of $\vec{r}(x, y, t)$. This is due to perspective projection and does not cause any problem.

3.2 Fast Sampling of the Permutation Space

In order to accomplish our stochastic sampling without visible aliasing, we used permutations of sampling locations inside a kernel. For each \mathbf{Z}_i associated with a kernel of size n^2 we need a pseudo-random permutation ω_n of size n^2 while on each time-step t , we need only one element from ω_n .

Most methods for generating permutations require remembering previously generated elements, which is infeasible due to GPU’s hardware limitations such as limited number of registers, slow global memory, etc. Other methods such as loop-less methods either require calculating $(n^2)!$ or they are not suitable for real-time rendering [49, 10]. Some methods that use mathematical sequences to generate permutations could have been ideal, however, they generate a single randomized permutation over and over [33].

None of these methods are suitable for our rendering task due to our limited memory and performance requirements. We generate permutations one element at a time for each time-step. We do not need to keep track of the elements we generated so far, and we do not need to calculate $n^2!$. The requirements for our method is to have a seed permutation for different permutation lengths less than or equal to n_{max}^2 where n_{max} is the maximum size of the kernel for maximum desired blur level, a set of prime numbers, a random number generator, and time-step counter t .

3.2.1 Seed Permutations

Generating DoF requires different levels of blur. In our method blur level directly depends on the size of the kernel. As a result, one need to have access to seed permutations of different sizes. A one-dimensional array of size n_{max}^2 can hold all our seed permutations for $n \in [1, n_{max}]$. An example of $n_{max} = 6$ is shown in Figure 3.5. In our implementation for all $n \in [1, n_{max}]$,

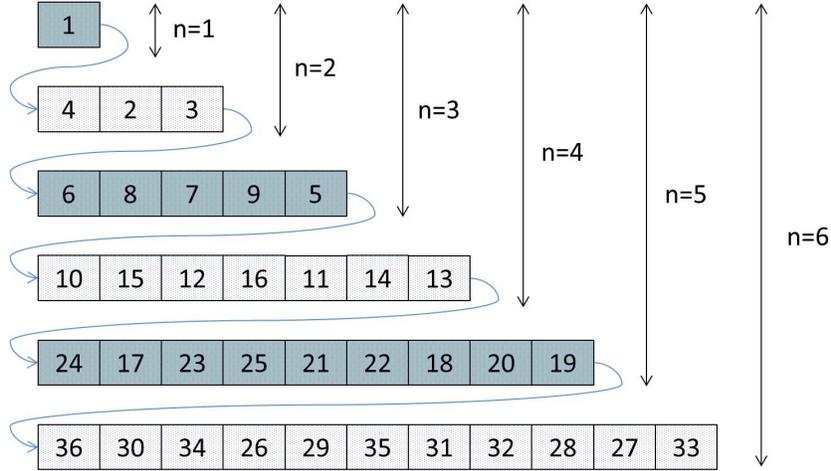


Figure 3.5: One of many different ways of storing n different randomized permutation seeds in an array of size n_{max}^2 . In this example $n = 6$. The seed permutations are not required to be the same as ones presented here.

the first n^2 elements of the one dimensional seed permutation array contain a permutation of numbers between 1 and n^2 . If $\lambda_i \in \mathbb{N}$ is the value at index i of the permutation array, $\mathbf{\Lambda}$, then $\mathbf{\Lambda}$ is defined as follows:

$$\mathbf{\Lambda} = \{\mathbf{\Lambda}_n \mid n \in [1, n_{max}] \in \mathbb{N}\} \quad (3.5)$$

$$\mathbf{\Lambda}_n = \{\lambda_i \mid i \in \mathbf{I}_n \wedge (n-1)^2 < \lambda_i \leq n^2\} \quad (3.6)$$

$$\mathbf{I}_n = \{i \mid (n-1)^2 < i \leq n^2\} \quad (3.7)$$

$$\forall \lambda_i \forall \lambda_p \in \mathbf{\Lambda}, \lambda_i = \lambda_p \Leftrightarrow i = p \quad (3.8)$$

3.2.2 Set of Prime Numbers

We store a set of prime numbers with numbers larger than n_{max}^2 in constant memory on GPU. Being larger than n_{max}^2 is important since in this case the largest common multiple of the prime numbers with all the numbers from 1 to n_{max}^2 range is going to be 1. This means that one can use them in the way that we describe in Algorithm 3.1 to read our seed permutation each time in a different order.

In order to completely eliminate any visible aliasing, we need enough di-

iversity among the permutations that we generate. This diversity depends on the number of elements we keep in our set of prime numbers. Algorithm 3.1 describes how for a \mathbf{Z}_i that is associated with a kernel of size n^2 the remainder of a randomly chosen prime number over n^2 is used. In our implementation, $n_{max} = 30$ and the set of prime numbers contains 732 prime numbers from 1000 to 7000.

3.2.3 Generating Permutations

The current element of permutation ω_n at time t is calculated using Algorithm 3.1 and denoted as *element*. Moreover, here the seed permutation array is denoted as \mathbf{S} and the set of prime numbers of size N_p is denoted as \mathbf{P} . In this algorithm, the fact that single elements of a seed permutation can also be used as a pseudo-random index for the seed permutation itself has also been used to read the seed permutation each time in a different order. In order to generate a pseudo-random number, we used Multiply With Carry (MWC) method proposed by Marsaglia *et al.* [34]. This method, which is denoted as *MWCRand()* in Algorithm 3.1, requires minimal computation and generates pseudo-random numbers with a very long period. In this pseudo-code α is an offset to make sure that when a transition between two blur levels happen, a new permutation is started from its beginning.

In summary, although we do not claim that our sampling of permutation space follows a specific distribution, it has proven to be effective in eliminating visible aliasing and moiré patterns to the point that it is undetectable by the eye. Figure 3.8 compares our permutation based sampling results with purely pseudo-random selection of the first sampling location on a kernel. Results show that visible aliasing has been eliminated by using permutation-based sampling.

Algorithm 3.1 Generating one pseudo-random element of a permutation at a time inside the ray casting loop.

Input: seed permutation \mathcal{S} , set of prime numbers \mathbf{P} with size N_p , kernel size n^2

Output: current element in a permutation of size n^2 denoted as *element*

```

 $\alpha \leftarrow 0$ 
for  $t = 0 \rightarrow \infty$  do                                \\\ Loop for casting a single ray
   $\vdots$                                                 \\\ Ray casting initializations
  if  $n$  has changed then                                \\\ Handling  $\mathbf{Z}^*(t_i, n)$  transition
     $\alpha \leftarrow t \bmod n^2$ 
  end if
  if  $((t - \alpha) \bmod n^2) = 0$  then                    \\\ New permutation
     $r_1 \leftarrow MWCRand()$ 
     $r_2 \leftarrow MWCRand()$ 
     $p \leftarrow \mathbf{P}[r_1 \bmod N_p]$ 
  end if
   $index \leftarrow (\mathcal{S}[(t + 1) * p] \bmod n^2 + r_2) \bmod n^2$ 
   $element \leftarrow \mathcal{S}[index]$ 
   $\vdots$                                                 \\\ Other ray casting tasks
   $t \leftarrow t + 1$ 
end for

```

3.2.4 Transition Between Different Blur Levels

In addition to eliminating aliasing, permutations also helped us guarantee that in each $\mathbf{Z}(t_i, n)$ with blur level n , exactly n^2 unique sampling locations are devised in a pseudo-random order. These samples uniformly cover the whole area of the kernel.

In $\mathbf{Z}^*(t_i, n)$ where a change in kernel size and consequently a change in blur level occurs, a new permutation is devised and used from the current sampling location and the previous permutation is no longer followed. When a new permutation is devised, it is important to make sure that sampling starts from the beginning of the new permutation. Failing to do so will result in artifacts and errors that can be seen specially when a transition from in focus to out of focus regions happen.

Figure 3.6 illustrates the process of changing blur levels with an example. Here $\mathbf{Z}^*(t_i, n_j)$, $\mathbf{Z}(t_{i-1}, n_j)$, and $\mathbf{Z}(t_{i-2}, n_j)$ are associated with a kernel of size

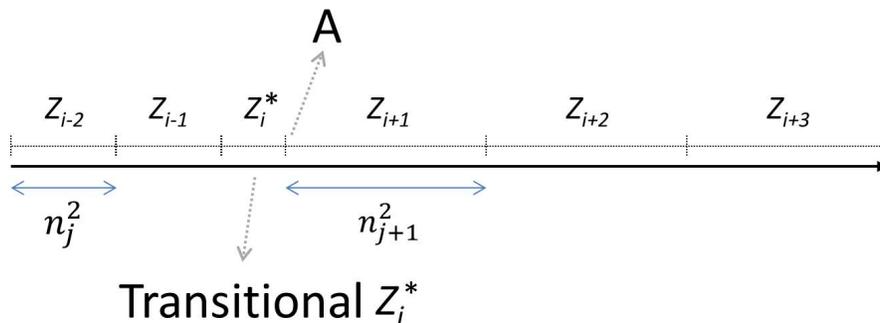


Figure 3.6: Transition of different blur levels. In this example, all three regions $\mathbf{Z}(t_{i-2}, n_j)$, $\mathbf{Z}(t_{i-1}, n_j)$, and $\mathbf{Z}^*(t_i, n_j)$ are associated with a kernel of size n_j^2 . However, a change of blur levels from n_j^2 to n_{j+1}^2 while traversing $\mathbf{Z}^*(t_i, n_j)$ resulted in abandoning the rest of $\mathbf{Z}^*(t_i, n_j)$ and starting $\mathbf{Z}(t_{i+1}, n_{j+1})$ at the point labeled as A .

n_j^2 . However, the length of $\mathbf{Z}^*(t_i, n_j)$ is shortened due to a change in kernel size from n_j^2 to n_{j+1}^2 where $n_j^2 \neq n_{j+1}^2$. A new permutation of size n_{j+1}^2 is devised and followed from A before taking all n_j^2 samples in $\mathbf{Z}^*(t_i, n_j)$.

3.2.5 Varying Blur Levels for Simulating Depth-of-Field

In order to simulate the DoF effect, we need to vary the blur level based on the distances from the image and lens planes. The theory of DoF has been discussed in many textbooks and papers so we are not going to go into details of lens theory.

Majority of volume rendering algorithms use thin lens model which implies that the thickness of the lens compared to its focal length is negligible and no light refraction happens inside the lens. It also assumes that entrance and exit pupils of the lens have the same size. This model makes the calculations for ray casting easier.

Light travels from each visible point on an object through the lens. For each point on an object a footprint of light is formed on the image plane. If this footprint is small enough to be perceived as a single point then that point on the object is said to be in focus. In contrast, if the point on the object is

not between near and far focus planes, then the light footprint is larger than a point on the image plane. This is called the circle of confusion (CoC). In our system, we used the diameter of CoC in direct correlation with the size of kernel. Although we used square shaped kernels, it is possible to use weighted kernels to simulate circular effect.

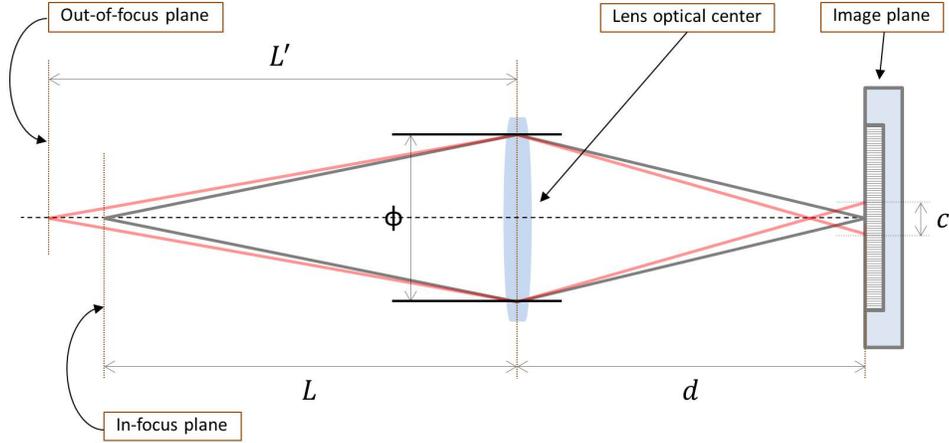


Figure 3.7: A thin lens model. Here Φ is the aperture diameter. Points on the In focus plane form points on the image plane while points on the Out of focus plane form circles of diameter “ c ” on the Image plane. Image is taken and modified from [41].

Let f be the focal length of the lens which can be calculated from:

$$\frac{1}{f} = \frac{1}{L} + \frac{1}{d} \quad (3.9)$$

The diameter of the CoC created by objects at distance L' from the lens' optical center is calculated by:

$$c = \Phi \frac{|L' - L|}{L'} \frac{f}{L - f} \quad (3.10)$$

Equation 3.10 can be simplified if the magnification m of the lens is known. Lens magnification can be calculated using:

$$m = \frac{f}{L - f} \quad (3.11)$$

Finally, the diameter of the CoC can be calculated using:

$$c = \Phi m \frac{|L' - L|}{L'} \quad (3.12)$$

by knowing the aperture diameter Φ , lens magnification m , focal length f , the distance between optical center of the lens and object plane L' , and the distance between optical center of the lens and focus plane L as shown in Figure 3.7.

By knowing the distance that has been traveled from the image plane towards the negative z direction of the camera coordinates (away from camera and towards the volume), it is possible to determine the kernel size at each sampling location during ray traversal. By converting the coordinates of the current sampling location to camera coordinates, current distance from the image plane can be calculated. Also, one can accumulate the distances that the ray travels towards negative z direction for each sample. It is possible to determine the kernel size by using Equation 3.12 based on the accumulated distance.

3.3 Blurred Edge Highlighting

While the average of a sample and its point reflection is calculated, it is possible for either or both of these samples to be taken from the empty space around the volume. This happens around the edges of the rendered object in blurred areas. As a result, a color is assigned to the empty space inside the transfer function. This will facilitate the calculation of the average of the sampled values. In an RGBA color scheme, the value of A , which is reserved for opacity, is used to make the empty space invisible. For example, in a 4 byte RGBA color, a value of $(128, 128, 128, 0)$ can be used as a neutral grey empty space to avoid any color or luminosity shift.

By using different colors assigned to empty space while keeping the opacity

0, the edges of the blurred areas can be highlighted with no extra computation. For example, in the same 4 byte RGBA scheme, assigning $(0, 0, 0, 0)$ would darken the edges or assigning $(255, 255, 255, 0)$ would lighten the edges. Trivially assigning any value with $A \neq 0$ would produce fog. Figure 3.9 shows the effect of our edge highlighting method applied to the rendering of a cardiovascular arteries CT data set. Researchers have proposed different methods with extra computational cost to create such volumetric halos and showed that it positively affects perception [3, 56], while in our method this effect exists at no additional cost.

3.4 Discussion of the Proposed Model

Under-sampling can produce aliasing, noise, and potential missing of details. By using permutations, we have eliminated the visible aliasing that was created by patterns in stochastic sampling. Noise has also been dealt with by using central reflection sampling as discussed in Section 3.1.4.

Since we are not sampling each ray on its own path, someone might argue that it is possible to miss very little details in blurred areas. Permutation-based choosing of the sampling locations inside each kernel forces all neighboring rays of a ray to compensate for what is perceived as a small probability of missing of details in blurred areas in the following manner.

Most of \mathbf{Z}_i while traversing a single ray are of the form $\mathbf{Z}(t_i, n)$. While sampling $\vec{r}_n(x, y, t)$ it is guaranteed that a single sample on the ray itself will be taken once in each $\mathbf{Z}(t_i, n)$. This is also possible on $\mathbf{Z}^*(t_i, n)$ but not guaranteed. For all $\vec{r}_n \in \mathbf{R}_n(x, y, t)$ it is guaranteed that a stochastic sample from each of the $n^2 - 1$ neighboring rays will be on $\vec{r}(x, y, t)$ in a $\mathbf{Z}(t_i, n)$ exactly once. Moreover, in almost all cases the point reflection of a single stochastic sample on each of the neighboring rays will also fall on $\vec{r}(x, y, t)$. The worst case scenario would be that in a single \mathbf{Z}_i all n^2 samples of the

n^2 rays in the kernel that are supposed to be on ray $\vec{r}(x, y, t)$ happen at the same time. However, using seed-permutations as well as unique seed points for random number generation using MWC for each ray will prevent such events by guaranteeing different permutations for different rays in a single \mathbf{Z}_i . As a result, the probability of missing minute details in out of focus areas is extremely low.

3.5 Implementation

We used CUDA and OpenGL on a laptop with GeForce GTX 680M and an Intel Core i7 running at 2.4GHz. Although it is possible to implement this method with CG or GLSL, we chose CUDA to have more control over the available resources on the GPU. Volumetric data was stored in texture memory while the array of prime numbers and seed-permutations reside in constant memory.

Our method augments ray casting with DoF. Except for early ray termination (ERT), we did not take advantage of any other DVR speed optimization method. Also, shading has not been used in the results that we present here. A single cube containing the entire volume was rendered to two textures to obtain ray start and end coordinates [23]. Moreover, a post-interpolative piecewise linear real-time transfer function [16] has also been used to convert from intensity values to colors and opacities. In our results when no DoF is applied, the DoF calculation is bypassed entirely. This is different than calculating the DoF for a 0 *mm* aperture diameter where the overhead of calculating DoF might still slow down the rendering. Also, we have limited the maximum aperture opening to 40 *mm* in our implementation.

Finally, our method is compatible with other optimization and enhancement methods for DVR. However, some of these methods might need simple modifications to take advantage of our method. For example, many meth-

ods have been devised to improve performance of ray casting by skipping the empty space. In our method, blurring effect is created by the rays sampling the volume and a subset of rays that are sampling the empty space around the volume. As a result, in order to incorporate empty space skipping it is important not to skip the rays that are contributing to the blur around the volume. If n_{max} is the maximum kernel size, then the rays that are passing into the empty space and no further than $\frac{n_{max}\sqrt{2}}{2}$ from the volume should still be cast. This distance is half of the diagonal of the biggest kernel.

3.6 Results

The frame rate of a rendering algorithm can vary due to different parameters, data, implementation technique, and hardware. This makes frames per second (FPS) an inaccurate measure of performance to compare methods that are implemented in different settings. However, FPS can provide us with other insights such as whether a method is real-time or not, or the amount of change in rendering speed because of extra computation. The frame rate in our DoF method when a volume is rendered with maximum aperture opening is approximately between 40% to 75% of the frame rate when the same volume is rendered without DoF.

Changes in the frame rate are not solely the result of the DoF calculations. Other factors such as the distance between samples on a single ray also play an important role in how much the frame rate changes. The Nyquist-Shannon sampling theorem states that at least $2 \times f_{max}$ samples should be taken from a signal with maximum frequency f_{max} to avoid aliasing in signal reconstruction. Since we are already under-sampling our kernels, Nyquist sampling rate based on f_{max} of the volume will likely produces noise in the final image. In order to reduce this noise, one can compensate by taking more samples on each ray in addition to the method discussed in Section 3.1.4. The adverse effect

of taking $2 \times (2 \times f_{max})$ samples on performance has already been considered whenever the changes in frame rates are presented in this chapter. Also, details of parameters, frame rate, and changes in the frame rate are presented in the captions of the images for the following results.

Figure 3.9 shows a close up view of the patient’s heart. In these images blurred-edge-highlighting capability of our method to highlight blurred arteries is demonstrated. Figure 3.10 shows a set of 3 images rendered with full aperture opening from the same heart CT data as Figure 3.9. This set of images present the rendering of cardiovascular system around heart using our DoF method compared to plain rendering without DoF.

Figure 3.11 shows a set of 6 images produced by our method. Figures 3.11a, 3.11b, and 3.11c are showing both knees of a patient with different aperture openings. These three images aim to illustrate how our method can help separating the interfering background from the foreground.

Figures 3.12 and 3.13 compare the speed and quality of our stochastic DoF generation with full-kernel sampling as a reference. Full-kernel sampling is to sample all n^2 samples on each individual kernel. This can be seen as a brute force method to perform full sampling over the area of the lens. Figures 3.12a, 3.12b, and 3.12c show in order a scene without DoF, with DoF generated with full-kernel sampling, and with DoF generated by our method. Figures 3.13a and 3.13b are 2.5 times magnified crops of the full-kernel DoF compared with our method. Figures 3.13c, and 3.13d are 14 times magnified, while Figures 3.13e, and 3.13f are 56 times magnified crops of the full-kernel DoF compared with our method.

In the results that are presented here details about the time to generate each frame, size of data, and type of data is given in the caption of the Figures.

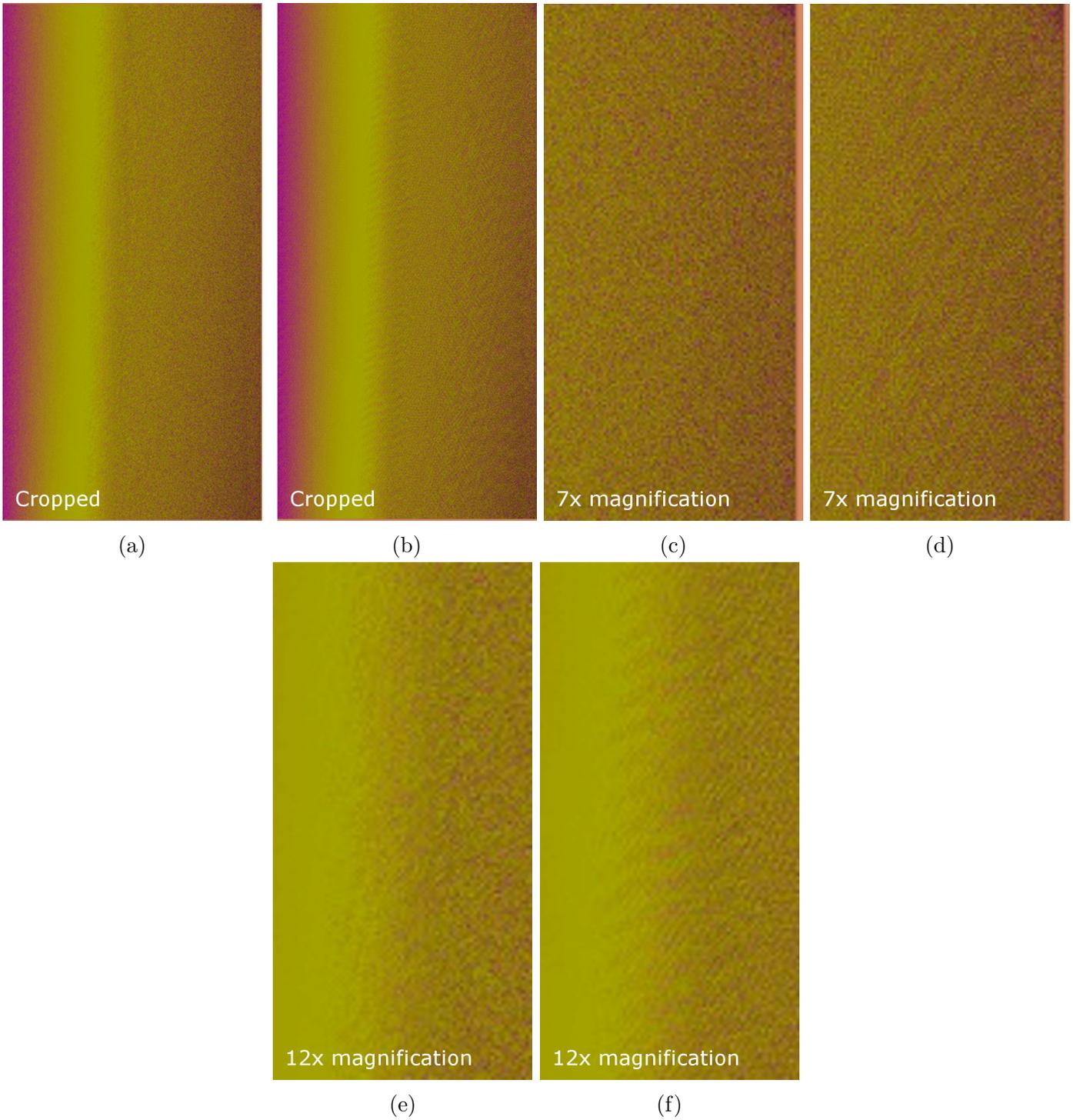


Figure 3.8: Comparison of permutation based sampling with purely pseudo-random sampling. Figures 3.8a, 3.8c, and 3.8e are rendered with our proposed permutation based sampling. Figures 3.8b, 3.8d, and 3.8f are rendered with pseudo-random selection of the first sampling point. Visible aliasing that can be seen in pseudo-random sampling has been removed by the permutation-based sampling.

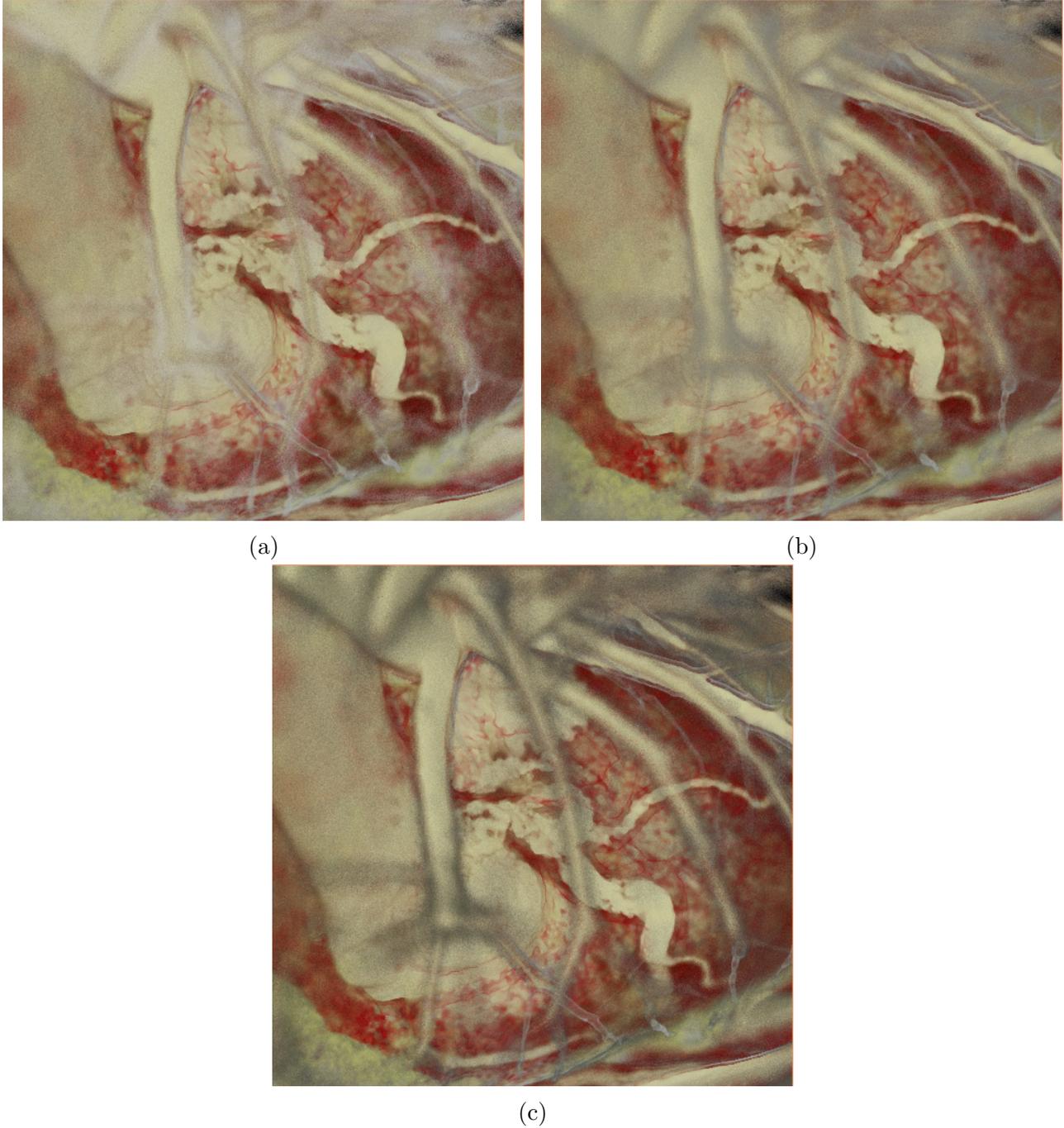


Figure 3.9: For all 3 images the number of rays cast is 512^2 and 1350 samples were taken on each ray while the lens magnification is $m = 2$. Figures 3.9a, 3.9b, and 3.9c show a side view of a $512 \times 512 \times 308$ heart CT data set. All images are rendered at ≈ 3.78 FPS ($264.5ms/frame$). These figures show our method’s ability to lighten or darken edges of blurred areas without any additional computation in order to enhance perception. Figure 3.9b shows the neutral rendering of the edges while in 3.9a the edges are lightened and in 3.9c they are darkened.

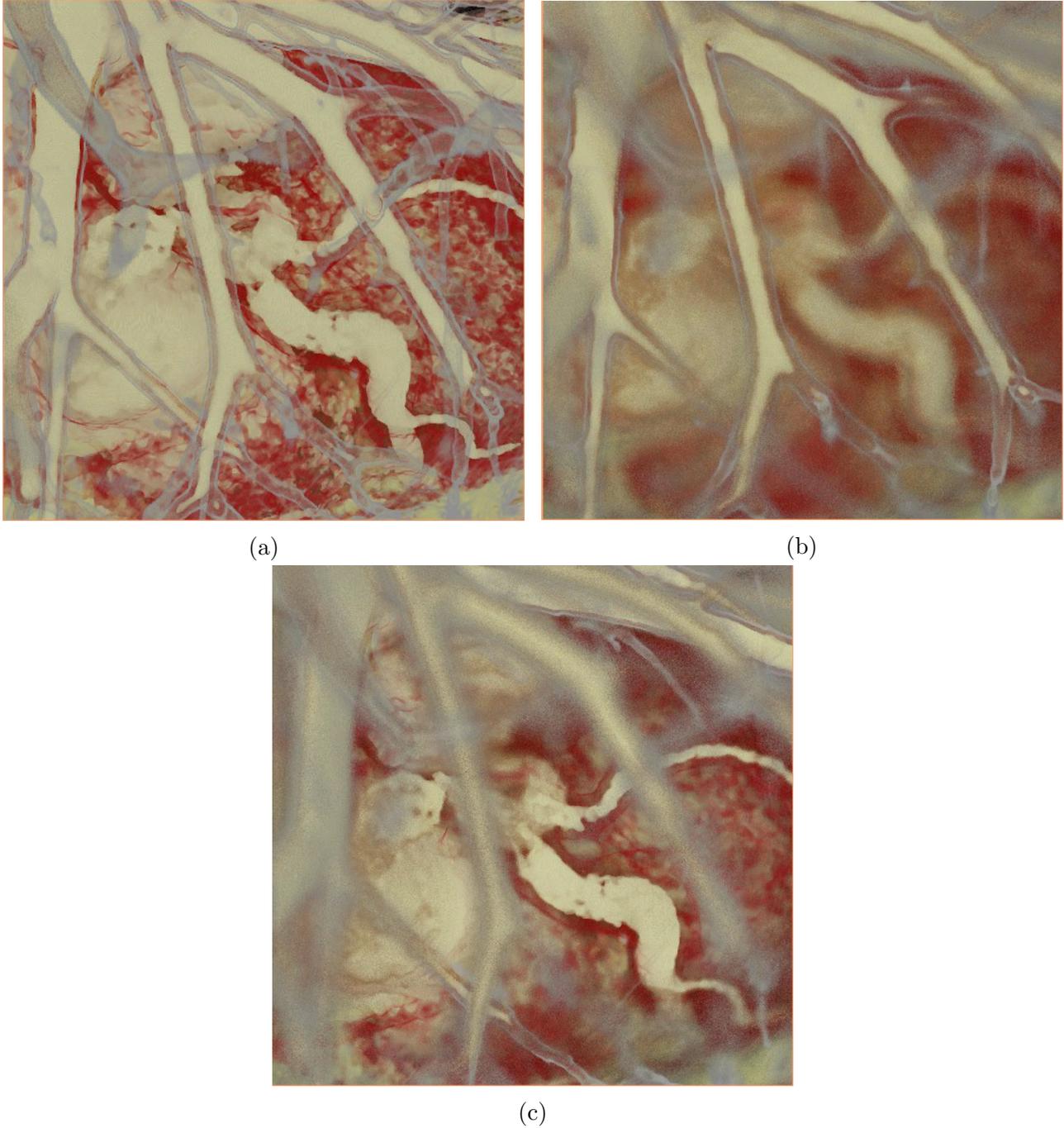
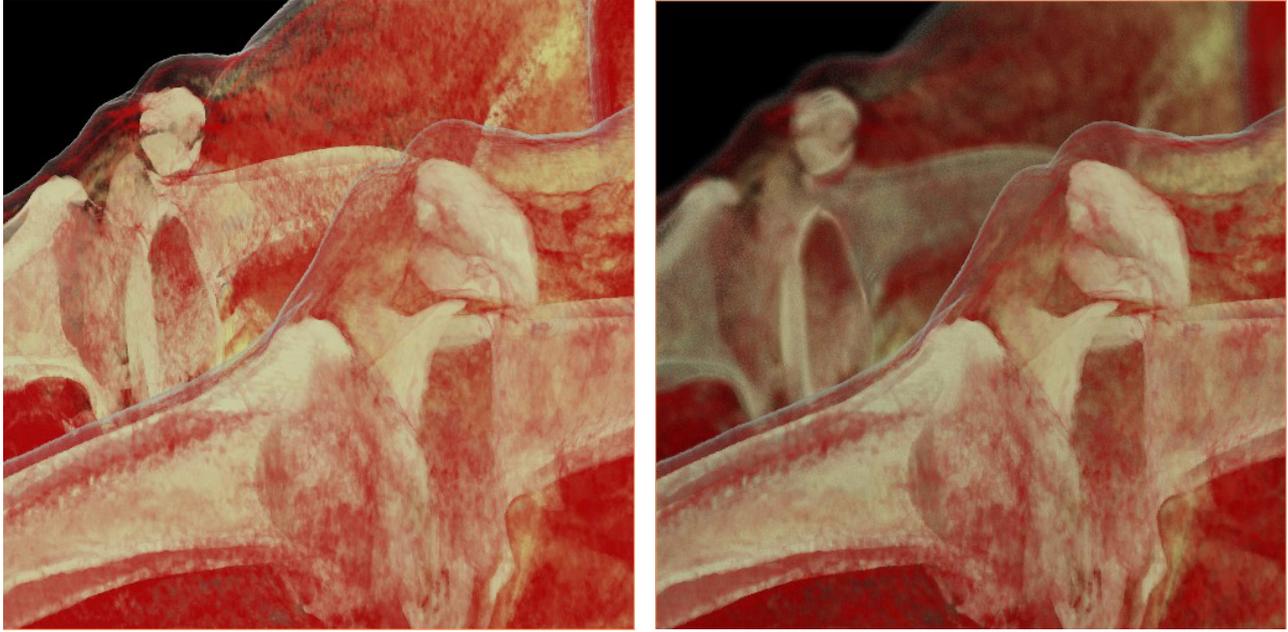
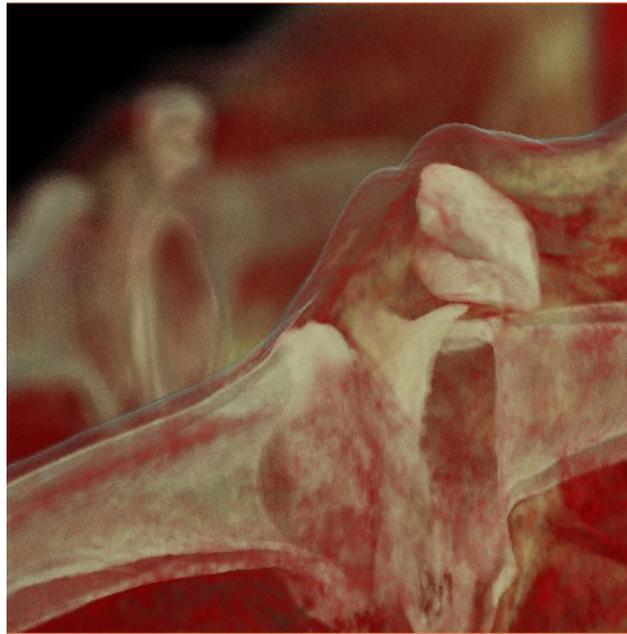


Figure 3.10: For all 3 images the number of rays cast is 512^2 and 1350 samples were taken on each ray while the lens magnification is $m = 2$. Figures 3.10a, 3.10b, and 3.10c show a side view of heart and cardiovascular system from a $512 \times 512 \times 308$ CT data from Osirix [40]. Figure 3.10a shows the scene without any DoF rendered at ≈ 7.55 FPS ($132.3ms/frame$). Figure 3.10b has full DoF effect (aperture diameter = $40mm$) and focused on the foreground arteries while rendered at ≈ 3.22 FPS ($310.2ms/frame$ and $\approx 42\%$ of the plain rendering speed). Figure 3.10c has full DoF effect (aperture diameter = $40mm$) while focused on the background arteries and rendered at ≈ 4.79 FPS ($208.8ms/frame$ and $\approx 63\%$ of the plain rendering speed).



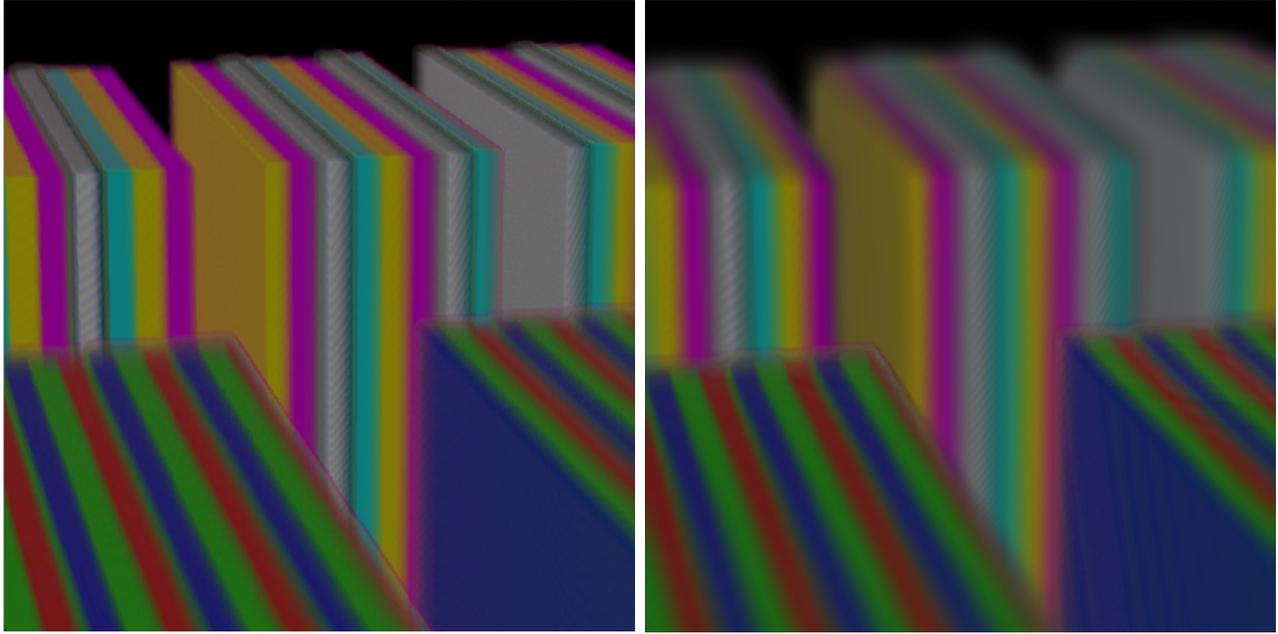
(a)

(b)



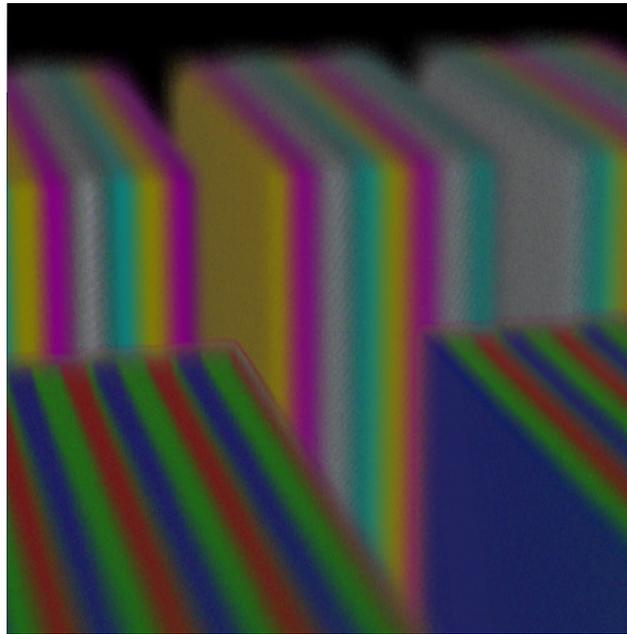
(c)

Figure 3.11: For all 3 images the number of rays cast is 512^2 and 1350 samples were taken on each ray while the lens magnification is $m = 2$. Figures 3.11a, 3.11b, and 3.11c show both knees of a person from a $512 \times 512 \times 912$ CT data from Osirix [40]. Figure 3.11a shows the scene without any DoF rendered at ≈ 7.7 FPS ($129.87ms/frame$). Figure 3.11b has moderate DoF effect (aperture diameter = $20mm$) while rendered at ≈ 3.8 FPS ($263.15ms/frame$ and $\approx 48\%$ of the plain rendering speed). Figure 3.11c has shallower DoF (aperture diameter = $40mm$) and rendered at ≈ 3.1 FPS ($322.58ms/frame$ and $\approx 40\%$ of the plain rendering speed).



(a)

(b)



(c)

Figure 3.12: Quality and speed of generating results from our DoF method is compared to full kernel sampling (brute force ground truth) using a 256^3 volume. There are 512^2 rays cast simultaneously and 1350 samples are taken on each ray. Figure 3.12a shows the plain image with no DoF rendered at ≈ 5.14 FPS ($194.5ms/frame$). Figure 3.12b shows the results of full kernel sampling (reference) rendered at ≈ 0.01 FPS ($104044.6ms/frame$ or 1 minute and 44 seconds per frame). Figure 3.12c was rendered at ≈ 2.58 FPS ($387.8ms/frame$) using our stochastic sampling DoF method and same camera setting as Figure 3.12b. Our method was rendered 268 times faster than full kernel sampling and at 50.1% of the speed of rendering the scene without any DoF. Both Figures 3.12b and 3.12c are rendered with aperture diameter of $40mm$.

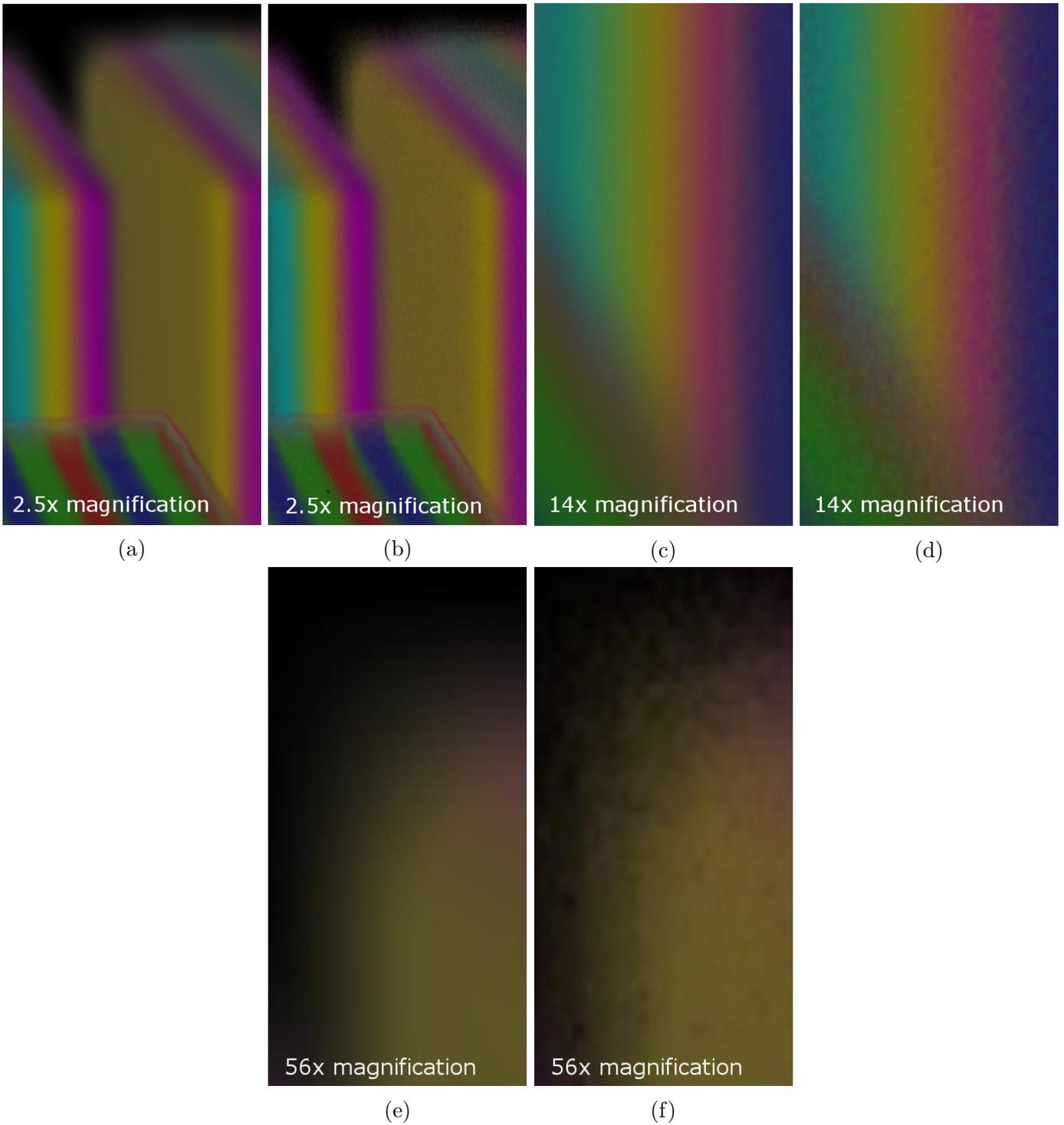


Figure 3.13: Quality and speed of generating results from our DoF method is compared to full kernel sampling (brute force ground truth) using a 256^3 volume. There are 512^2 rays cast simultaneously and 1350 samples are taken on each ray. Figures presented here show a comparison between the quality of our method and significantly slower full kernel sampling in pairs of two. Each two similar images have the same magnification factor as well. Figures 3.13a, 3.13c, and 3.13e are magnified from 3.12b, while Figures 3.13b, 3.13d, and 3.13f are magnified from 3.12c.

3.7 Conclusion

In this chapter, we presented a method to create blur in interactive DVR, and we have used it to render images with DoF. Our main goal was to create a flexible way of incorporating DoF as a depth cue in DVR so it can improve perception of depth in conjunction with other depth cues. The method proposed here satisfies the criteria for an acceptable DoF simulation proposed by Barsky [1].

In this method there is no need for any pre-computation that involves the volumetric data. The only preparation step is to store the array of seed permutations and a subset of prime numbers in memory. It is also possible to hard code both of these arrays. As a result, our algorithm does not limit the developer by introducing extra preprocessing time. The proposed algorithm is compatible with different optimization methods such as empty space skipping and image enhancement methods such as various types of shading.

Previously there was no practical methods of augmenting ray casting with blur capabilities. Our proposed ray casting based method operates in parallel at ray level (similar to traditional ray casting) with no need for any synchronization or dependency among different processes and threads. The slice based approach is not required for our method and consequently it is possible to blur any portion of any extent in the volume, which was unavailable in previously proposed methods.

Although the results presented here only show DoF effect created by our method, in future it is possible to create other effects in DVR using our method since each ray is blurred separately. One of these effects is the tilt effect [15]. In this effect the lens plane is not parallel to the image plane creating a variation of blur levels even on the surfaces parallel to the lens. This effect can change the perception of distance and size [15]. Moreover, the control and flexibility

available in this method provide the opportunity for new and novel approaches to enhance perception in direct volume rendering.

Chapter 4

Directed Visual Perception

In the past many methods have been proposed to enhance DVR. Many of them have been targeting rendering speed and quality while some others have been targeting visual perception. The latter can be enhanced through naturally occurring visual cues such as depth of field (DoF), or artificial cues such as pseudo-chromadepth. Both of these types of cues require specifically designed algorithms to be simulated in real-time.

In this chapter, we propose two new artificial visual cues that can be incorporated in DVR at a very low cost. Also, we will discuss their variants and combinations with each other in addition to their combination with other natural and artificial cues. These cues enhance perception, direct viewers' attention to objects of interest, provide context, and help verify correctness. Although we have designed these cues with CT image analysis in mind, they can be used for any other DVR application.

4.1 Method

Our method consists of two main visual cues and their combinations either with each other or with already existing cues. We will first discuss Transfer Function Based Blurring (TFBB), then we will discuss Focus-Based Chroma-Coding (FBCC), and finally we will talk about their possible combinations.

4.1.1 Transfer Function Based Blurring

The challenge of creating DoF for real-time DVR has been tackled in recent years by works by Schott *et al.* [47] and the work presented in Chapter 3 of this thesis [51]. The method proposed in the previous chapter has a special property that we will exploit to create our new visual cue. Using our method it is possible to blur any portion of the volume at any depth and in any direction, which grants us the liberty of extending the work of Kosara *et al.* [18] on focus-context and semantic DoF to DVR.

Contrary to DoF and instead of relying on depth, focus distance, and lens properties to determine the blur level of an object, we have decoupled depth from blurring. In TFBB whether a chunk of volume appear blurred or not in the final image is determined based on each individual material. The role of a Transfer Function (TF) in DVR is to map voxel values to colors and opacities. This mapping combined with volume rendering integral then determines how to visualize color and density of different ranges of voxel values. TFBB augments TFs with the ability to dynamically control how much each material is blurred. Moreover, the amount of blur for each material can be controlled either by the user or by an automatic algorithm in real-time.

This method provides the viewer with the advantages of semantic depth of field for visual perception. In addition to rendering the objects of interest in focus, the blurred objects provide the necessary context and spatial reference to the viewer. It has been previously shown [18] that this is also an effective way to direct viewers' attention to the region of interest in an image.

Figure 4.1 shows arteries from a cardiovascular CT. The CT scan was performed on a patient who was injected with a contrast agent. The contrast agent absorbs x-ray similar to bone, so the contrast agent and the bone share

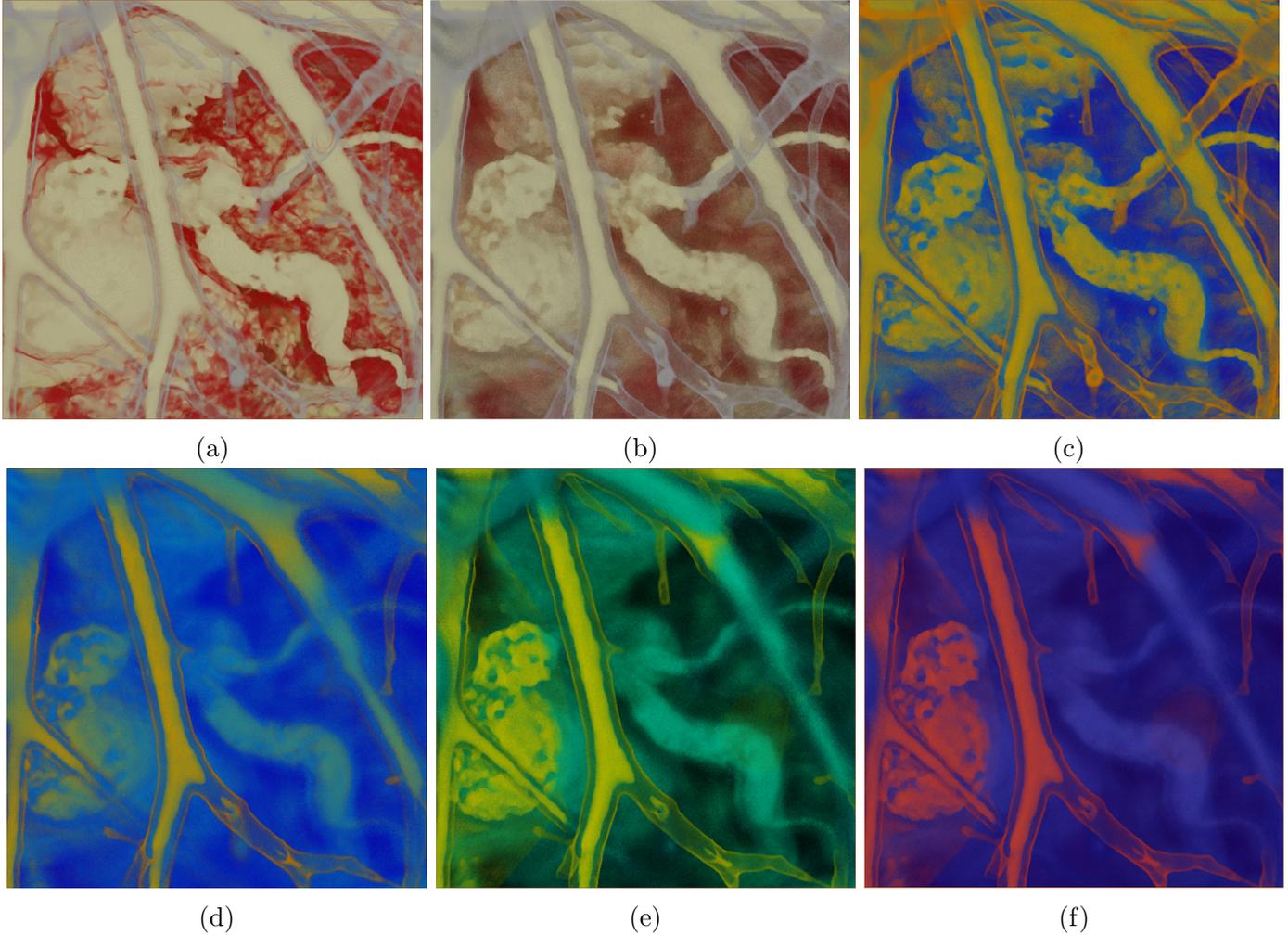


Figure 4.1: For all the images above there are 512^2 rays cast and the number of samples taken on each ray were 1350. All of these images show a similar view, and are rendered from a $512 \times 512 \times 308$ cardiovascular CT volumetric data. The patient has been injected with contrast agent. Figure 4.1a shows the volume rendered without TFBB, FBCC, or DoF at ≈ 13.61 FPS ($73.5ms/frame$). Figure 4.1b shows the scene rendered with TFBB at ≈ 8.56 FPS ($116.9ms/frame$) while having only the lumen and contrast agent density selected to be in focus. Figure 4.1c shows the same image with lumen and contrast agent in focus because of TFBB and uses FBCC gradient coloring with a spectrum between red and blue. It has been rendered at ≈ 7.78 FPS ($128.6ms/frame$). Figures 4.1d, 4.1e, and 4.1f use the combination of TFBB, DoF, and FBCC. Lumen and contrast agent densities are selected for focus in TFBB and the focus distance of DoF is chosen to have an artery with a blockage in focus. Figure 4.1d uses the gradient with red-blue spectrum for FBCC and is rendered at ≈ 3.86 FPS ($258.7ms/frame$). Figure 4.1e is rendered with the discrete color approach with colors yellow and cyan at ≈ 4.49 FPS ($223.0ms/frame$). Figure 4.1f is also rendered with the discrete color approach with colors red and blue at ≈ 4.03 FPS ($248.4ms/frame$).

the same range of values inside TF. In the TF used to render Figure 4.1b all different ranges of values except for the range of values corresponding to bone/contrast agent and the range of values corresponding to the blood vessel walls were assigned a non-zero blur level. Consequently, only the contrast agent and walls of blood vessels are in focus. Visually comparing plain rendering in Figure 4.1a with TFBB rendering of the same view in Figure 4.1b illustrates the effect of TFBB on partially removing visual clutter and revealing the objects of interest while maintaining the context in an effective way.

4.1.2 Focus-Based Chroma-coding

Pseudo-chromadepth is based on chromostereoscopic process proposed by Steenblich [54] and is used in some of the current commercial medical imaging software. In pseudo-chromadepth the goal is to help the viewer better perceive depth by shifting colors towards blue or red. The fact that different wavelengths of light are refracted at different angles by the lens of the eye is the reason behind perceiving shades of red closer than shades of blue [45].

Considering a DVR-rendered image that is using either DoF or TFBB scheme, there will be different regions of the volume that are rendered either as being blurred or as being in focus. It is safe to assume that in focus areas are the areas of interest and blurred areas are only providing context and spatial reference for the viewer. Focus-based chroma-coding is to apply chroma-coding during the rendering process based on in focus and out of focus areas. This enhances visual perception, directs viewer's attention to the object of interest, and provides a way to verify focus when there is a chance of mistaking small blur levels with an object being completely in focus.

Contrast and more specifically color contrast has already been established as an effective factor in visual perception [69]. In addition to red-blue color

scheme, it is possible to use other colors in TFBB that can enhance the color contrast such as yellow-cyan scheme, which are complementary colors. Furthermore, we have employed two different approaches to apply FBCC.

In the first approach a gradient of colors for different levels of blur is used. The second approach uses one color only for the in focus area while the other color is applied to any blurred area regardless of their level of blur. The former reflects the smooth or sudden transitions between blur levels, while the latter can be used for focus verification. As a result of our definition of FBCC, it can only produce the desired results while a blurring scheme such as DoF, TFBB, or both are enabled during the rendering.

4.1.3 Combinations

Since FBCC relies on focus, it should be used while a blurring scheme is in effect. It is possible to combine different color schemes with different blurring methods. TFBB and DoF can be used both individually or as a combination. In this chapter, we show the results of four different FBCC color combinations: normal coloring, gradient red-blue, discrete red-blue, and discrete yellow-cyan.

Normal coloring is when the colors in the transfer function are assigned so that they resemble the natural colors of the materials. In this mode the focus does not affect the colors. Gradient mode, as discussed before, is when each level of blur is colored based on its blur strength. The discrete modes are the ones with the first color only being applied to the focused region of the volume and the second color being applied to all blurred regions of the volume.

Figure 4.1a shows a view of the cardiovascular system rendered with normal coloring and no blurring applied (see Section 4.1.1 for more details). This is an example of the plain rendering of the volume. Figure 4.1c shows the volume rendered with gradient red-blue FBCC combined with only TFBB. It is worth

noting that the colors are shifted towards red and blue and are not replaced with red and blue. The shifting parameters can be easily tweaked to produce a stronger or a weaker shift towards red and blue.

Figures 4.1d, 4.1e, and 4.1f are all rendered using TFBB, DoF, and FBCC simultaneously. Figure 4.1d uses the gradient red-blue FBCC. Figure 4.1e uses the discrete yellow-cyan FBCC, and Figure 4.1f uses the discrete red-blue FBCC. A preliminary visual assessment of the images shows how the combination of the three methods can make the regions of interest much more distinguishable from their surroundings while maintaining the context and spatial references. In the case of images in Figure 4.1, an abnormal clogged artery where the contrast agent and blood do not flow is much more clearly visible in Figures 4.1d, 4.1e, and 4.1f. This abnormality is almost invisible in the plain rendering of the image in Figure 4.1a.

4.2 Implementation

In order to implement the proposed methods we used OpenGL and Cuda on a laptop using Nvidia GeForce GTX 680M and Intel Core i7 running at 2.4GHz. These methods can be easily implemented using CG, GLSL, or other platforms for GPU programming. We have used Cuda to better maintain and monitor using of GPU resources.

Our method has been implemented as an augmentation to DoF-enabled volume ray casting. The GPU ray casting technique proposed by Kruger *et al.* [23] has been used in conjunction with the DoF and rendering-time blurring method proposed in Chapter 3.

4.2.1 Implementation of TFBB

Although TFBB can augment any form of TF including but not limited to gradient-based, second order-partial derivatives and hessian matrix-based TFs, etc. we have chosen to demonstrate TFBB using a post-interpolative piecewise linear TF.

In a piecewise linear TF each material is defined by a range of corresponding voxel values. Each material is then assigned a color and opacity. Additionally, we have added a separate blur level value for each material. In this way it is also possible to change the blur level that is assigned to each material in real-time. In addition to updating the TF, we have updated the rendering algorithm in order to use the material blur information stored inside of the TF to produce TFBB. The pseudocode in Algorithm 4.1 demonstrates the changes that need to be made to a normal ray casting algorithm in order to enable it to take advantage of TFBB and DoF both in individual or combined forms.

4.2.2 Implementation of FBCC

FBCC can be added to the ray casting at virtually no additional cost. The time and place that we decided to add the FBCC step during rendering is just before integrating the color and opacity of the current sample with the corresponding pixel color. At this point both the blur level and the TF-based color of the sample are known. Knowing the blur level is required since FBCC alters the color of a sample based on its focus state.

The pseudocode in Algorithm 4.2 shows how to simply add FBCC to the ray casting loop. In this pseudocode it is assumed that each color c consists of 3 channels (r, g, b) where the value of each channel is in range $[0, 255]$. The blur level is in range $[1, 29]$ and the coefficient seen in lines 4 and 6 of the

Algorithm 4.1 TFBB and DoF enabled volume ray casting.

Input: Rendering mode $rMode$, depth of current sample $depth$, camera settings $camParams$, and current sampling coordinate (x, y, z) .

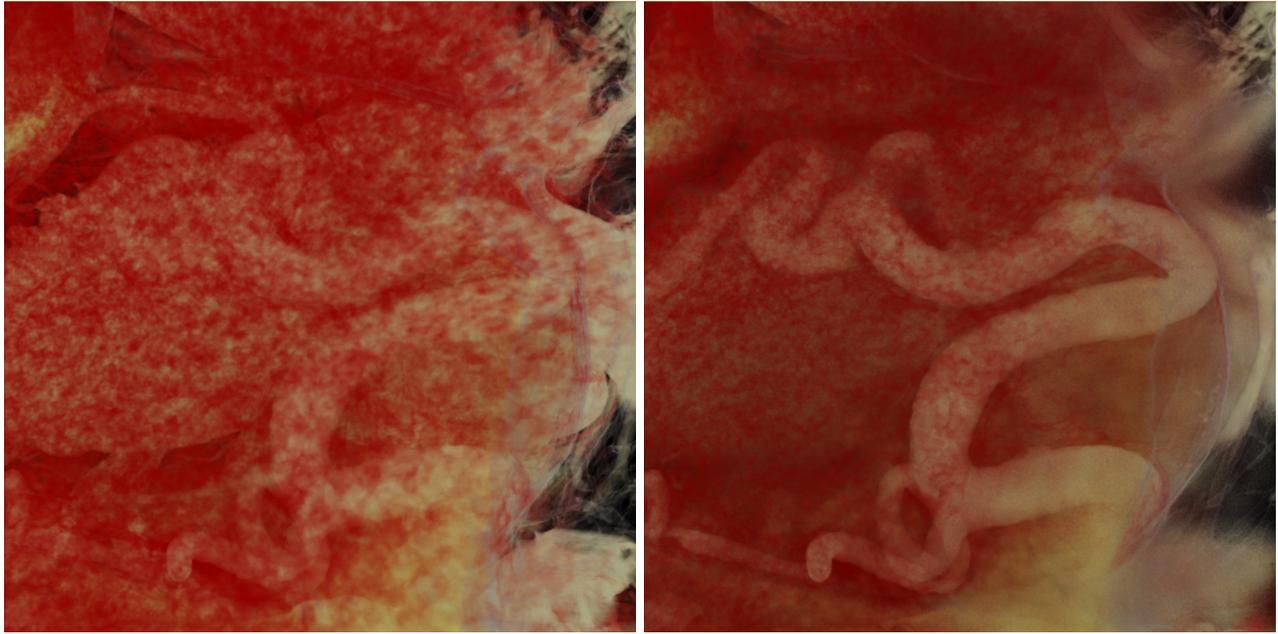
Output: $blurLevel$ based on DoF, TFBB, DoF+TFBB, or no effect.

```
for  $t = 0 \rightarrow \infty$  do                                \\ Loop for casting a single ray
  :
   $blurLevel \leftarrow 1$                              \\ Ray casting initializations
  \\ In focus
  if ( $rMode = DoF$ ) then
     $blurLevel \leftarrow DoFCalc(depth, camParams)$ 
  else
    if ( $rMode = TFBB$ ) then
       $vSample \leftarrow SampleVolume(x, y, z)$ 
       $blurLevel \leftarrow GetBlurLevelFromTF(vSample)$ 
    else
      if ( $rMode = DoF + TFBB$ ) then
         $vSample \leftarrow SampleVolume(x, y, z)$ 
         $blurLevel \leftarrow GetBlurLevelFromTF(vSample)$ 
        if ( $blurLevel = 1$ ) then                       \\ not blurred by TFBB
           $blurLevel \leftarrow DoFCalc(depth, camParams)$ 
        end if
      end if
    end if
  end if
end if
  :
end for                                             \\ Other ray casting tasks
```

Algorithm 4.2 with the value equal to 8.8 is the result of dividing range of values of a channel, 256, by range of blur levels, 29, to create a gradient coefficient.

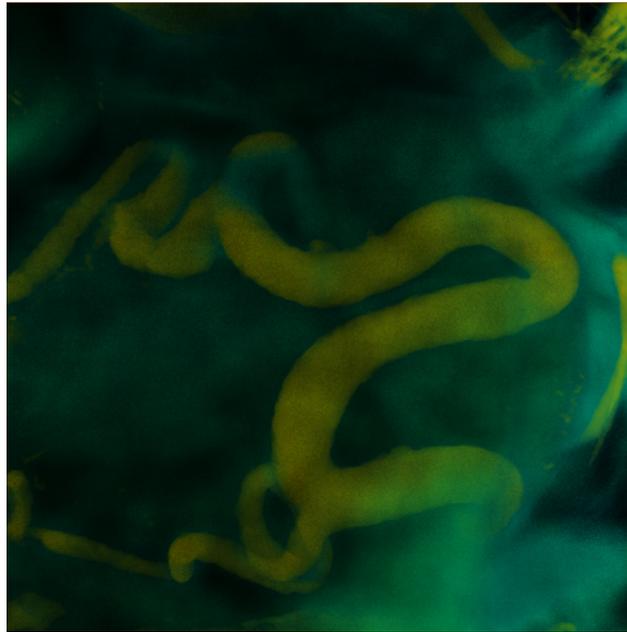
4.3 Results

The goal of the proposed methods is to enhance visualization by reducing clutter, directing viewer's attention, and enhancing correct diagnosis in DVR rendered medical images. Figure 4.1 was discussed in Section 4.1.1 and we will continue discussing the rest of the results of this chapter here. Figures 4.2, 4.3, 4.4, and 4.5 provide different sets of images. Each image in a set of images is taken from the same view as the rest of images in that set. However, each image in a set is rendered differently for the sake of comparison. These images are all rendered from volumetric data from CT scanning of patients injected



(a)

(b)



(c)

Figure 4.2: For all the images above there are 512^2 rays cast and the number of samples taken on each ray were 1350. All images are rendered from a $512 \times 512 \times 308$ cardiovascular CT volumetric data. The patient has been injected with contrast agent. Figure 4.2a shows the scene rendered without TFBB, FBCC, or DoF at ≈ 8.46 FPS ($118.2ms/frame$). Figure 4.2b shows the scene rendered with DoF at ≈ 4.25 FPS ($235.1ms/frame$). Figure 4.2c shows the same scene rendered at ≈ 2.69 FPS ($372.4ms/frame$) using DoF, TFBB with lumen and contrast agent in focus, and FBCC with discrete yellow-cyan coloring.

Algorithm 4.2 FBCC-enabled volume ray casting.

Input: Blur level of the current sample $\mathbf{blurLevel} \in [1, 29]$, current sample color $(\mathbf{r}, \mathbf{g}, \mathbf{b})$, chroma strength $\mathbf{cStrength} \in (0, 5]$, and chroma-coding mode selector \mathbf{cMode} .

Output: Altered sample color $(\mathbf{r}, \mathbf{g}, \mathbf{b})$.

```
1: for  $t = 0 \rightarrow \infty$  do                                     \\ Loop for casting a single ray
2:    $\vdots$                                                  \\ Ray casting initializations, sampling, and blurring
3:   if ( $\mathbf{cMode} = \textit{Gradient Red Blue}$ ) then
4:      $\mathbf{r} \leftarrow (30 - \mathbf{blurLevel}) \times 8.8$ 
5:      $\mathbf{g} \leftarrow \mathbf{g}/\mathbf{cStrength}$                  \\ use ( $\mathbf{cStrength} \times 2$ ) for deeper reds.
6:      $\mathbf{b} \leftarrow \mathbf{blurLevel} \times 8.8$ 
7:   else
8:     if ( $\mathbf{cMode} = \textit{Discrete Red Blue}$ ) then
9:       if ( $\mathbf{blurLevel} = 1$ ) then
10:         $\mathbf{r} \leftarrow (\mathbf{r} + 256)/2$ 
11:         $\mathbf{g} \leftarrow \mathbf{g}/(\mathbf{cStrength} \times 2)$ 
12:         $\mathbf{b} \leftarrow \mathbf{b}/(\mathbf{cStrength} \times 2)$ 
13:      else
14:         $\mathbf{r} \leftarrow \mathbf{r}/(\mathbf{cStrength} \times 2)$ 
15:         $\mathbf{g} \leftarrow \mathbf{g}/(\mathbf{cStrength} \times 2)$ 
16:         $\mathbf{b} \leftarrow (\mathbf{b} + 256)/2$ 
17:      end if
18:    else
19:      if ( $\mathbf{cMode} = \textit{Discrete Yellow Cyan}$ ) then
20:        if ( $\mathbf{blurLevel} = 1$ ) then
21:           $\mathbf{b} \leftarrow 0$                                \\  $\mathbf{r}$  and  $\mathbf{g}$  unchanged.
22:        else
23:           $\mathbf{r} \leftarrow 0$                                \\  $\mathbf{g}$  and  $\mathbf{b}$  unchanged.
24:        end if
25:      end if
26:    end if
27:  end if
28:   $\vdots$                                                  \\ final integration and other ray casting tasks
29: end for
```

with contrast agent.

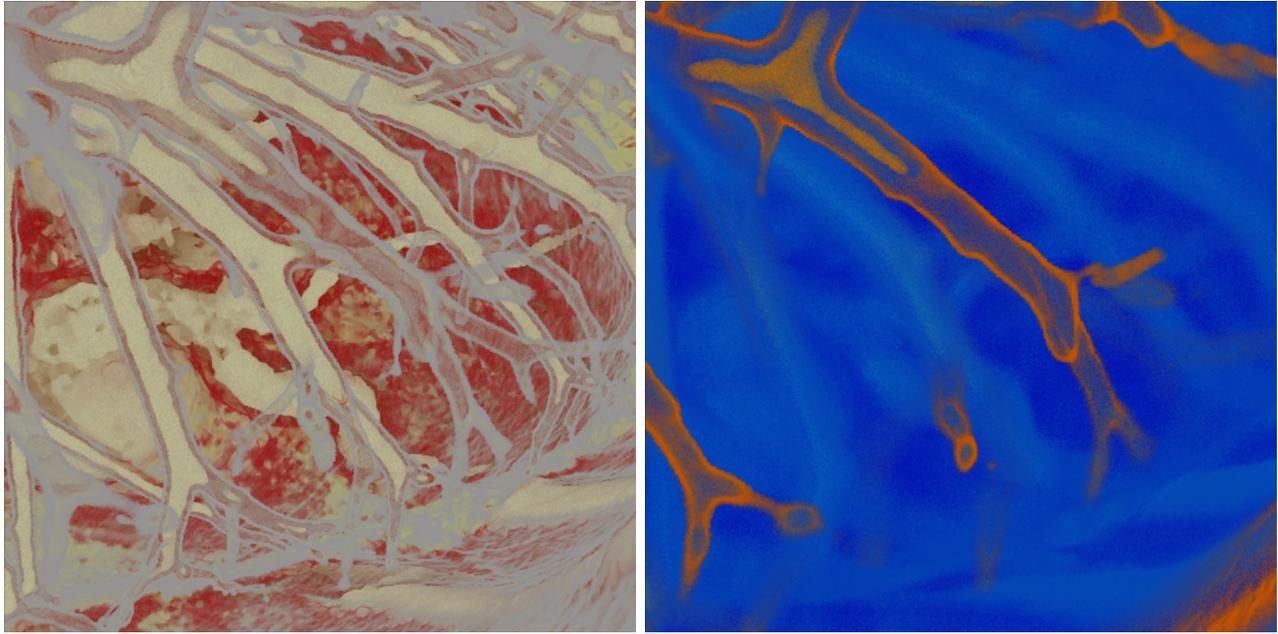
Figure 4.1 provides a comparison between different techniques that can be employed to render a scene. The blocked artery can be easily missed if the viewer only relies on Figure 4.1a, which we consider the reference. The reference is normal ray casting without our visual enhancements. Colors used to render reference are designed to resemble the natural colors of materials as much as possible. Figure 4.1b is rendered while using TFBB with lumen and contrast agent (bone) selected to be in focus. It can be seen that the visibility of the arteries and lumen are enhanced. FBCC is enabled in Figures 4.1c while

using TFBB. Figures 4.1d, 4.1e and 4.1f illuminate the difference that was made in visibility of the defect while combining TFBB and FBCC with DoF. The combination of the three methods provides the more precise view of the object of interest, which in this case is the artery. The choice of FBCC method depends on the viewer and the scene. A color scheme might work better in one scene compared to another. This can be seen in the bottom row of images in Figure 4.1.

Figure 4.2 shows results from 3D rendering of a cardiovascular CT using our implementation. The huge blood vessel at the bottom of the heart is almost completely concealed behind smaller blood vessels and soft tissue while rendered with normal ray casting and conventional colors (reference) as shown in Figure 4.2a. A shallow DoF has been used in Figure 4.2b to reduce the clutter occluding a direct view of the blood vessel. However, it is only partially successful in providing an unobstructed view. By using a combination of DoF, TFBB, and FBCC methods in Figure 4.2c the remaining clutter has been removed and a clear view of the blood vessel can be used by physicians for fast and accurate diagnosis.

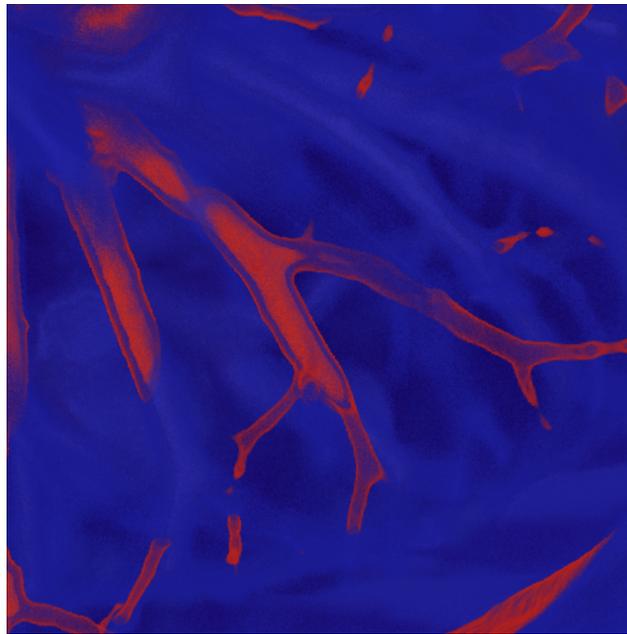
Figure 4.3a shows the reference rendering results from a group of cardiovascular and pulmonary arteries from a cardiovascular CT using our implementation. Because of the number of blood vessels rendered in this view, diagnosis will be time consuming and prone to errors. Both Figures 4.3b and 4.3c use DoF, TFBB, and FBCC on the same view to isolate arteries in which the blood flow is restricted. Using our methods, it is simply possible to isolate arteries to assess them one at a time with less occlusion, and lesser chance of error.

Figure 4.4 shows 3 images rendered from a chest CT scan of a patient injected with contrast agent. Although this patient suffers from aortic dissection, which is visible in Figures 4.4b and 4.4c, there is almost no abnormality



(a)

(b)



(c)

Figure 4.3: For all the images above there are 512^2 rays cast and the number of samples taken on each ray were 1350. Each row of the images are taken from a different view, and are rendered from a $512 \times 512 \times 308$ cardiovascular CT volumetric data. The patient has been injected with contrast agent. Figure 4.3a is rendered in full focus with normal coloring at ≈ 13.12 FPS ($76.2ms/frame$). Figure 4.3b is rendered with the gradient blue-red color approach using TFBB, DoF, and FBCC at ≈ 3.78 FPS ($264.3ms/frame$). Figure 4.3c is rendered with the discrete blue-red color approach using TFBB, DoF, and FBCC at ≈ 2.96 FPS ($337.6ms/frame$).

visible in figure 4.4a. Correct clinical diagnosis is critical in such cases where the patient needs to be treated very fast.

Figure 4.4a is the reference and does not take advantage of our visual cues. Figure 4.4b is taking advantage of TFBB which makes the abnormality stand out and be easier to detect. Figure 4.4c is rendered with both TFBB and FBCC which makes it virtually impossible to miss the aortic dissection. Moreover, the spatial orientation as well as the location of the abnormality in relation to its surroundings are more clear.

Figure 4.5 shows a set of 4 images rendered from CT scan of the cardiovascular system of a patient injected with contrast agent. All these images are rendered from the same location with the same orientation of the camera. The camera is pointed at a coronary artery with severe stenosis. The reference rendering of the image, Figure 4.5a, does not reveal any abnormality. The reason is that a lot of smaller blood vessels and soft tissue are blocking the observer from viewing the abnormal artery. Paying close attention to the differences in rendering speeds of the four images reveal that all the rays cast in Figure 4.5a are cut short really fast by early ray termination compared to the other three. The reason for this is the thick layer of blood vessels and soft tissue. As a result, the higher rendering speed of reference is actually not to our benefit.

Figure 4.5b is rendered using the combination of depth of field and FBCC. Although the coronary artery in focus is revealed much better than the reference, it is still very hard to examine the artery for abnormalities since little details are hidden under many smaller blood vessels. Figure 4.5c uses TFBB instead of depth of field. The combination of TFBB and FBCC clears the image off the clutter and unwanted details. It reveals the artery in a way that the diagnosis is less prone to cause errors.

Finally, Figure 4.5d takes advantage of TFBB, depth of field, and FBCC all at the same time. It is important to note that in our implementation

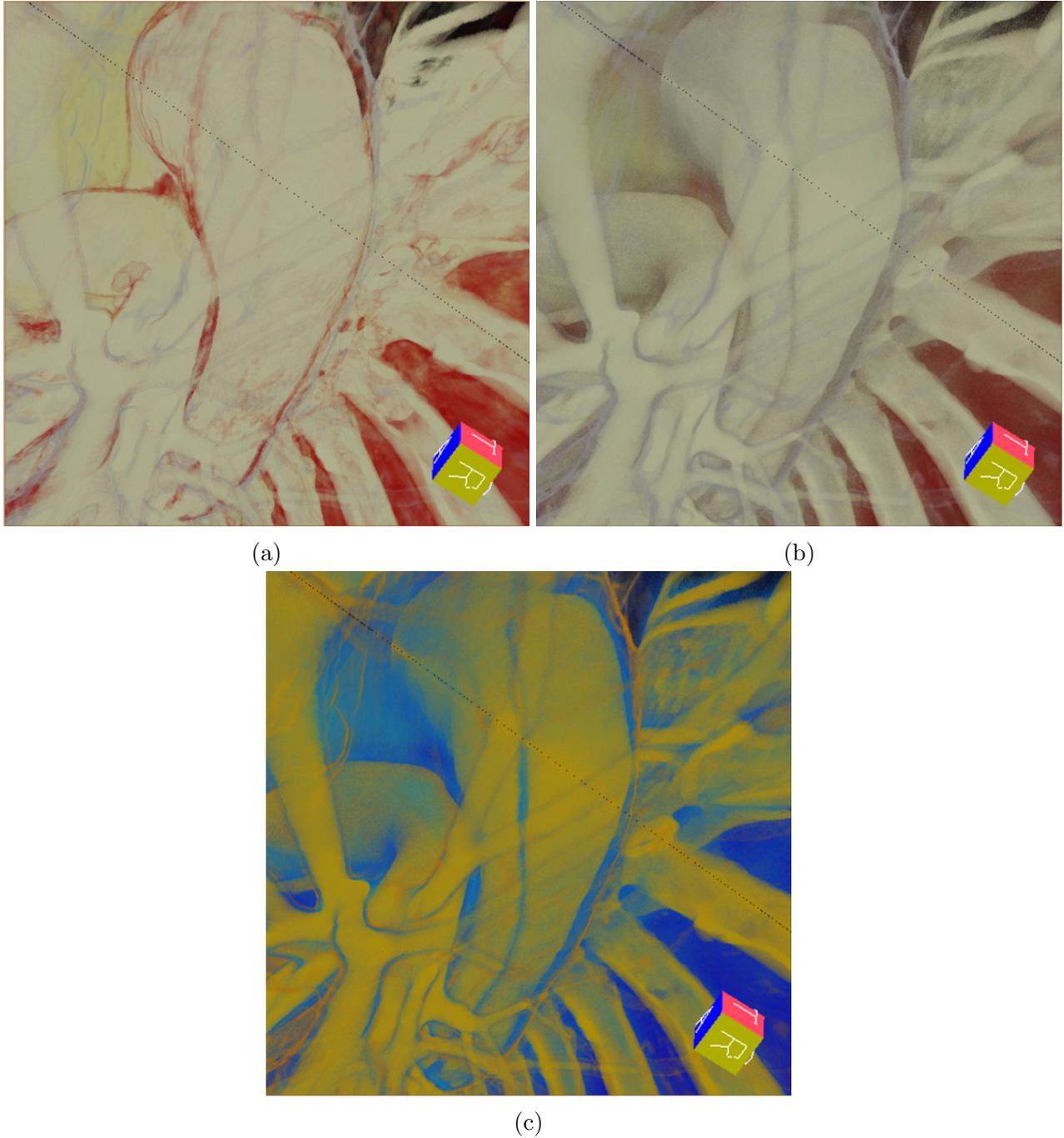


Figure 4.4: For all the images above there are 512^2 rays cast and 1350 samples were taken on each ray. All images are taken from the same view, and are rendered from a $512 \times 512 \times 386$ chest CT volumetric data. The patient has been injected with contrast agent. Figure 4.4a shows the scene rendered without TFBB, FBCC, or DoF at ≈ 16.53 FPS ($60.5ms/frame$). Figure 4.4b shows the scene rendered with only TFBB with lumen and contrast agent in focus at ≈ 11.00 FPS ($90.9ms/frame$). Figure 4.4c shows the same scene rendered at ≈ 10.09 FPS ($99.1ms/frame$) using TFBB with lumen and contrast agent in focus, and FBCC with gradient red-blue coloring. The abnormality (aortic dissection) can be easily missed with normal rendering.

depth of field blur overrides TFBB when used simultaneously. That is, if a part of image is supposed to be out of focus because of depth of field then it will be out of focus regardless of TFBB. The mechanism for this has been shown in Algorithm 4.1. Depth of field examines everything that is marked as in-focus by TFBB. This will blur all that fall out of depth of field's focus range. Consequently, the combination of the two provides a depth-limited and uncluttered focus on the objects of interest as can be seen in Figure 4.5d. When TFBB is being applied without depth of field the image is considered in focus throughout all its depth. In this case TFBB controls all the decisions regarding blurring or not blurring of all material.

The rendering speed when using TFBB, DoF, and TFCC all at the same time is approximately between 60% and 75% of rendering speed when the volume is rendered using only DoF. This means that interactivity is maintained while using the aforementioned methods. The reason for the frame rate drop in TFBB is that we need to take one extra sample from the volume during the ray casting loop in order to read the blur level from TF. This sample can later be used instead of one of the required samples if we need a sample on the ray itself. The results presented in this chapter do not take advantage of reusing the extra sample. Moreover, only early ray termination has been used as a means of boosting performance and other speed enhancement methods are not used.

4.4 Conclusion

In this chapter, we have proposed two methods to improve ray casting algorithm for DVR. These methods are aimed at reducing clutter, enhancing visual perception, and directing viewer's attention. Both of these methods rely on the fact that the volume can be blurred in real-time [51] as described in Chap-

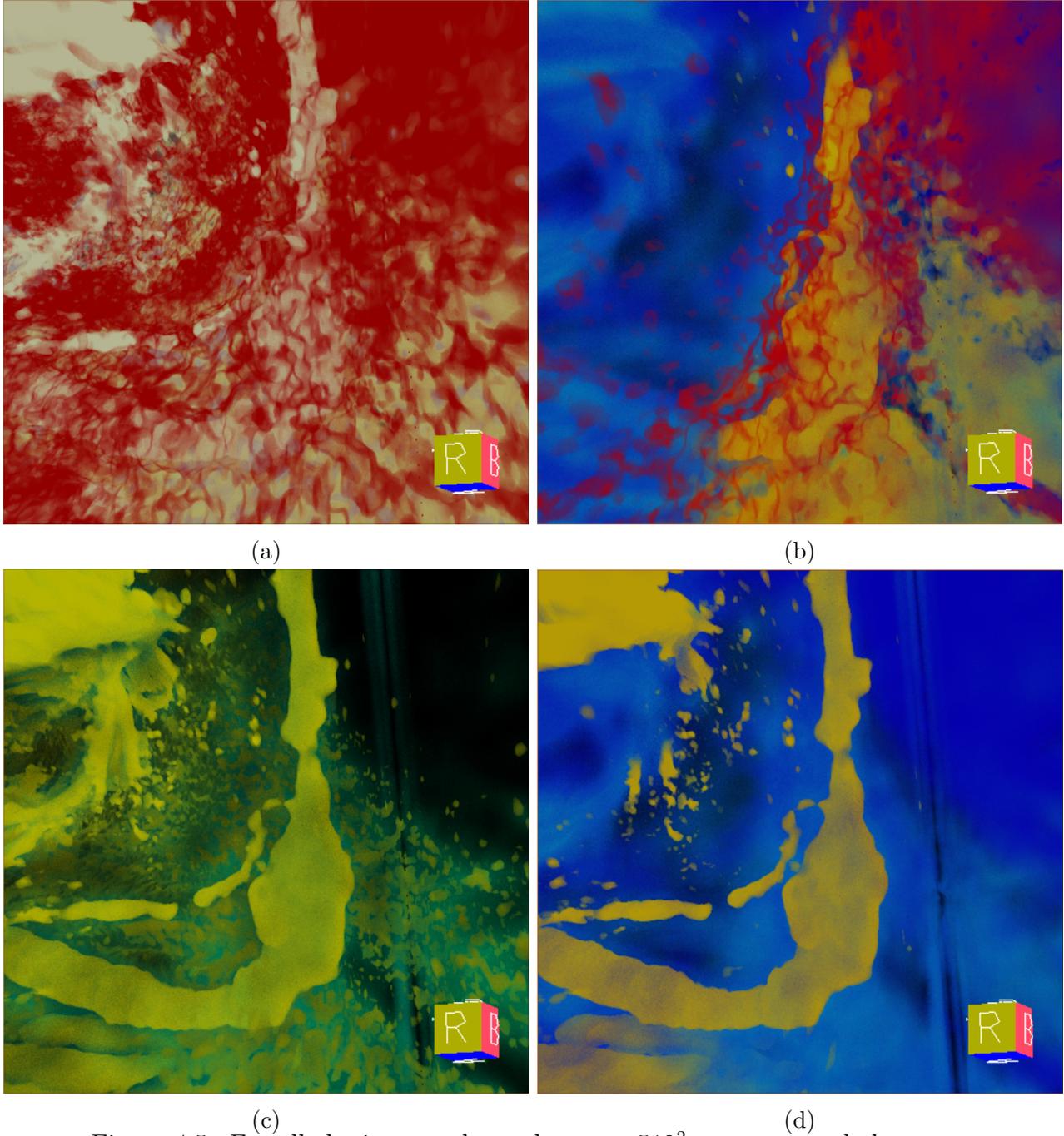


Figure 4.5: For all the images above there are 512^2 rays cast and the number of samples taken on each ray were 1370. All images are taken from the same view, and are rendered from a $512 \times 512 \times 299$ cardiovascular CT volumetric data. The patient has been injected with contrast agent. Figure 4.5a shows the scene rendered without TFBB, FBCC, or DoF at ≈ 20.79 FPS ($48.2ms/frame$). Figure 4.5b shows the scene rendered with DoF and FBCC with gradient red-blue coloring at ≈ 5.60 FPS ($178.5ms/frame$). Figure 4.5c shows the same scene rendered at ≈ 5.10 FPS ($196.1ms/frame$) using TFBB with lumen and contrast agent in focus, and FBCC with discrete yellow-cyan coloring. Figure 4.5d is rendered with the gradient red-blue color approach using TFBB, DoF, and FBCC at ≈ 4.56 FPS ($219.2ms/frame$).

ter 3. Our results show that a physician can use our proposed methods even in the most complex scenes to remove the unnecessary clutter and provide an unobstructed view of the object of interest.

Since our methods are interactive, it is also possible to take advantage of motion parallax while using our methods. As discussed, in Chapter 2 many visual cues work best if combined with other visual cues. However, these combinations need to be studied through usability studies to be validated.

Basically any visualization that already uses DVR can benefit from the proposed methods. Our main focus has been on medical imaging and medical visualization, however, visualization of other 3D sampled data sets or scalar fields can also benefit from our proposed methods.

Chapter 5

Usability Study

No scientific endeavor is complete without evaluation. In previous chapters we have proposed multiple new visualization enhancement techniques and algorithms. The effectiveness of some of the proposed techniques such as Depth-of-Field on human visual perception has been vastly studied before as discussed in Chapter 2. However, the effect of some of the newly proposed methods in Chapter 4 on human visual perception and more specifically on imaging-aided clinical diagnosis has not been studied before. In this chapter, we will explain the usability study that we have designed and conducted to test the effectiveness of our methods [51, 52] for clinical decision making.

5.1 Evaluation Targets

In Chapter 4 of this dissertation, we introduced Transfer Function Based Blurring (TFBB), Focus-Based Chroma-Coding (FBCC), and their various combinations as new visualization techniques to augment current volume rendering methods. The primary goal of these methods is to improve clinical decision making by enhancing accuracy, and decreasing chances of wrong diagnosis. The secondary goals of these methods are enhancing the time required for clinical diagnosis, quality of diagnosis, as well as reducing the training time of physicians. In our usability study, we have tried to address as many ques-

tions as possible within the boundaries and limits set by our access to human resources. We will discuss the limitations of the study later.

Among all the possible combinations of the visual enhancement methods we have chosen 6 to be compared to the reference. The reason for excluding some variations is to reduce the number of images that the participants needs to view as well as the time they needed to do the usability study. Table 5.1 shows the visualization methods that were chosen to participate in our usability study and their code names.

Table 5.1: Participating Methods in the Usability Study

Focus Mode →	Full Focus (FF)	TFBB	TFBB + DoF
↓ Color Mode			
No Chroma-Coding (NCC)	✓	✓	✓
Gradient Red Focus Blue Out of Focus (RFBO+)		✓	✓
Binary Red Focus Blue Out of Focus (RFBO)			
Binary Yellow Focus Cyan Out of Focus (YFCO)		✓	✓

In the rest of this chapter, we are going to mention our different visualization methods many times. In order to avoid confusion we are going to refer to each visualization method with a tuple. For example, $\langle TFBB, DoF, NCC \rangle$ means using transfer function based blurring (TFBB), depth-of-field, and no chroma-coding. All the code names for visualization categories used in this notation as well as the labels on the axes of figures can be found in Table 5.1. The reference, $\langle FF, NCC \rangle$, is defined as DVR rendered images where there is no blurring and no special chroma-coding applied. NCC means the colors

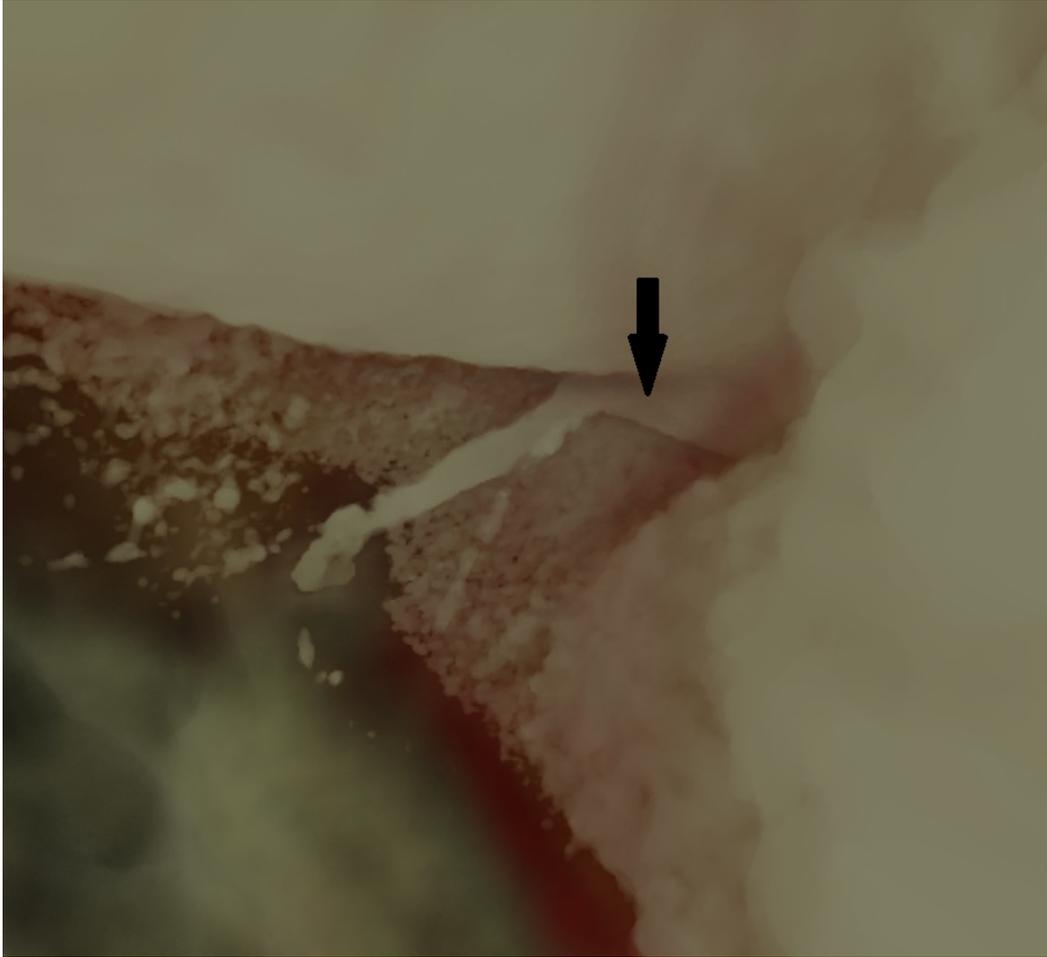


Figure 5.1: In some cases an arrow was provided to indicate the artery of interest in the question.

used are the ones generally perceived as the natural colors of the tissue, blood, muscles, and bones.

The null hypothesis of this study is that there is no meaningful difference between the reference and the other 6 categories of images in terms of assessments made by the test takers.

5.2 Study Design

Our study evaluates the differences between the seven participating visualization methods from four different aspects. These aspects are detection of an abnormality, marking of an abnormal region, time to detect and mark the

abnormal region, and distance of the marked location to the center of the abnormal region.

5.2.1 Data Extraction

The data for the study was extracted by an expert radiologist from 20 anonymized patient data. As a result of 7 categories of images and 20 patients we extracted 140 images. For each one of the visualization categories one image from each patient data has been captured. During extraction, we made sure that each captured image is from a different angle and view so it does not resemble other images captured from the same patient.

Each image was then marked by the expert radiologist as whether an abnormality is visible within the image or not. We made sure images with abnormality exhibit only a single abnormality and not multiple ones. In some cases, an arrow was provided in the image to indicate the blood vessel in question. An example of this is shown in Figure 5.1.

In order to eliminate bias among different visualization methods the order of extracting images for the 7 categories was changed for each patient. In addition to the expert radiologist assessing the images, the previous clinical diagnosis of the patients were also available and provided additional confirmation of the abnormality in the extracted images.

5.2.1.1 Primary & Secondary Questions

There are two questions associated with each image. These questions were also designed and written by the expert radiologist. The primary question of the image asks whether a particular abnormality is visible in the image or not. The correct answer to the primary question is either yes or no depending on the visibility of the abnormality. A few examples of the primary questions are listed here.

- *The structure marked with an arrow is a coronary artery. Is the coronary artery abnormal?*
- *We are looking from above. Is there a filling defect in this visualization of a pulmonary artery?*
- *Look at the aorta. Is there an abnormality?*
- *Is there a stenosis in this coronary artery?*
- *In the pulmonary artery branch indicated by the arrow, is there a thrombus?*

The secondary questions ask the tester to indicate the location of abnormality with a single left click of the mouse. These questions are only presented to the tester if the answer to the primary question was positive. In case of a negative answer to the primary question, which means the test taker evaluated the image as a normal one the test records the result of the current image and proceeds to the next image in the test. However, all images, even the ones without abnormality, had a secondary question assigned to them in case the test taker mistakenly indicated that an abnormality is present in a normal image. A few examples of the secondary question are listed here.

- *Left click on the center of the most severe abnormality.*
- *Mark the center of the most proximal aspect of the clot.*
- *Mark the centre of the clot.*
- *Left click on the center of the largest diameter of the stenotic vessel.*
- *Left click on the center of most proximal stenosis in the branch of this coronary artery.*

5.2.1.2 Abnormality Marking and Error Threshold

For each image with abnormality a single point of interest is marked by the expert radiologist as the location of the abnormality. The secondary questions

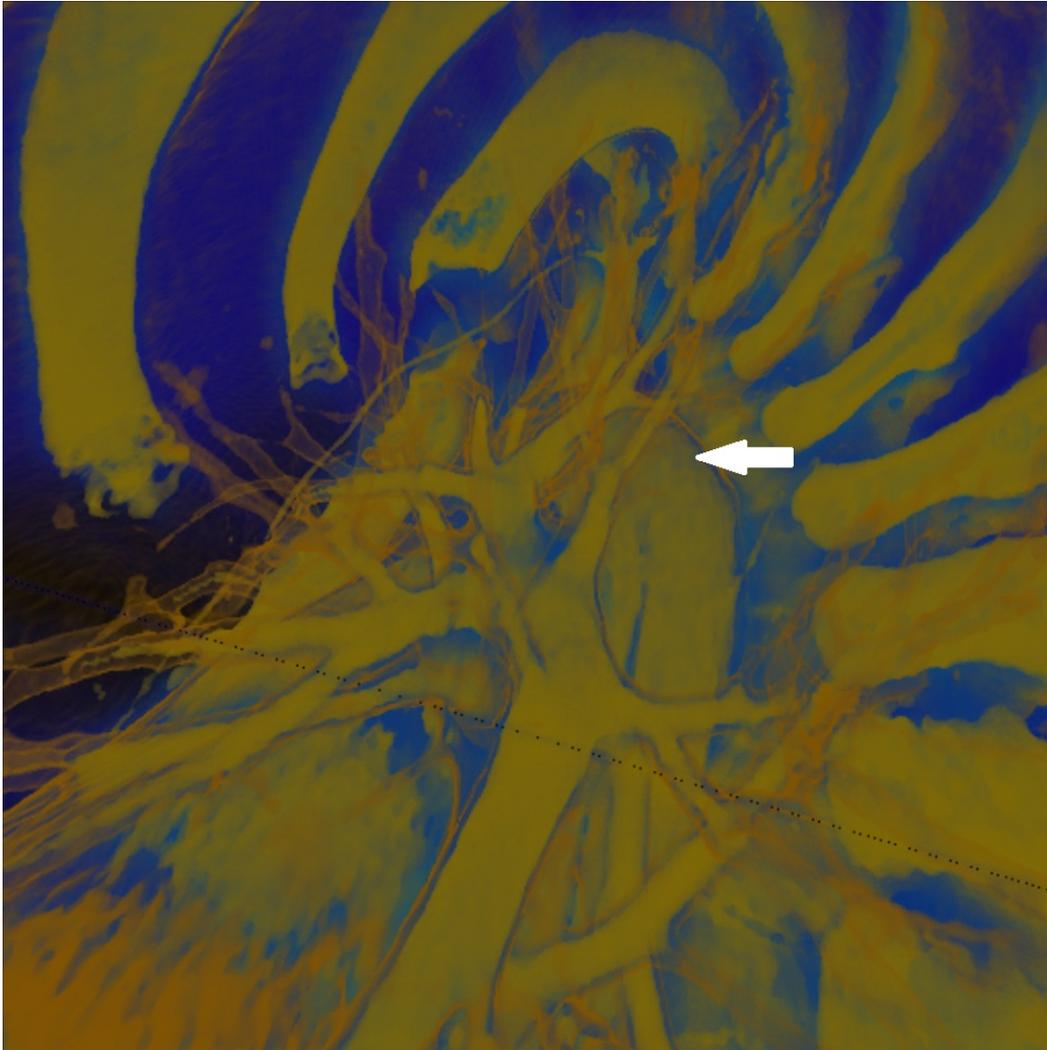


Figure 5.2: Example of an image that the visible abnormality is wider than a single point. In this case the arrow is pointing to aorta and the vertical blue line on the bottom left of the arrow is an aortic abnormality.

are designed to make sure the tester will not be confused when the abnormality is covering an area wider than a single point.

For example, in Figure 5.2 the test program asks the tester to “*Left click on the most proximal part of the abnormality*”. In this case the aorta of the patient is marked with an arrow. The abnormality is seen as a vertical blue line on the bottom right of the arrow. Since the abnormality is wider than a single point, the secondary question is asking for the most proximal part of the abnormality.

For each abnormality the expert radiologist assigned an error threshold as well. This error threshold is the diameter of a circle centered at the marked location of the abnormality. Clicking within this threshold circle by the tester is considered as correctly marking the abnormality. If the tester clicks outside of this threshold circle the marking result will be recorded as wrong marking. The reason for having this threshold is to take into consideration the distance of the abnormality to the camera, zooming or widening of the lens of camera, and the size and type of abnormality. This threshold was visually presented to the radiologist in the image editing program as seen in Figure 5.3 and could be edited there. This threshold circle was not presented to the tester and the testers were not aware of the existence of such threshold.

5.2.1.3 Time

The test as a whole is not timed. However, for each image a total of 15s seconds was assigned by the expert radiologist. This time is used to detect the abnormality in the image as well as marking it.

Two cases of time out are possible in the test. If the time runs out when the tester is deciding whether or not an abnormality is present in the image the result is recorded as wrong assessment and wrong marking. If the time runs out after the tester finished answering the primary question the answer to the primary question is recorded normally but the answer to the secondary question is recorded as wrong marking.

5.2.2 Usability Study Steps

There are 140 images in the test. These images are presented to the tester one at a time in a random order to eliminate biases.

In the first step for a new image the primary question is presented to the tester. At this time the image is hidden and the timer is not started. The

The screenshot displays a software interface for image editing. On the left, a medical scan image is shown with a purple circle and a white arrow pointing to a specific area. On the right, a list of rendering parameters is visible:

- Name: 20150515123102.ssi
- Rendering Mode: TF Based
- Sampling Rate: 1350
- Illumination: False
- Sampling Model: 1
- Permutation Sampling: True
- Chroma Coding Mode: Red-Focus Blue-Out of Focus +
- Blur Power: 15
- Focus Distance To Camera: 0.145
- Half in Focus Range: 0.06500001
- Material Selector Value: 35
- Chroma Strength: 100
- Solidity Coefficient: 31
- Defect Exists: True
- Defect Coordinates X: 594
- Defect Coordinates Y: 498
- Rotation X: 246.2995
- Rotation Y: 15.30028
- Rotation Z: 249.5999
- Field of View: 96

At the bottom of the interface, there is a control panel with the following elements:

- Buttons: Undo, Marked as having abnormality (checked), Open in Pair, Remove Image
- Field: Diameter of error tolerance circle (value: 82.0)
- Text: Question of the image: The aorta is indicated by an arrow. Is there an aortic abnormality?
- Text: Primary Question: The aorta is indicated by an arrow. Is there an aortic abnormality?
- Text: Secondary Question: Left click on the most proximal part of the abnormality
- Buttons: Temporary Save & Show Previous, Temporary Save & Show Next, Temporary Save & Close

Figure 5.3: Example of an image, in which the visible abnormality is wider than a single point, is seen in the image editor. The purple circle is the error threshold and the cyan point is the location of the abnormality chosen by the expert radiologist. In this case the arrow is pointing to aorta and the vertical blue line on the bottom left of the arrow is an aortic abnormality. The window seen here is the image editor window in the usability study program. This window can edit properties of an image such as primary question, secondary question, and error threshold.

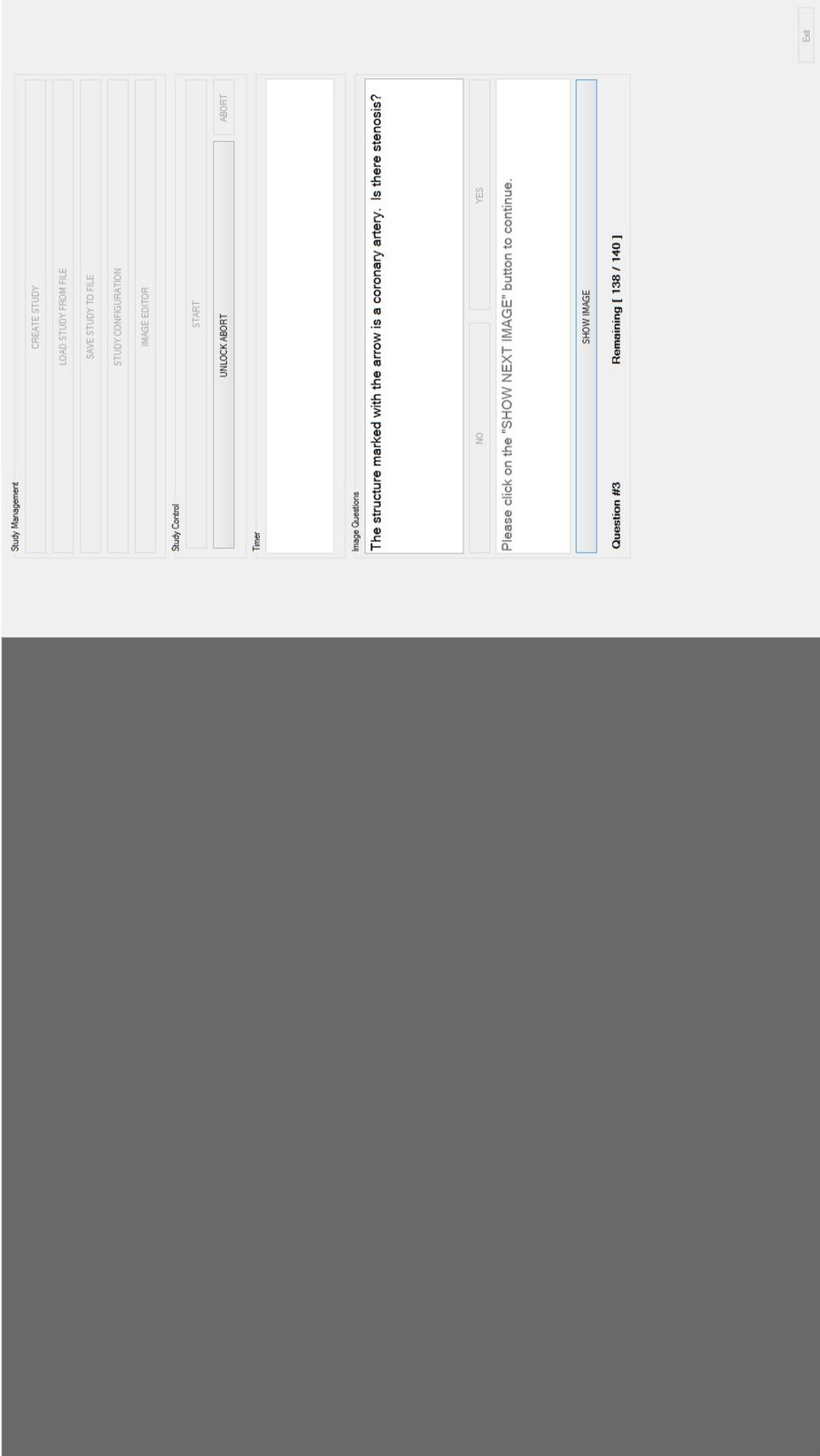


Figure 5.4: Usability study interface presenting the tester with primary question. At this time the image is hidden and the timer is not started.

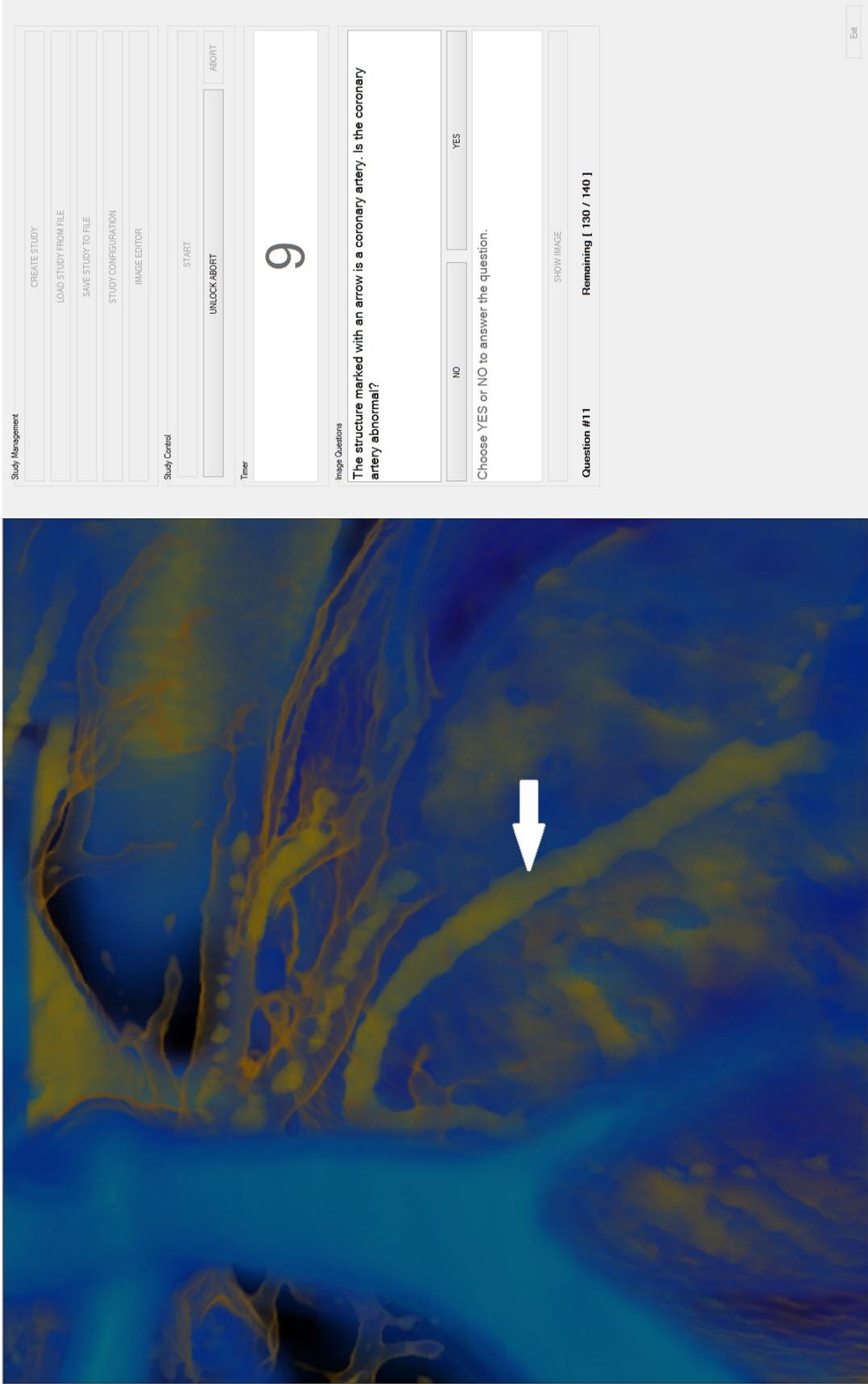


Figure 5.5: Usability study interface presenting the tester with primary question after pressing the *Show Image* button. At this time the image is shown and the timer is running.

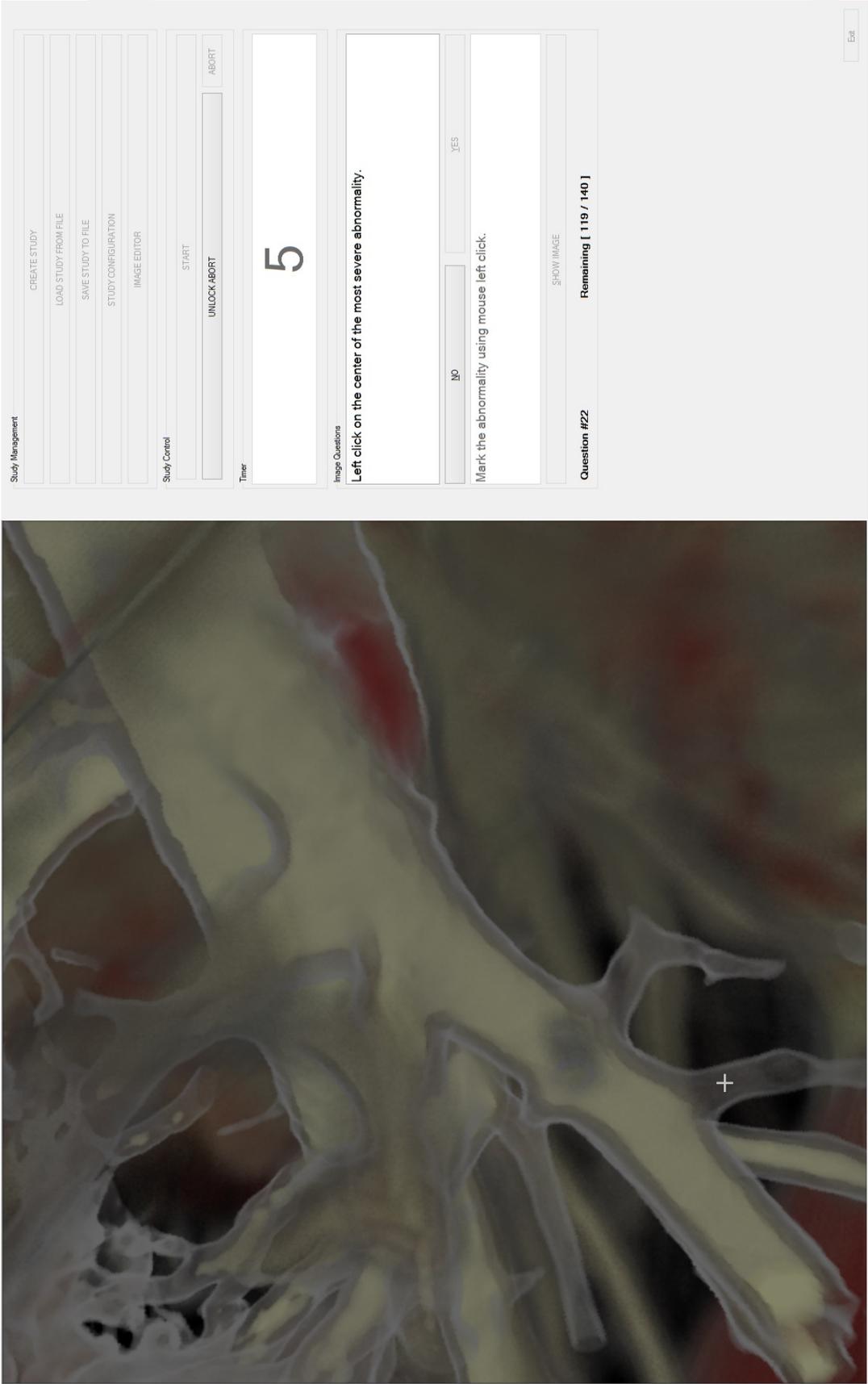


Figure 5.6: Usability study interface presenting the tester with secondary question after answering positively to the primary question. At this time the usability study program is waiting for one of the following: marking of the abnormality, running out of time, or changing primary question answer to negative. The mouse cursor also changes to plus (+) for marking of the abnormality.

tester can take as much time as desired to read the primary question or rest. Figure 5.4 shows the interface of the usability study program in this first step.

By pressing the *Show Image* button the image is shown and the timer starts counting down. At this time the tester can answer either yes or no to the primary question as shown in Figure 5.5. If the tester's answer to the primary question is negative the usability study program records the answer and proceeds to the first step for the next image. However, if the answer to the primary question is positive then the usability study program presents the tester with the secondary question and wait for one of the following three events to happen to proceed to the first step of the next image: marking the defect, running out of time, or changing the answer to the primary question to negative. This last step is shown in Figure 5.6.

After all of the images have been shown the results are saved. If the test is aborted during the test, no data will be saved and all the results of the current test will be erased.

5.2.3 Volunteers

In order to evaluate our methods using this usability study we needed volunteers who are trained in radiology. As a result, the pool to get volunteers from was extremely limited. Our volunteers were mostly radiology residents and a few practicing radiologists. One of the biases that is against us in this study is the differences in knowledge among our volunteers. We would have preferred to only have practicing radiologists or at least last year residents to take the test, however the limited number of qualified and willing volunteers forced us to take residents from all years. Of course volunteers who are residents of early years of radiology are less knowledgeable, less experienced, and consequently less consistent in their results in comparison to the more senior residents or radiologists.

Prior to a volunteer's participation in the study, he/she was presented with a consent form to read and sign. This consent form can be seen in Appendix A. There were a total of 11 testers in our study.

5.2.4 Factors Against Our Usability Study Results

In order to better understand the results of this study, we have to consider all the factors that played a role in these results.

5.2.4.1 Lack of Motion Parallax

The first factor that could help our results but was not present in the usability study was motion parallax. One of the important aspects of our algorithms is that they are real-time and interactive. However, we worked with still images for the usability study for the following reasons. Learning the interface of the program would make the time a tester need to dedicate to study much longer, which would have resulted in less volunteers. Moreover, the interface could have played a role in changing the results because of its design and we did not want the interface design to be a factor in evaluating the visualization methods. Consequently, motion parallax was not available to help viewers understand the spatial arrangement of objects as well as depth better. Motion parallax has been shown to positively affect perception of depth as discussed in Chapter 2.

5.2.4.2 No Spatial Orientation

Still images also made it impossible for the testers to understand the orientation, as well as the location of the camera relative to the volume. This was helped a bit by creating more descriptive primary questions. However, we believe our visualization methods can potentially convey much more information if the tester could freely navigate through the 3D visualization and understood their orientation and location in space. Although this is available during the

normal use of our method, we could not take advantage of it during the study because of the nature of still images.

5.2.4.3 Concurrent Availability of Methods

The visualization methods that we test individually can be present all at the same time to the user. Therefore, the user of these methods can freely switch between various modes of visualizations to take advantage of their different properties depending on the situation. In our test a single visualization method was used for each image. This does not represent the ability of this method to be used along with other methods. There are cases where one visualization method works better than the other. However, this ability was left out of the usability study.

5.2.4.4 Lack of Previous Training

Radiologists go through many years of training. Moreover, for each new commercial software there could be many months of training in addition to academic training. However, none of our testers were trained with any of our visualization methods except for the reference, which resembles current commercial 3D visualization methods. The only training radiologists received about this study was an information sheet that was presented to them and was designed to take roughly 5 minutes to read. It explained the steps of the study as well as a single image for each visualization method. This information sheet is presented in Appendix B.

5.3 Results

In this section, we present the results of our usability study. We have analyzed the data we gathered from our testers using statistical methods. As stated before the general null hypothesis of this study is that the images generated by

our new visualization methods do not differ from the reference images in terms of clinical diagnosis. We have examined four different aspects of clinical diagnosis that we hypothesized our visualization methods might improve. These four different aspects are abnormality assessment, correctness of marking, distance of marked location, and time. Each of these aspects of clinical diagnosis are then divided into further subcategories that will be discussed in detail.

We have used analysis of variance (ANOVA) [11] with $\alpha = 0.05$ to test our null hypothesis. Analysis of variance can test if the means of all participating groups are the same or not. However, in order to get insight into pairwise comparison of groups we need to use another test. In this case t-test with the desired alpha value cannot be used since by comparing each pair the chances of wrongly rejecting the null hypothesis (type I error) increases. In such situations multiple comparisons test can provide an upper limit to the alpha value and avoid the type I error problem caused by using t-test multiple times. In multiple comparisons test a corrected alpha value is used for testing each pair of groups so that the final alpha value is not greater than the desired significance level. Various alpha correction methods with different guarantees, strengths, and weaknesses are available to be used with multiple comparisons test to compensate for the error while comparing multiple groups in a pairwise manner.

We have used the most conservative correction method called the Bonferroni correction. Bonferroni correction guarantees that the chances of type I error never goes beyond our desired level (in this case 5 percent). However, it increases the chances of type II error, where a significant difference between two group means might be more likely to remain undetected.

The fact that we are seeing many significant differences between group means in the upcoming results shows that even with using methods as conservative as Bonferroni correction type II error is not big concern. We use the

same argument to dismiss the need for power analysis. Power analysis determines the sample size required to limit the chances of type II error. However, since our results already show significant differences between the group means power analysis is unnecessary. In a few cases in our results there are boundary cases where using a less conservative multiple comparison correction method can result in some group means to be significantly different from others.

Our results from ANOVA test as shown in Figure 5.8 are represented by box plots. In such plots each visualization category is represented by a column. Matlab's explanation of the box plots used in this chapter is presented here: *"On each box, the central mark is the median and the edges of the box are the 25th and 75th percentiles (1st and 3rd quantiles). The whiskers extend to the most extreme data points that are not considered outliers. The outliers are plotted individually. The interval endpoints are the extremes of the notches. The extremes correspond to $q_2 - \frac{1.57 \times (q_3 - q_1)}{\sqrt{n}}$ and $q_2 + \frac{1.57 \times (q_3 - q_1)}{\sqrt{n}}$, where q_2 is the median (50th percentile), q_1 and q_3 are the 25th and 75th percentiles, respectively, and n is the number of observations¹."* Figure 5.7 breaks down the information represented by each box in the box plots such as Figure 5.8.

In order to illustrate the pairwise comparison and the significant difference between different group means, visualization of the results from multiple comparisons test has been used. An example of such visualization is shown in Figure 5.10. Each short horizontal line represents the mean estimate with the comparison interval around it for a visualization category. The means for two groups are significantly different if there is no vertical overlap between the horizontal line plotted for the two groups. However, if there is a vertical overlap then we cannot reject the null hypothesis that there is no difference between the means for the two groups.

¹<http://www.mathworks.com/help/stats/anova1.html> (accessed Dec 5th, 2015).

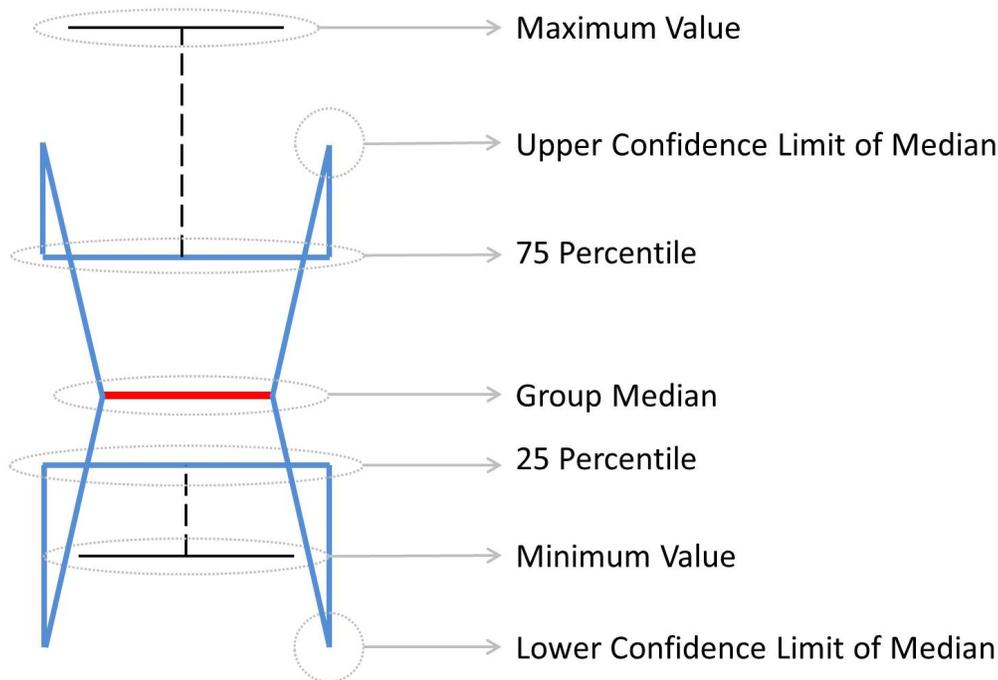


Figure 5.7: The information presented by each box in a box plot for each visualization category. The angled blue lines are referred to as the notches.

5.3.1 Abnormality Assessment

One of the most important aspects of our visualization techniques that needed to be studied is their impact on detecting abnormalities by radiologists. The correctness of answers to primary questions, where the tester is asked to confirm whether an abnormality is present or not, are used as an indicator of the impact of our proposed visualization methods on the ability of radiologists to detect various types of abnormalities. Consequently, the test-taker's answer to the primary question is the basis for this category of results. We have divided this test into 3 sub-tests to ensure we have taken all the different aspects of assessment of abnormalities into consideration.

The first test in this category is the number of correct answers to the primary questions. This test evaluates the number of abnormalities that were detected correctly and the number of abnormalities that were missed. The

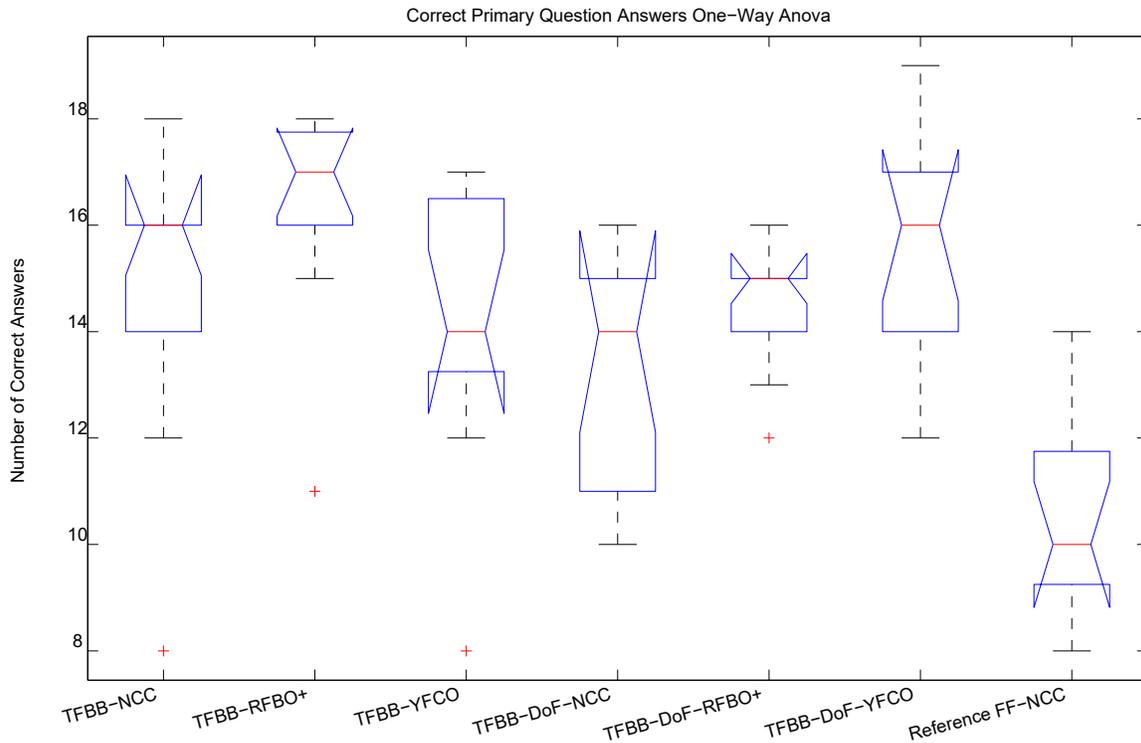


Figure 5.8: Results from comparing mean number of correct primary question answers among different groups using one-way ANOVA.

Source	SS	df	MS	F	Prob>F
Columns	236	6	39.3333	8.21	9.82612e-07
Error	335.455	70	4.7922		
Total	571.455	76			

Figure 5.9: ANOVA table from comparing mean number of correct primary question answers among different groups using one-way ANOVA.

second test examines false positive answers to primary questions of different images. In this case, we have compared the possibility of detecting an abnormality while there was none among different visualization methods. The third and final category assesses false negatives. In this test, we have compared the possibility of missing an abnormality while there is one present in the image shown to the tester among different visualization methods.

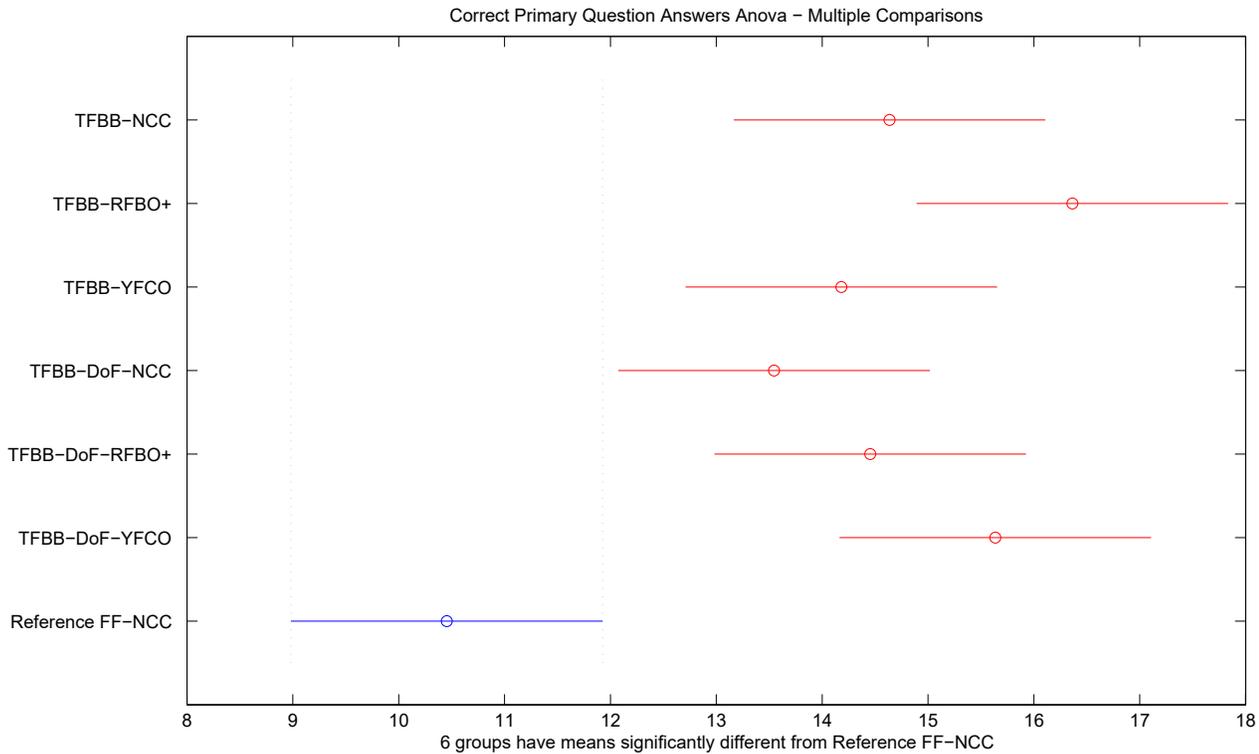


Figure 5.10: Results from comparing the reference with our visualization techniques using multiple comparisons tests. All our visualization techniques have significantly higher mean number of correct answers to primary question.

The first and third tests are very important in medical imaging. Although the second test is also important, it is not as vital as the other two categories. The reason is that in case of a false positive the patient is usually sent for further testing and assessment, however, in case of a missed abnormality the life of the patient could be in danger.

5.3.1.1 Correct Assessment

Figure 5.8 shows the results from ANOVA followed by Figure 5.10 that shows results from multiple comparisons test for comparing the number of correct answers to primary questions among the visualization techniques under test. Our results show that all of our proposed visualization methods are significantly better than the reference, $\langle FF, NCC \rangle$, in terms of mean correct clinical di-

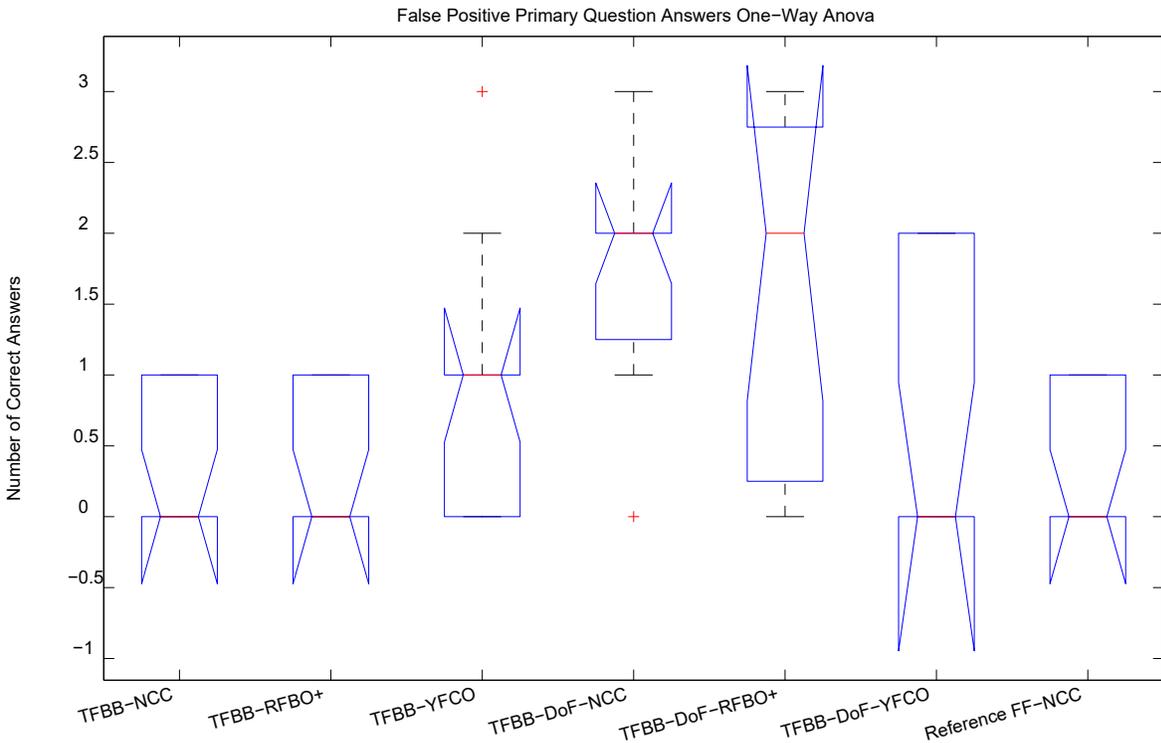


Figure 5.11: Results from comparing mean number of false positive primary question answers among different groups using one-way ANOVA.

Source	SS	df	MS	F	Prob>F
Columns	21.4545	6	3.57576	5.12	0.0002
Error	48.9091	70	0.6987		
Total	70.3636	76			

Figure 5.12: ANOVA table from comparing mean number of false positive primary question answers among different groups using one-way ANOVA.

agnosis. Our results also show that currently one cannot say that there is a significant difference in terms of correctly assessing the existence or lack thereof of an abnormality in a patient between our visualization methods.

5.3.1.2 False Positive Assessment

Figure 5.11 shows the results from an ANOVA test comparing different visualization methods. Analyzing the results from the test by multiple comparisons

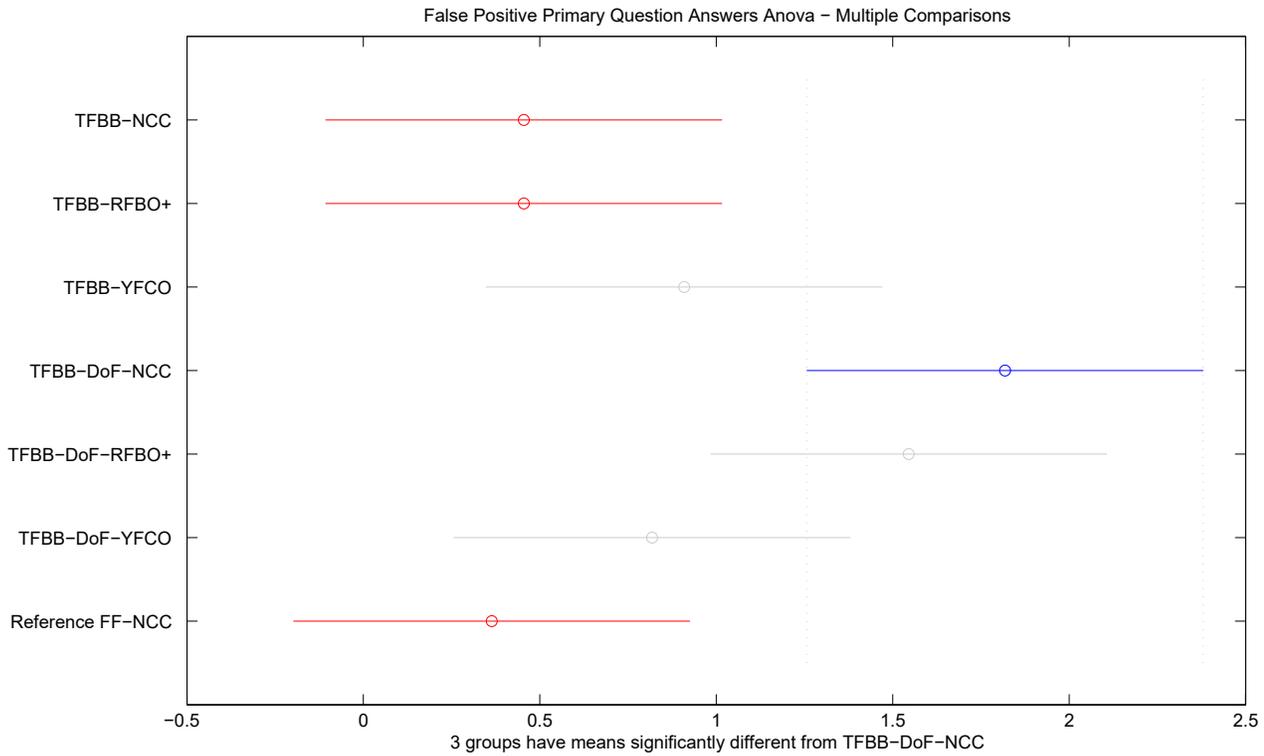


Figure 5.13: Results from comparing the $\langle TFBB, DoF, NCC \rangle$ with the rest of visualization techniques using multiple comparisons tests. $\langle TFBB, DoF, NCC \rangle$ results in significantly higher mean number of false positive answers to the primary question in comparison with $\langle TFBB, NCC \rangle$, $\langle FF, NCC \rangle$, and $\langle TFBB, RFBO+ \rangle$.

test reveals an interesting behavior among our different visualization methods. Figure 5.13 shows that rendering using $\langle TFBB, DoF, NCC \rangle$ causes the number of false positives in terms of existence or lack thereof of an abnormality to be significantly higher than the reference $\langle FF, NCC \rangle$, $\langle TFBB, NCC \rangle$, and $\langle TFBB, RFBO+ \rangle$.

It is worth noting that enabling DoF while using $\langle TFBB, NCC \rangle$ increases the possibility of falsely reporting a non-existing abnormality. Moreover, since there is no significant difference in terms of mean number of false positives between reference, $\langle FF, NCC \rangle$, and $\langle TFBB, RFBO+ \rangle$ it is interesting to observe from Figure 5.14 that introducing DoF to $\langle TFBB, RFBO+ \rangle$ makes the

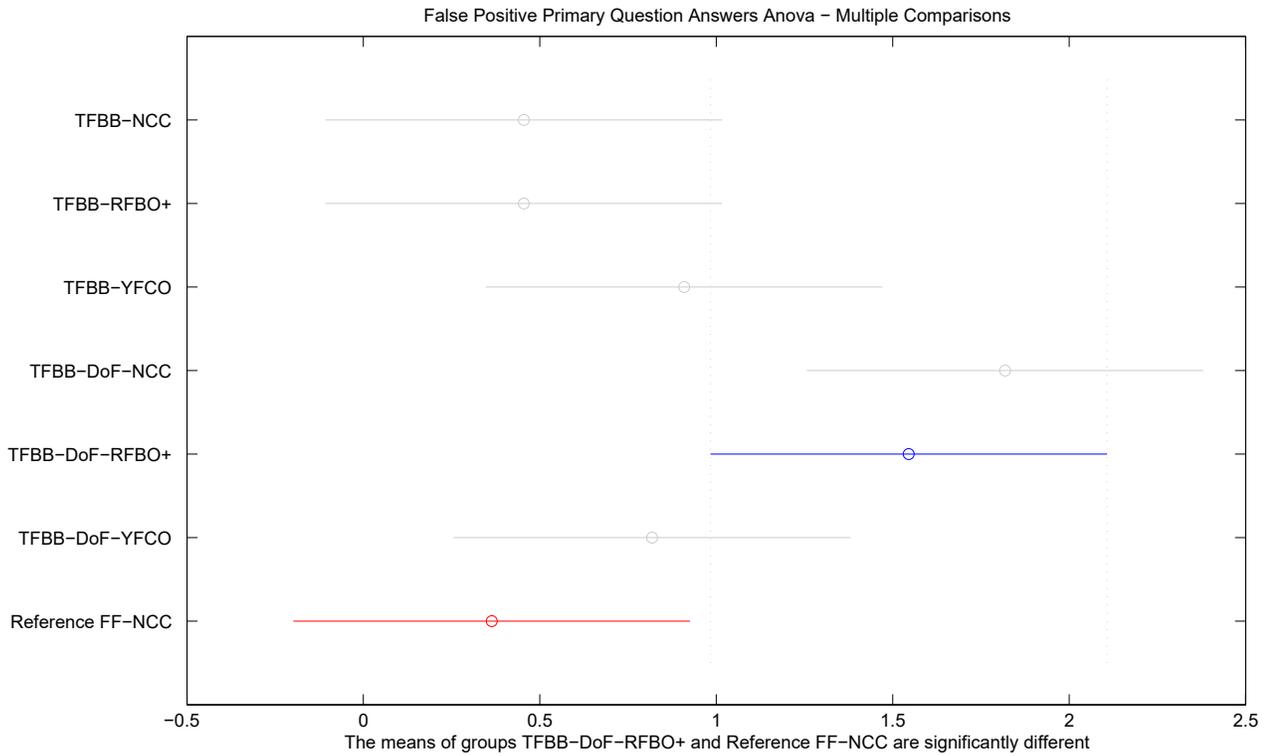


Figure 5.14: Results from comparing the $\langle TFBB, DoF, RFBO+ \rangle$ with the rest of visualization techniques using multiple comparisons tests. $\langle TFBB, RFBO+ \rangle$ results in significantly higher mean number of false positive answers to the primary question in comparison with $\langle FF, NCC \rangle$.

mean number of reported false positives significantly higher than the reference as well. It seems that in both cases addition of DoF caused physicians to mistakenly report more abnormalities. It is very important to us to find the methods that are better than the existing methods in every possible way, including false positives and false negatives.

5.3.1.3 False Negative Assessment

False negative mistakes are very important in medical diagnosis. False negatives will result in abnormalities to remain undetected, untreated, and consequently put the patients' health at risk. Figures 5.15 and 5.17 are showing the results of ANOVA and multiple comparison testing for the mean number of

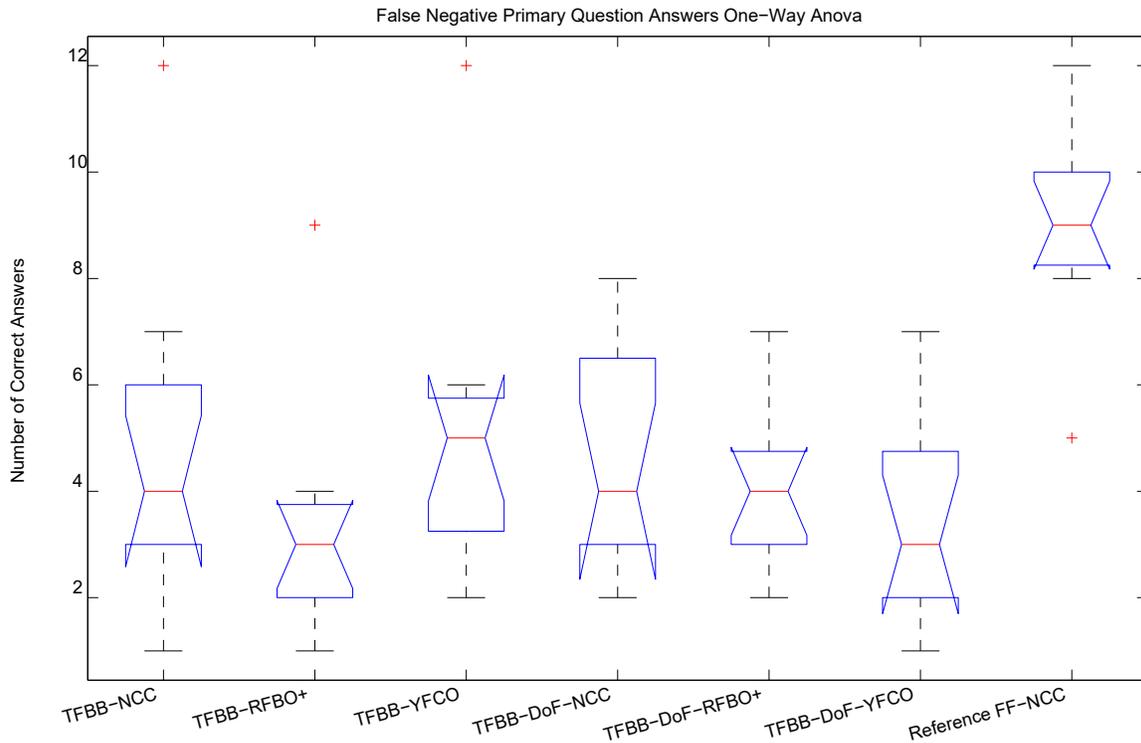


Figure 5.15: Results from comparing number of false negative primary question answers among different groups using one-way ANOVA.

Source	SS	df	MS	F	Prob>F
Columns	264	6	44	9	2.7788e-07
Error	342.364	70	4.89091		
Total	606.364	76			

Figure 5.16: ANOVA table from comparing number of false negative primary question answers among different groups using one-way ANOVA.

false negatives for each visualization method. All of our visualization methods have significantly less chance of missing an abnormality when compared to the reference. This means that all of our visualization methods significantly reduce the risk that a patient’s problem remains undetected. Our results, however, are not showing a significant different among our proposed methods.

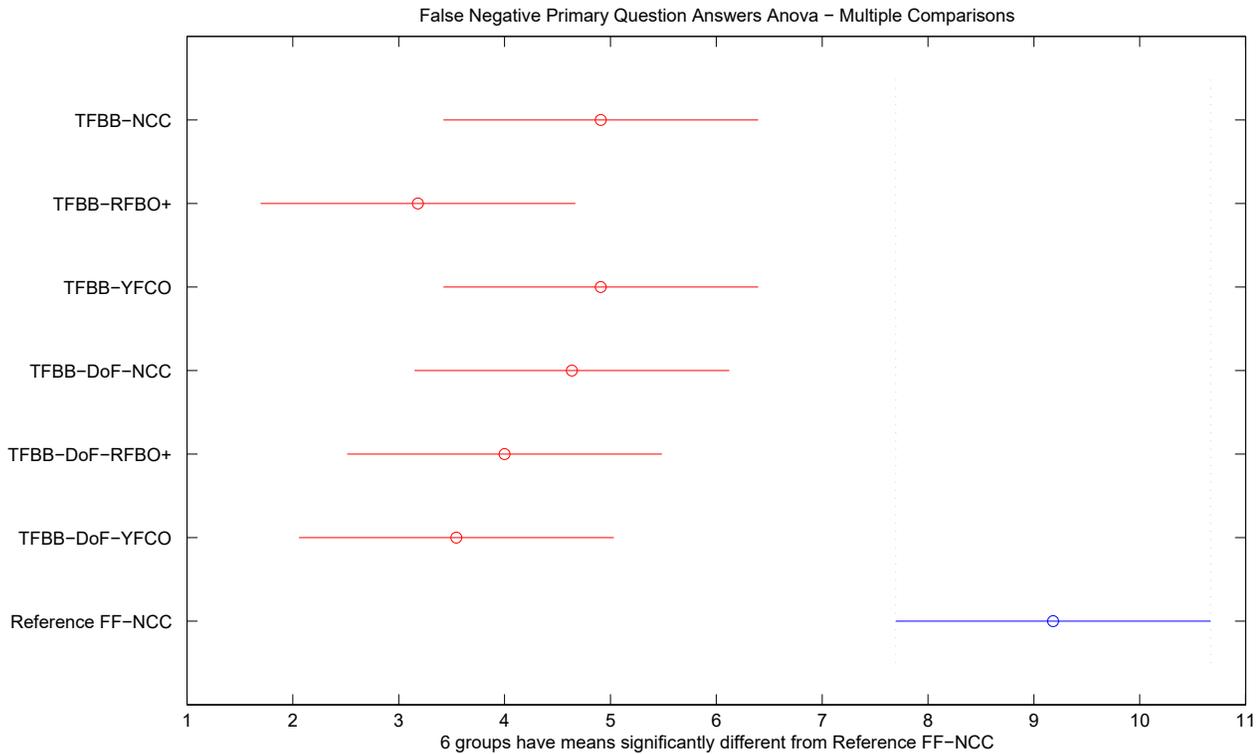


Figure 5.17: Results from comparing the reference $\langle FF, NCC \rangle$ with the rest of visualization techniques using multiple comparisons tests. $\langle FF, NCC \rangle$ results in significantly higher mean number of false negative answers to the primary question in comparison with all the rest of the visualization methods. It means that all of our methods significantly reduced the possibility of missing abnormalities.

5.3.2 Correctness of Marking

Although tests based on primary questions provided us with useful insight into the effectiveness of our visualization methods, we need an even stronger confirmation of the validity of our claim that our visualization methods enhance clinical diagnosis. Correctness of marking tests reveal more information regarding the effectiveness and practicality of our visualization methods. These tests are based on the correctness of the response to the secondary question. Secondary questions ask radiologists participating in the study to mark the defect with a single click of the mouse. This question is only presented to the

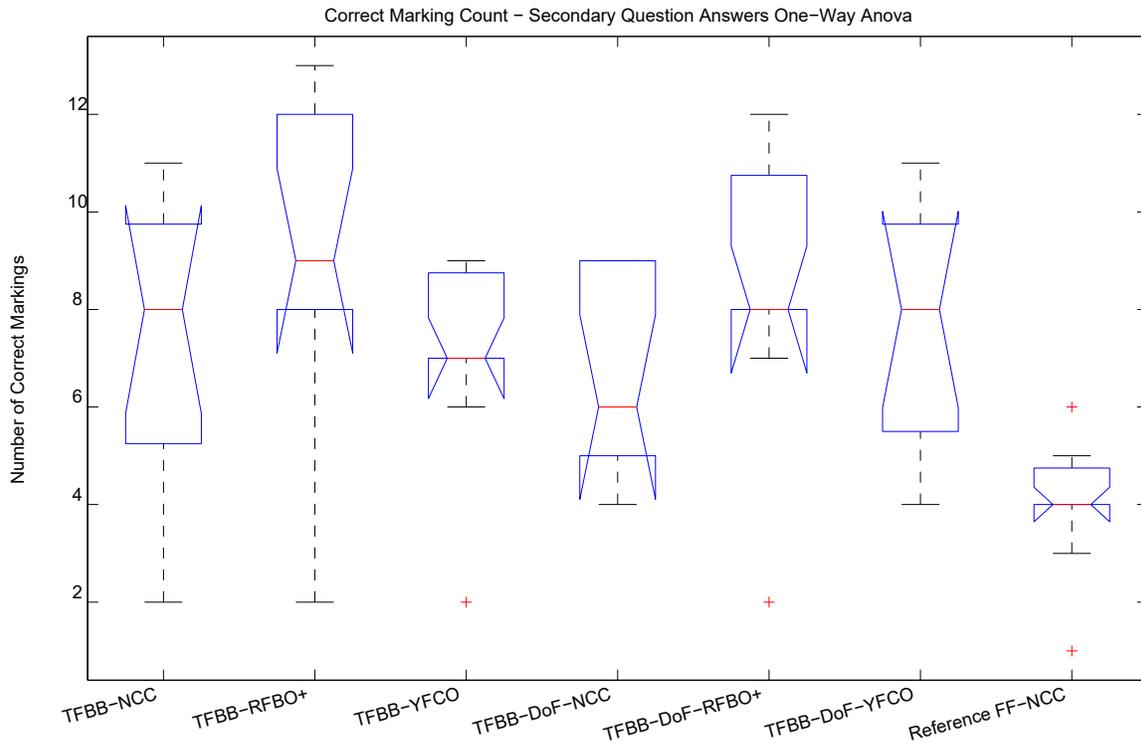


Figure 5.18: Results from comparing mean number of correct answers to secondary questions. That is, mean number of correct markings of the abnormality among different visualization techniques using one-way ANOVA.

Source	SS	df	MS	F	Prob>F
Columns	186.494	6	31.0823	5.5	0.0001
Error	395.818	70	5.6545		
Total	582.312	76			

Figure 5.19: ANOVA table from comparing mean number of correct answers to secondary questions. That is, mean number of correct markings of the abnormality among different visualization techniques using one-way ANOVA.

tester if the answer to the primary question was positive, that is if the tester indicated that the abnormality of interest is visible in the image.

Each secondary question asks for a specific part of the defect to be marked such as the center, the most proximal part of the defect, etc. The threshold of error is chosen by the expert radiologist for each image during the data

extraction. A click by the tester is recorded as wrong marking if the distance from the click to the pre-determined location of the abnormality exceeds this threshold. There are three outcomes as a result of secondary questions: correct marking, wrong marking, and time-out. To be on the safe side in this study we have considered time-out as wrong marking as well.

There are three sets of results that we are going to analyze here. Correct Marking Test compares our visualization methods with each other as well as with the reference in terms of the mean number of correctly marked abnormalities. Correct Assessment - Wrong Marking compares the mean number of times among different visualization methods that the physician answered positively to the primary question but marking was recorded as wrong marking. Finally, Combined Missed Abnormalities compares the visualization methods in terms of mean number of all false negatives. This includes both false negative answers to the primary question and wrong markings for secondary questions all at the same time.

Correctness of Marking Tests are very strict about the accuracy of clinical diagnosis. For example, in primary question tests one cannot distinguish between an abnormality correctly detected by the tester and a positive answer based on something other than the abnormality of interest. However, in Correctness of Marking Tests we can definitively distinguish between such cases. Consequently, by studying these results we can create a stronger argument with more evidence to support our claims about the effect of our visualization methods on clinical diagnosis.

5.3.2.1 Correct Marking

Correct Marking Test compares the mean number of correctly marked images among different visualization methods in the study. Figures 5.18 and 5.20 show the result of this test. We can claim that this is an even stronger test

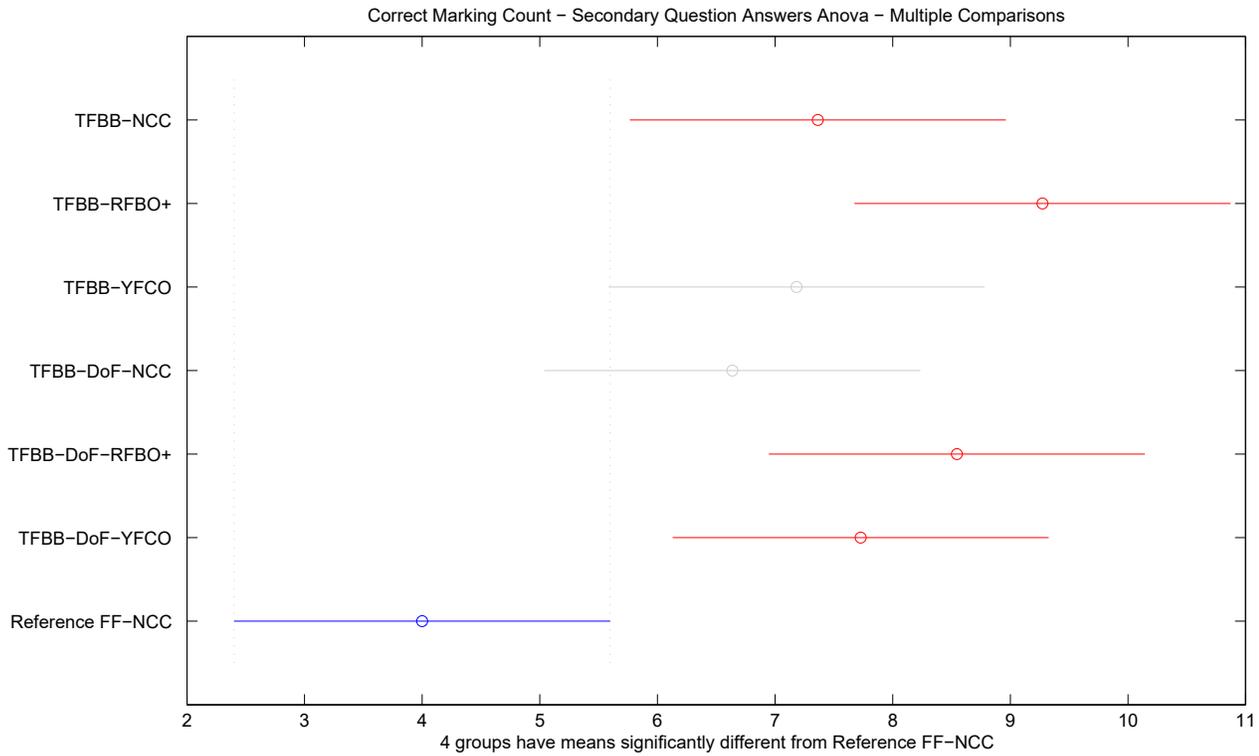


Figure 5.20: Results from comparing the reference with the rest of visualization techniques using multiple comparisons tests. Four visualization techniques are significantly better than the reference, $\langle FF, NCC \rangle$, in terms of correctly assessing and marking of the abnormality. This means that those four methods significantly increased the chances of correctly detecting an abnormality in visualization of CT data from a patient.

case for correctness of assessment. When a correct answer is recorded by this test it means two conditions were met. First, the abnormality needs to be correctly detected by the tester. Second, the marking needs to be accurate and within the small error threshold of the desired location. Consequently, this test eliminates the cases where the participant deemed something other than the intended abnormality as the desired abnormality. In such cases the tester would answer positively to the primary question despite missing the abnormality of interest.

Figures 5.18 and 5.20 show that $\langle TFBB, NCC \rangle$, $\langle TFBB, RFBO+ \rangle$,

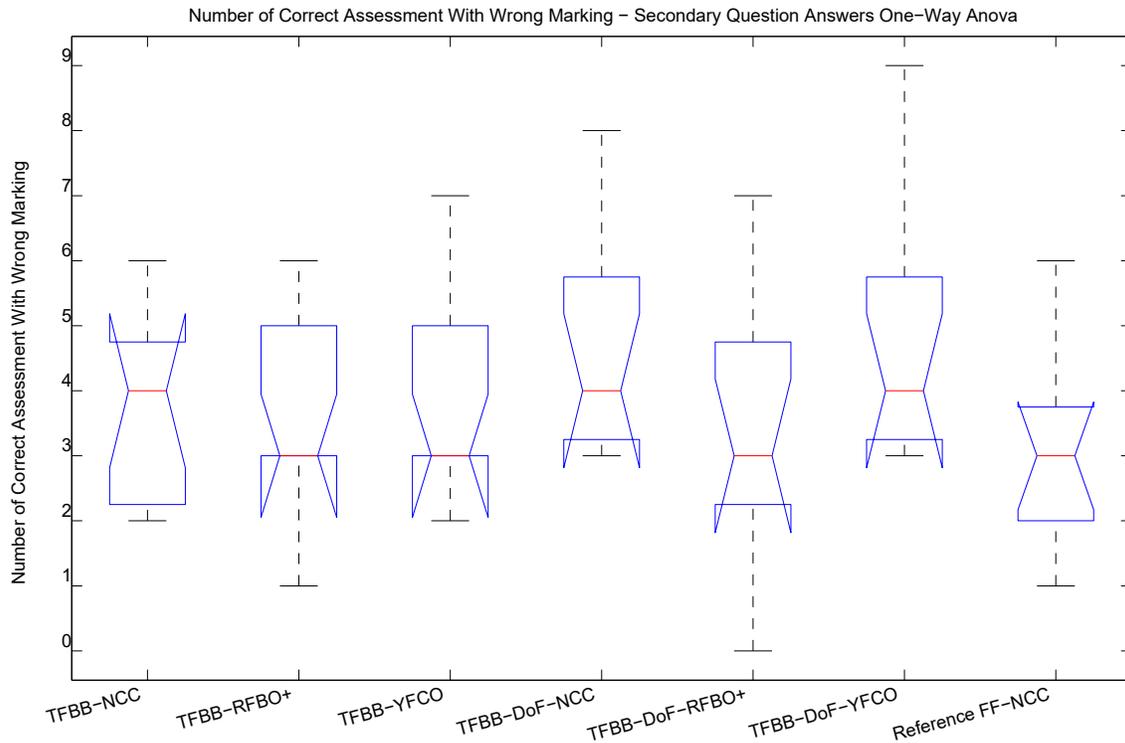


Figure 5.21: Results from comparing mean number of wrong answer to a secondary question while the tester correctly answered to the primary question of the image with abnormality.

Source	SS	df	MS	F	Prob>F
Columns	31.584	6	5.26407	1.87	0.0973
Error	196.545	70	2.80779		
Total	228.13	76			

Figure 5.22: ANOVA table from comparing mean number of wrong answer to a secondary question while the tester correctly answered to the primary question of the image with abnormality.

$\langle TFBB, DoF, RFBO+ \rangle$, and $\langle TFBB, DoF, YFCO \rangle$ are significantly better than the reference, $\langle FF, NCC \rangle$, in terms of mean number of correct markings of the abnormality. This means that with 95% confidence we can reject the null hypothesis, which states that there is no difference between each of these four visualization techniques when compared to the reference.

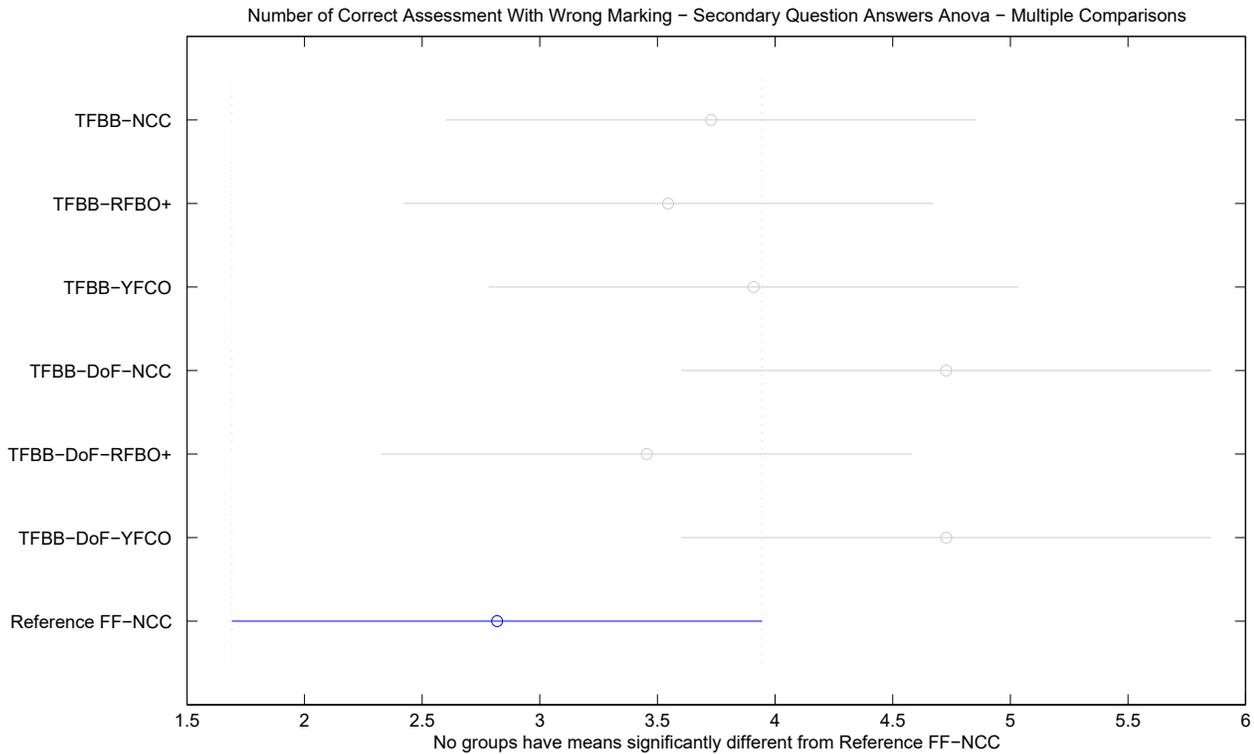


Figure 5.23: Results from comparing the reference with the rest of visualization techniques using multiple comparisons tests. No visualization techniques are significantly better or worse than the rest, in terms of correctly assessing the image while marking the image wrongly.

The results of this test provide strong evidence to back up our argument that some of our visualization techniques significantly enhance the number of correctly detected abnormalities for clinical diagnosis. This is one of the important aspects of enhancing clinical diagnosis in general.

5.3.2.2 Correct Assessment - Wrong Marking

Correct Assessment - Wrong Marking Test is to detect the cases where there was an abnormality present in the image and the testers answered positively to the primary question meaning they are seeing an abnormality; however, the result of marking the abnormality was recorded as wrong. We are going to look at different events that can result in such cases.

The first possibility is that the tester ran out of time. We mentioned earlier that we consider time-out results for secondary question as wrong marking. In this case, it is possible that the tester detected the abnormality correctly and did not have enough time to read the secondary question and mark the abnormality.

The second possibility that can result in a situation like this is when the tester detects the correct abnormality and manages to mark it in time, however, the single left click of the mouse falls out of the error threshold designated by the expert radiologist. We should note that these error thresholds are often quite small and subject to the opinion of the expert radiologist. It is possible that two different radiologists consider the center of an abnormality a bit differently without any of them missing the abnormality.

The third possibility is the case where the radiologist missed the abnormality of interest but found something else as abnormality. This case can itself be the result of two different causes. Either there was another abnormality that we either failed to hide or failed to dismiss properly by our question, or our image mislead the tester to consider a healthy part of the patient as abnormal and resulted in a false positive.

False positive in some cases results in further testing of the patient. Figures 5.21 and 5.23 visualize the result of our tests for mean number of correct assessments with wrong marking among different visualization techniques participating in the study. Our results suggest that we cannot reject the null hypothesis that there is no significant difference between our visualization methods and the reference method in terms of mean number of correct assessments with wrong markings. This means that we cannot claim that our methods increase or decrease such cases compared to each other or the reference method.

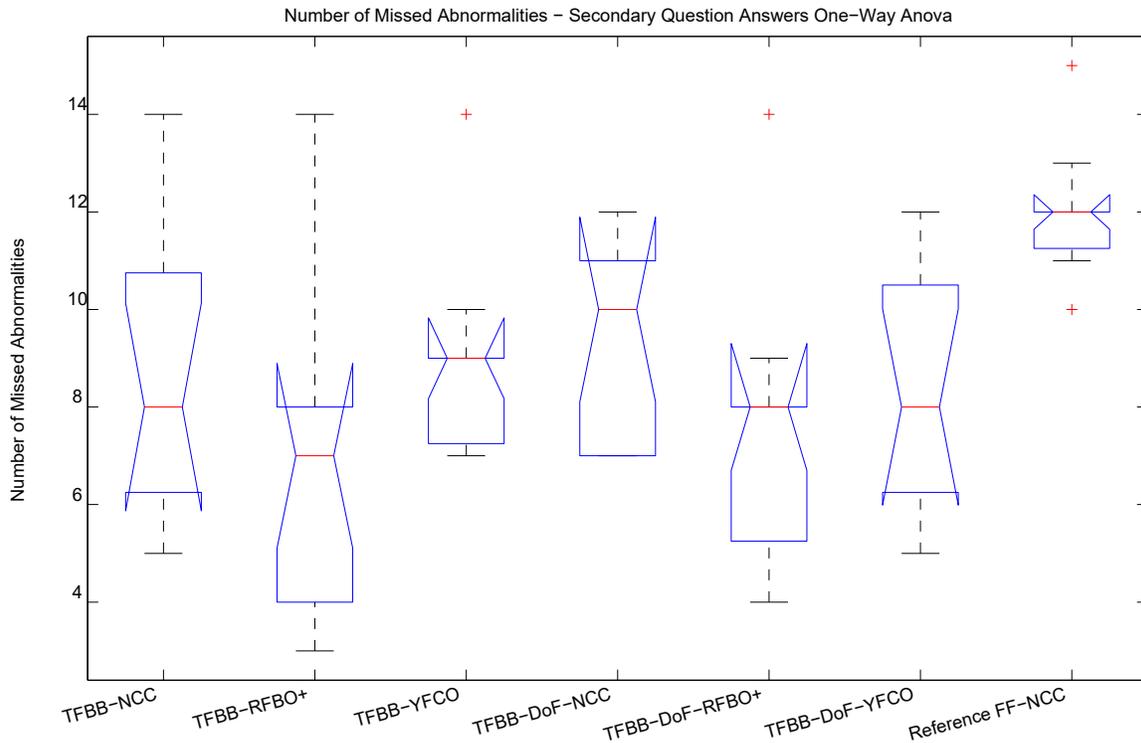


Figure 5.24: Results from comparing mean number of false negatives including answers to both primary and secondary questions for images with abnormalities.

Source	SS	df	MS	F	Prob>F
Columns	186.494	6	31.0823	5.5	0.0001
Error	395.818	70	5.6545		
Total	582.312	76			

Figure 5.25: ANOVA table from comparing mean number of false negatives including answers to both primary and secondary questions for images with abnormalities.

5.3.2.3 Combined Missed Abnormalities

In Section 5.3.1.3 we compared, our visualization methods with each other and with the reference in terms of false negative answers to the primary questions. False negatives are very important to us because they can be detrimental to the patient. The test in Section 5.3.1.3 only include the cases where the

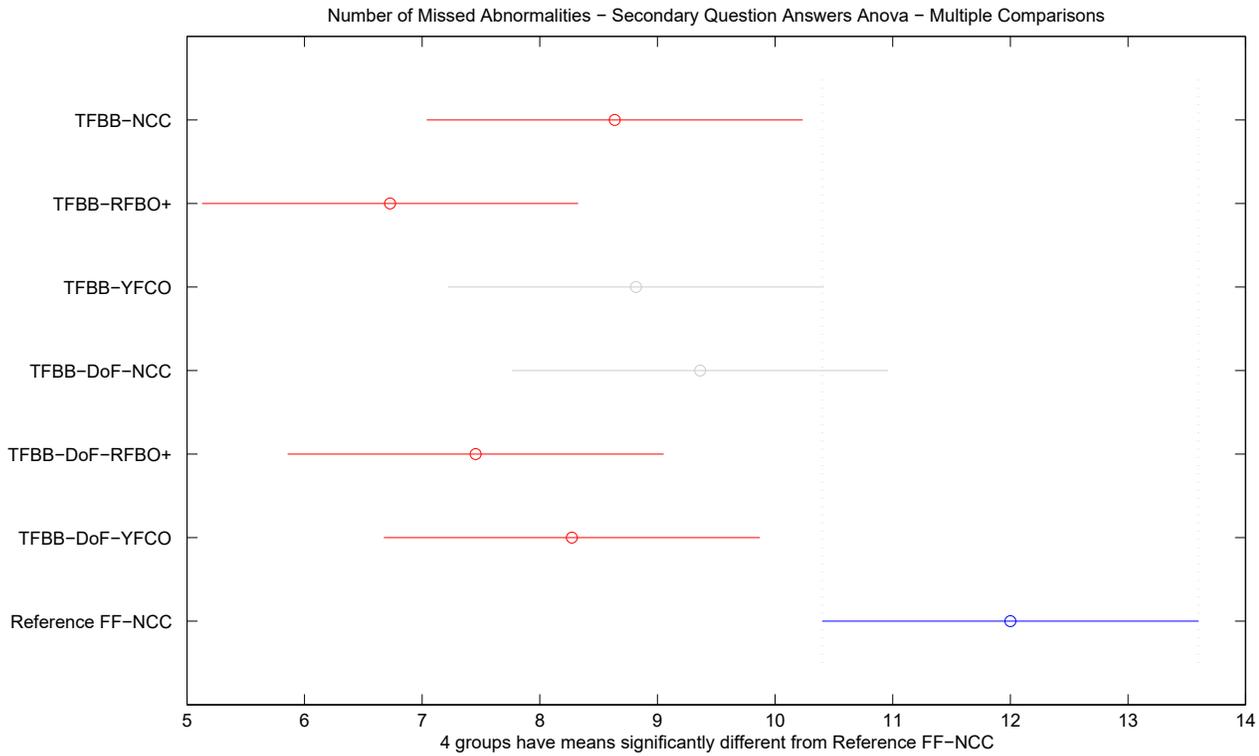


Figure 5.26: Results from comparing the reference with the rest of visualization techniques using multiple comparisons tests. Four of our visualization techniques significantly reduce the mean number of false negatives. This is very important since this shows that our visualization methods enhance the accuracy of clinical diagnosis and prevent an abnormality to be left untreated.

answer to the primary question is wrong. Combined Missed Abnormalities Test compares our visualization methods in terms of all false negative answers. This includes both false negatives of the primary questions and false negatives of the secondary questions.

Figures 5.24 and 5.26 show plots generated based on the results of Combined Missed Abnormalities test. ANOVA and the multiple comparisons test reveal that four of our proposed visualization methods namely $\langle TFBB, NCC \rangle$, $\langle TFBB, RFBO+ \rangle$, $\langle TFBB, DoF, RFBO+ \rangle$, and $\langle TFBB, DoF, YFCO \rangle$, are showing significantly less total number of false negatives when compared to the reference $\langle FF, NCC \rangle$. *That is, our methods significantly reduce the possi-*

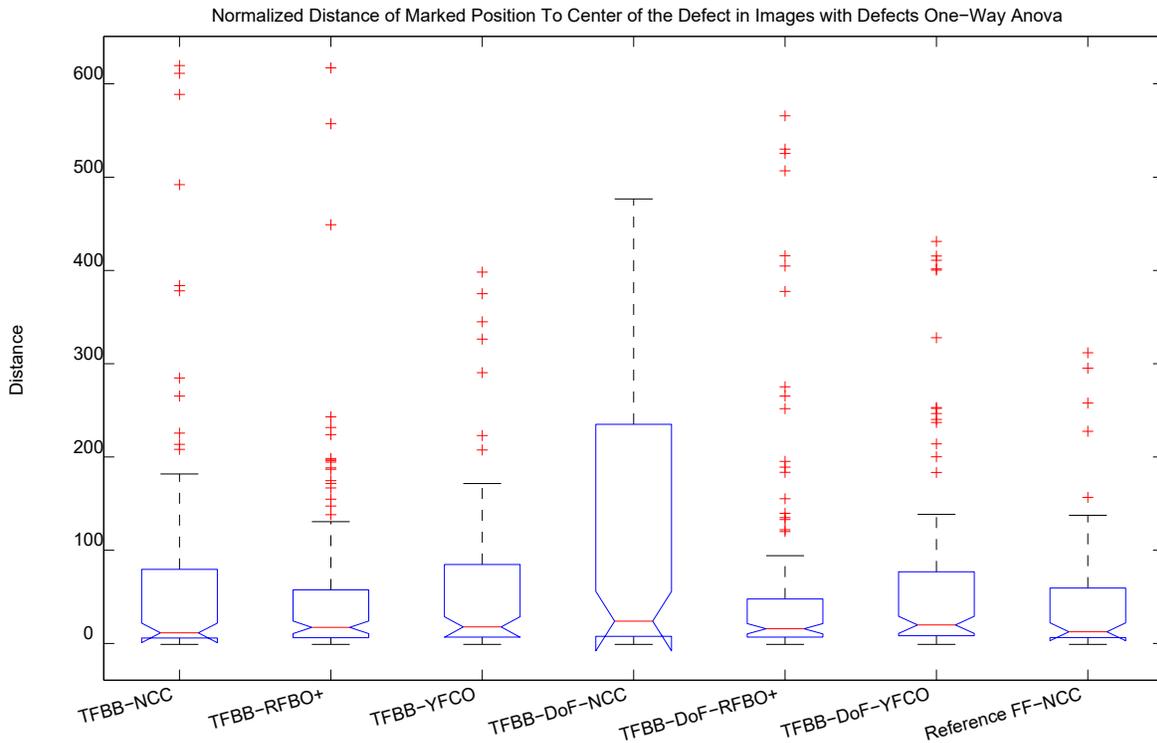


Figure 5.27: Results from comparing mean normalized distance to the center of the defect for images with correctly assessed abnormalities.

Source	SS	df	MS	F	Prob>F
Groups	329380.4	6	54896.7	4.83	7.35597e-05
Error	9626322.6	847	11365.2		
Total	9955703	853			

Figure 5.28: ANOVA table from comparing mean normalized distance to the center of the defect for images with correctly assessed abnormalities.

bility of missing abnormalities, which will ultimately lead to a more accurate clinical diagnosis.

5.3.3 Distance of Marked Location

A statistic that we could acquire and analyze during our study was the distance of the location marked by the tester to the location of the defect. Trivially

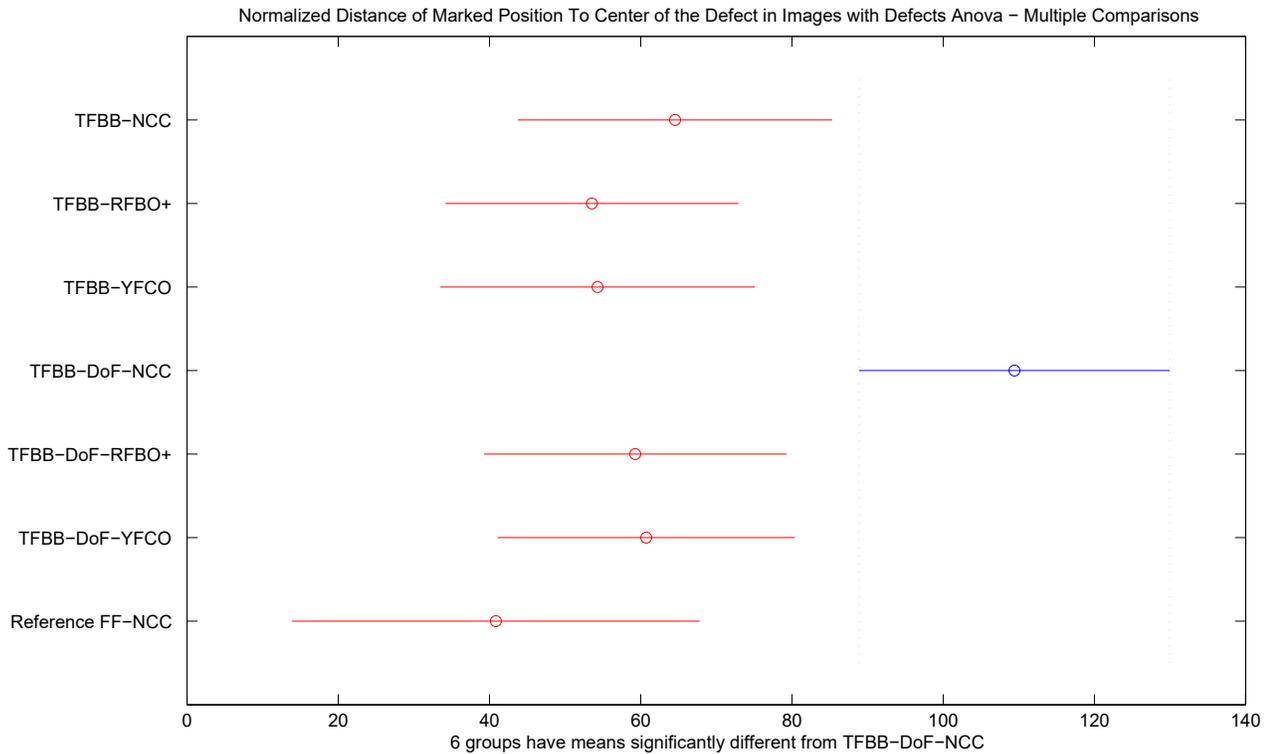


Figure 5.29: Results from comparing the $\langle TFBB, DoF, NCC \rangle$ with the rest of visualization techniques using multiple comparisons tests. Except for $\langle TFBB, DoF, NCC \rangle$ there is no significant difference between the rest of the methods. $\langle TFBB, DoF, NCC \rangle$ is showing significantly higher distance than the rest of the methods.

only images with abnormalities that were assessed correctly can be considered for this test. The location of the defect is a single point marked by the expert radiologist during data extraction. The location marked by the tester is the location of the response to the secondary question.

The error threshold chosen by the expert radiologist is there to take into account the size of the defect, camera properties, distance to the defect, etc. In order to take all these parameters into consideration when comparing distance statistics we have divided the distance by the error threshold and called it the normalized distance. Our normalized distance would be between 0 and 1 in case of correct marking and greater than 1 in case of wrong marking.

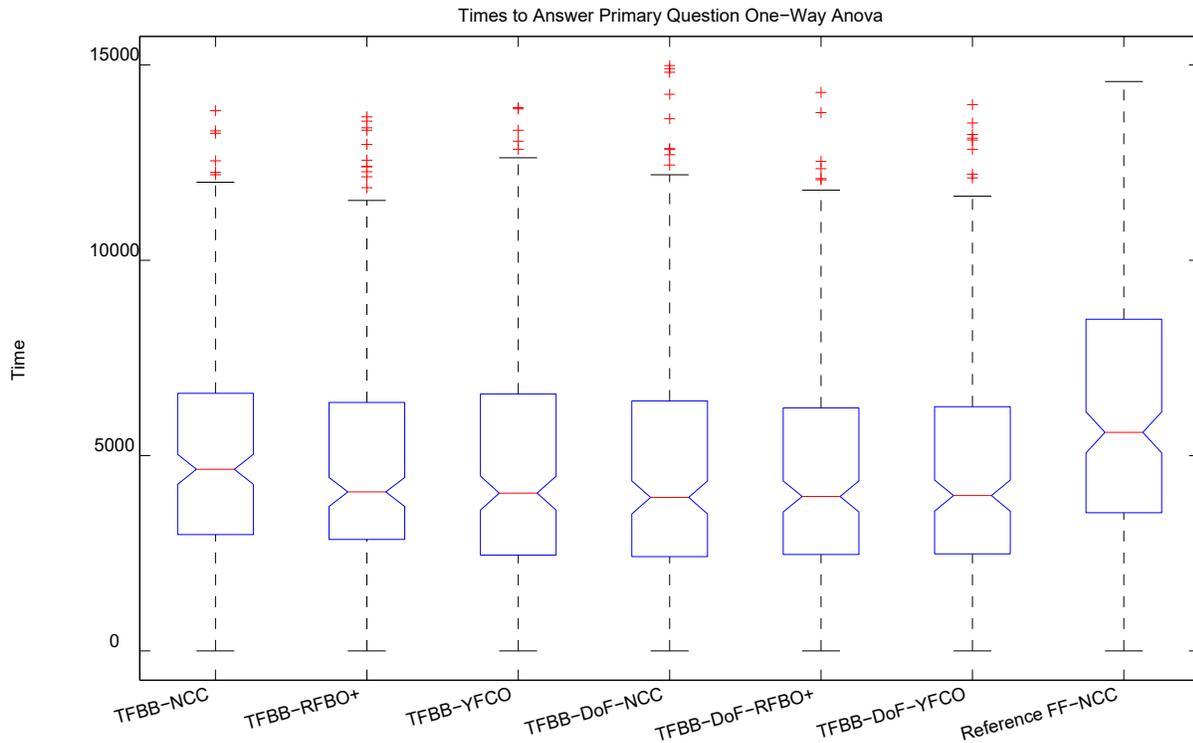


Figure 5.30: Results from comparing mean time to answer the primary question.

Source	SS	df	MS	F	Prob>F
Columns	3.12281e+08	6	5.20469e+07	5.49	1.24348e-05
Error	1.45417e+10	1533	9.48576e+06		
Total	1.4854e+10	1539			

Figure 5.31: ANOVA table from comparing mean time to answer the primary question.

Figures 5.27 and 5.29 show our results for comparing our visualization methods in terms of normalized distance statistics using ANOVA and multiple comparisons test. Except for $\langle TFBB, DoF, NCC \rangle$ we could not reject the hypothesis that there is no significance difference between the rest of our visualization methods. However, $\langle TFBB, DoF, NCC \rangle$ has a mean significantly different than the rest of the visualization methods including the reference. This shows the very interesting effect of chroma-coding on disambiguation of

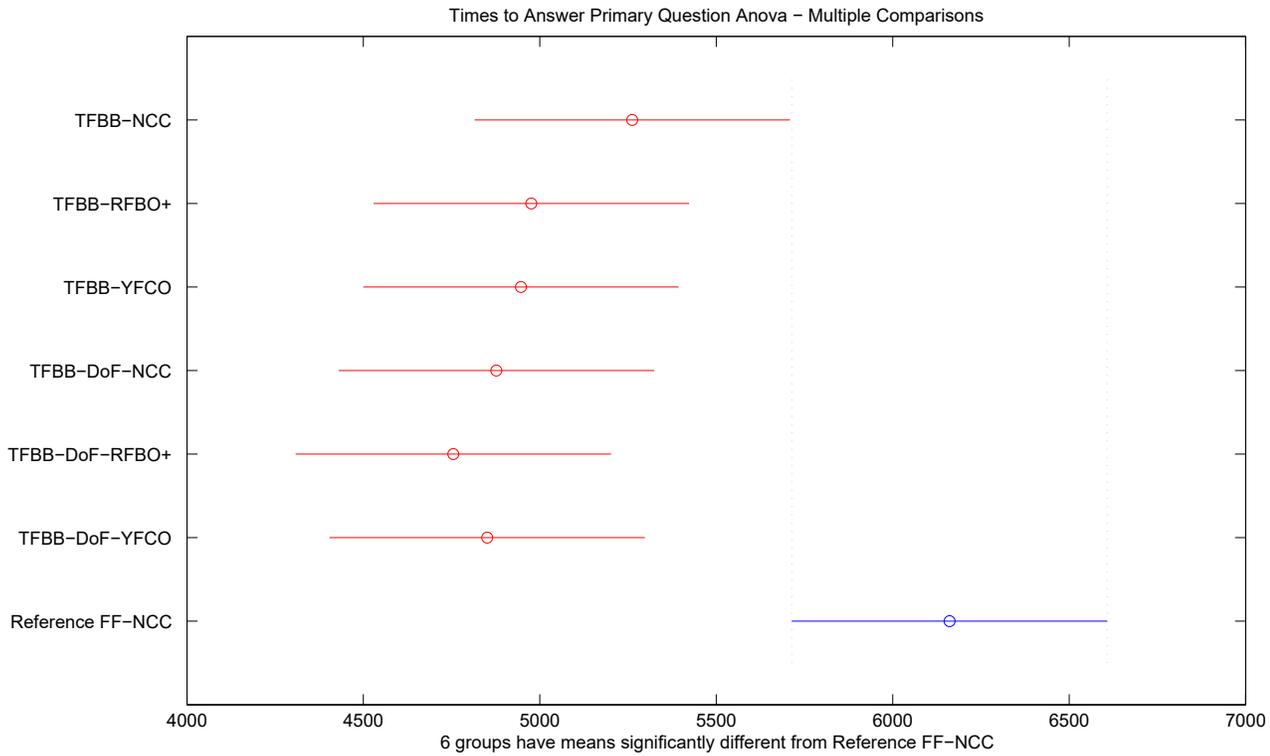


Figure 5.32: Results from comparing the $\langle FF, NCC \rangle$ with the rest of visualization techniques using multiple comparisons tests. All our visualization methods result is significantly shorter mean decision making time.

visualization.

Visualization methods $\langle TFBB, DoF, RFBO+ \rangle$ and $\langle TFBB, DoF, YFCO \rangle$ both show means not different than the rest. The only difference between these two visualization methods and $\langle TFBB, DoF, NCC \rangle$ is the absence of chroma-coding in $\langle TFBB, DoF, NCC \rangle$. The fact that in two different cases there is a significant difference between the mean normalized distance to the center of the defect caused by the absence or presence of FBCC provides us with strong evidence supporting the positive effect of FBCC on blur disambiguation and consequently clinical diagnosis.

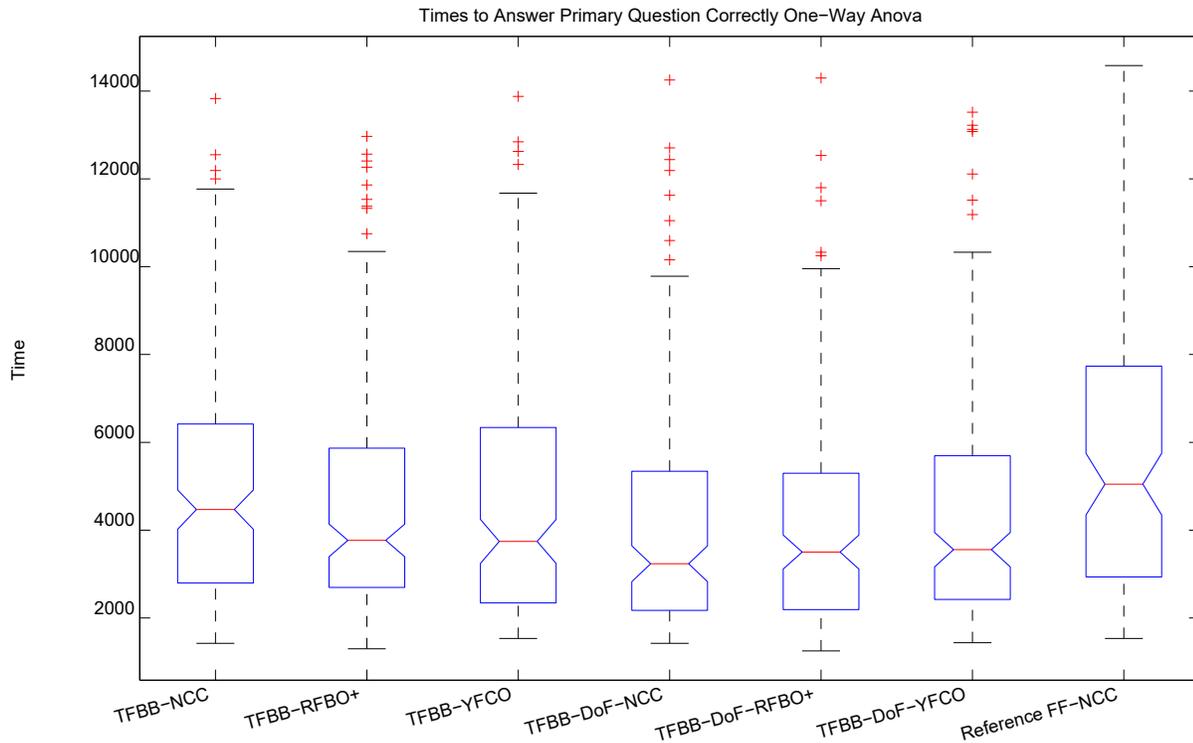


Figure 5.33: Results from comparing mean time to answer the primary question correctly.

Source	SS	df	MS	F	Prob>F
Groups	2.42538e+08	6	4.04231e+07	5.18	2.87275e-05
Error	8.46259e+09	1085	7.79962e+06		
Total	8.70512e+09	1091			

Figure 5.34: ANOVA table from comparing mean time to answer the primary question correctly.

5.3.4 Time

Our test is timed, which provides us with the opportunity of gathering time related statistics. Comparing visualization methods in terms of time to be answered correctly can provide us with insight into the level of confidence and unambiguity that different visualization methods can provide physicians with. However, we should note that it is also possible that visualization meth-

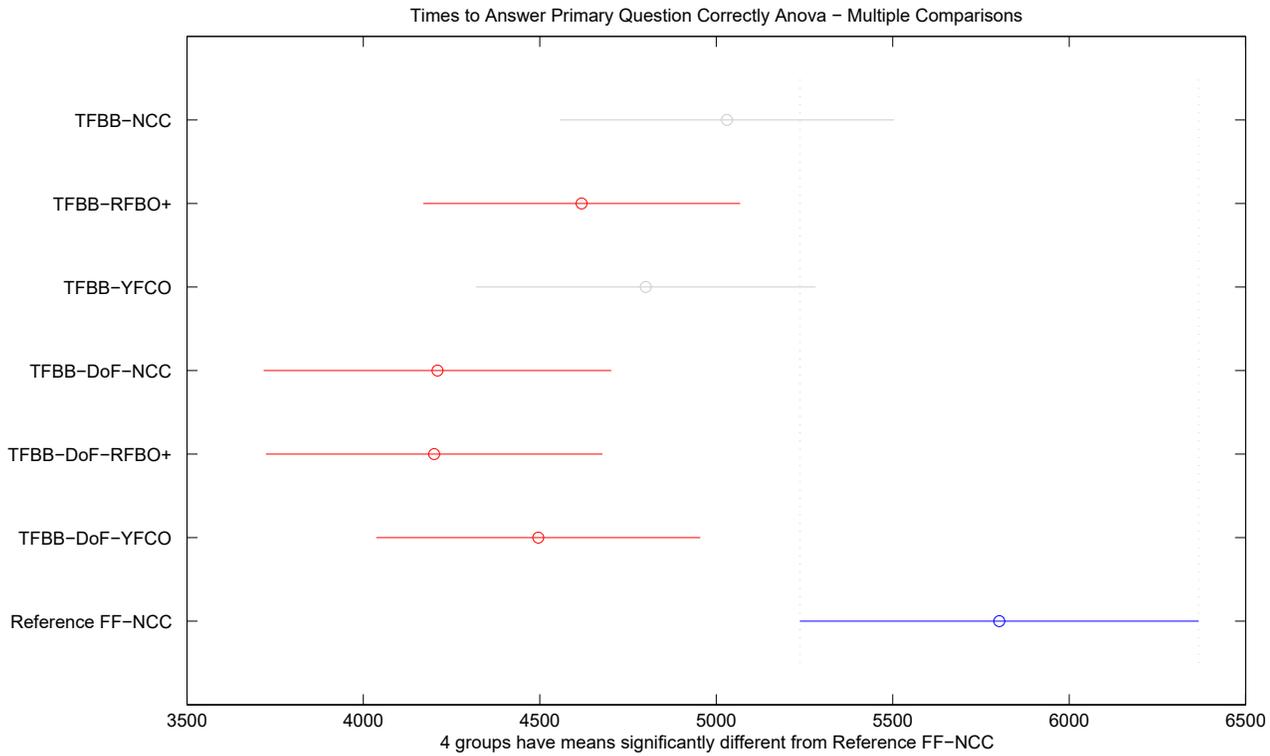


Figure 5.35: Results from comparing the $\langle FF, NCC \rangle$ with the rest of visualization techniques using multiple comparisons tests. $\langle TFBB, RFBO+ \rangle$, $\langle TFBB, DoF, NCC \rangle$, $\langle TFBB, DoF, RFBO+ \rangle$, and $\langle TFBB, DoF, YFCO \rangle$ result is significantly shorter mean correct decision making time.

ods achieve good time results because of providing physicians with wrong confidence and misleading them. We have so far shown that our visualization methods neither mislead physicians nor provide them with wrong information. Consequently, timed tests can be a good measure of clarity of the information that is conveyed to the physician.

5.3.4.1 Times To Answer Primary Questions

Times to Answer Primary Questions test compares participating visualization methods in terms of mean time it takes for the primary question to be answered. The results of this test presented in Figures 5.30 and 5.32 show that *it took the participants significantly less time to answer primary questions using*

our visualization methods compared to the reference. This means that participating radiologists could make a decision about whether the image exhibited the specified abnormality or not more rapidly.

We know from the results of previous tests that our visualization methods do not mislead physicians. Consequently, this test shows that not only our methods enhance the results, they also make it possible to come to a correct conclusion more rapidly.

5.3.4.2 Times to Answer Primary Questions Correctly

In order to test an even stronger variation of the previous test we are comparing our visualization methods in terms of mean time it took the testers to answer primary questions correctly. This test eliminates the cases where an incorrect answer was recorded for the primary question.

Figures 5.33 and 5.35 show the results for this test. Four groups namely $\langle TFBB, RFBO+ \rangle$, $\langle TFBB, DoF, NCC \rangle$, $\langle TFBB, DoF, RFBO+ \rangle$, and $\langle TFBB, DoF, YFCO \rangle$ have means significantly different than the reference $\langle FF, NCC \rangle$. *The results of this test provide us with stronger evidence that our visualization methods can reduce ambiguity and consequently the time it takes for the physicians to make a correct decision.*

5.3.5 Discussion

We have looked at different visualization methods and their effect on different aspects of clinical diagnosis. We compared our visualization methods with the reference, $\langle FF, NCC \rangle$, as a representative of what is generally available to physicians. Our results as well as the fact that 2D slices are predominantly the favored method of assessment in hospitals show that current commercial medical DVR methods lack the necessary means to convey correct information rapidly and accurately. *The goal of our study was to support our argument that*

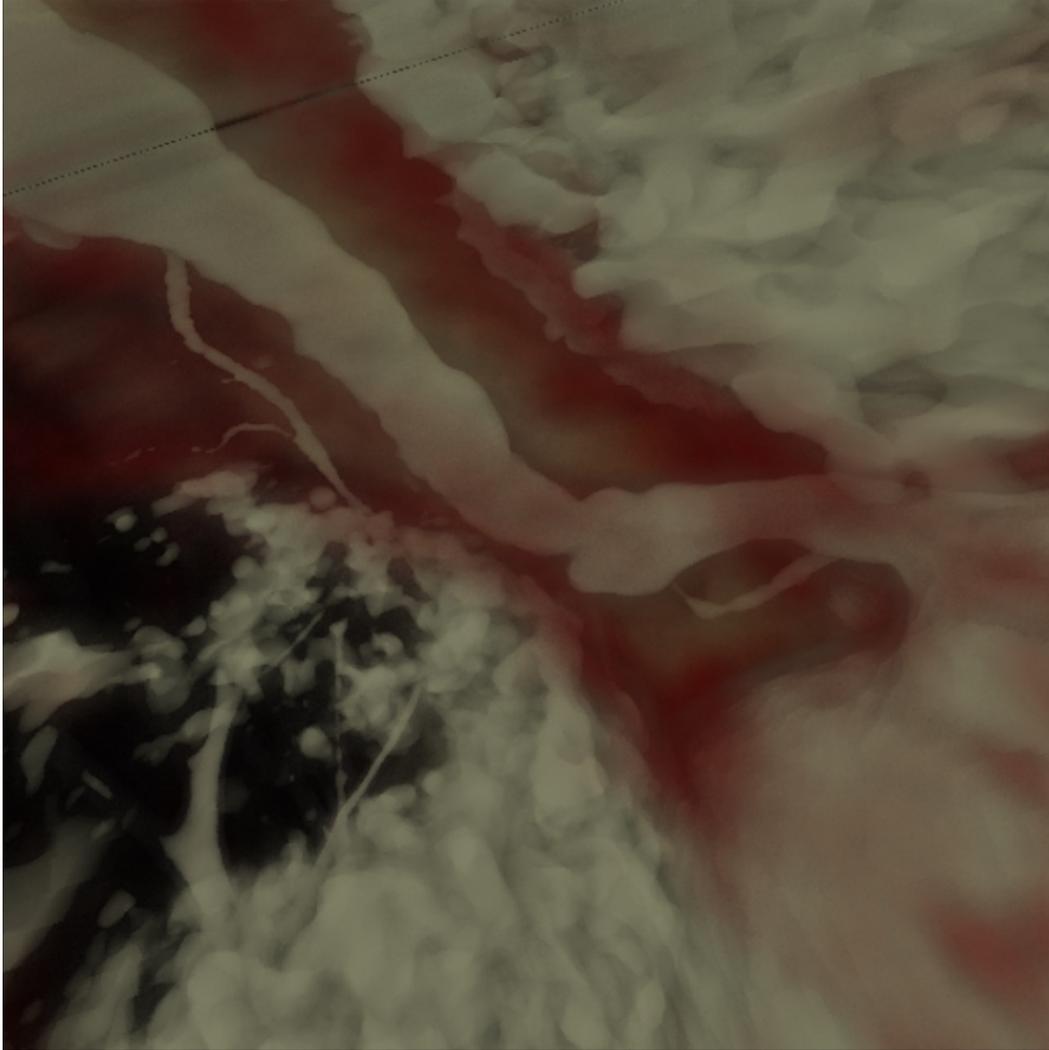


Figure 5.36: Coronary stenosis revealed by $\langle TFBB, NCC \rangle$.

our new visualization methods based on our algorithms proposed in Chapter 3 and Chapter 4 provide physicians with novel reliable, consistent, and accurate visualization methods.

So far in this chapter, we have presented the results from each test separately. It is now time to put all the results from all the different statistics, tests, and visualization methods together in order to better show the effect of our visualization methods on clinical diagnosis. Table 5.2 summarizes the result of comparing our visualization methods to the reference using ANOVA and multiple comparisons test. Rows represent tests and columns represent

our methods. Each cell represents the result of comparing the visualization method of its column with the reference using the test corresponding to the cell's row. A green cell with a check mark () means the corresponding visualization performed significantly better than the reference. A gray cell with a circle () means our test did not show a significant difference between the reference and our visualization method. Finally, a red cell with a “x” mark () shows that the corresponding visualization method performed significantly worse than the reference in that test.

Our results show that our methods generally performed better than the reference. They all significantly increased correct assessment, reduced possibility of missing abnormalities, and enhanced the time of assessment meaning they made it easier for the physician to make a decision. We are going to look at the pros and cons of each of our visualization methods.

5.3.5.1 $\langle TFBB, NCC \rangle$

$\langle TFBB, NCC \rangle$ uses only TFBB and no chroma-coding. Using this visualization method provided us with an opportunity to study the impact of TFBB alone on clinical decision making. Table 5.2 shows $\langle TFBB, NCC \rangle$ is significantly better than the reference, $\langle FF, NCC \rangle$, in five different tests and not significantly different in the other four. It significantly increases correct assessment, correct marking, and significantly reduces false negatives, total number of missed abnormalities, and time to assess a patient compared to the reference.

Using the results of tests on this method, one can conclude that TFBB alone significantly improves many important and vital aspects of clinical diagnosis while it does not negatively impact any of the tested aspects of clinical diagnosis. Figure 5.36 shows an image that is generated by the $\langle TFBB, NCC \rangle$. A coronary stenosis is easily identifiable in the image. Normally it is not easy

Table 5.2: Our visualization methods compared to the reference $\langle FF, NCC \rangle$.

Method \rightarrow \downarrow Test	TFBB-NCC	TFBB-RFBO+	TFBB-YFCO	TFBB-DoF-NCC	TFBB-DoF-RFBO+	TFBB-DoF-YFCO
Correct Assessment	✓	✓	✓	✓	✓	✓
False Positive Assessment	●	●	●	✗	✗	●
False Negative Assessment	✓	✓	✓	✓	✓	✓
Correct Marking	✓	✓	●	●	✓	✓
Correct Assessment Wrong Marking	●	●	●	●	●	●
Combined Missed Abnormalities	✓	✓	●	●	✓	✓
Distance of Marked Location	●	●	●	✗	●	●
Time of Assessment	✓	✓	✓	✓	✓	✓
Time of Correct Assessment	●	✓	●	✓	✓	✓

to get such a clear view. However, TFBB clears the clutter and makes it easier for the physician to make a decision.

5.3.5.2 $\langle TFBB, RFBO+ \rangle$

$\langle TFBB, RFBO+ \rangle$ exhibited one of the best results among different visualization methods. Compared to the reference it significantly increased mean number of correct assessments, and mean number of correct markings and significantly decreased mean number of false negative assessments, mean number of total missed abnormalities, mean time of assessment, and mean time of correct assessment. $\langle TFBB, RFBO+ \rangle$ outperformed $\langle TFBB \rangle$ in almost all of our tests. This also shows the positive impact of chroma-coding on enhancing the results of visualization with TFBB.

This configuration of our visualization is specially useful in cases of extremely cluttered views such as pulmonary abnormalities. For example, Figure 5.37 provides a view of pulmonary arteries taken from a patient's CT volume. The blue area close to the center of the image is a pulmonary embolism. It is easily detectable due to using TFBB to keep the lumen in focus and using RFBO+ chroma-coding, which emphasizes the difference of embolus region with its surroundings.

5.3.5.3 $\langle TFBB, YFCO \rangle$

$\langle TFBB, YFCO \rangle$ shows significant improvement in the number of correctly assessed images, number of false negative assessments and time to assess the image for abnormality when compared to the reference. Figure 5.38 shows an image rendered with $\langle TFBB, YFCO \rangle$ from a patient's CT data. The coronary stenosis can be easily seen while the TFBB removes the clutter and provides context, and FBCC provides useful color contrast between the focused materials and the blurred materials.

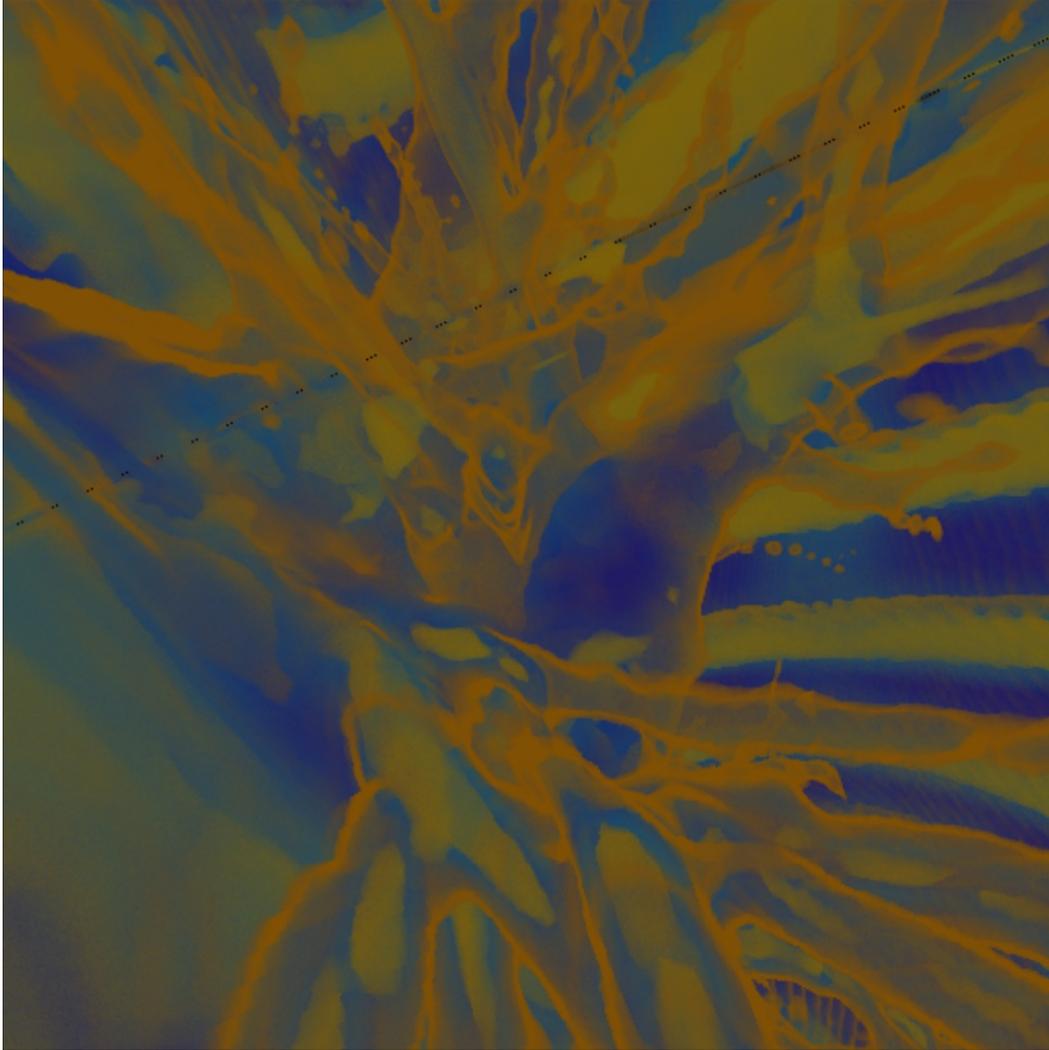


Figure 5.37: Pulmonary embolism, which is the blue area approximately at the center of the image, revealed by $\langle TFBB, RFBO+ \rangle$.

There is difference between the performance of $\langle TFBB, YFCO \rangle$ and $\langle TFBB, RFBO+ \rangle$. Both of these methods use TFBB and chroma-coding. But there are subtle differences between these two chroma-codings. The difference between YFCO and RFBO+ is in their choice of color and also the way color is applied to the image. In RFBO+ focused areas are shifted towards red and out of focus areas are shifted towards blue. Differences in blur level play an important role in RFBO+ creating a gradient of colors between red and blue. However, YFCO not only uses yellow and cyan as the extreme ends of its

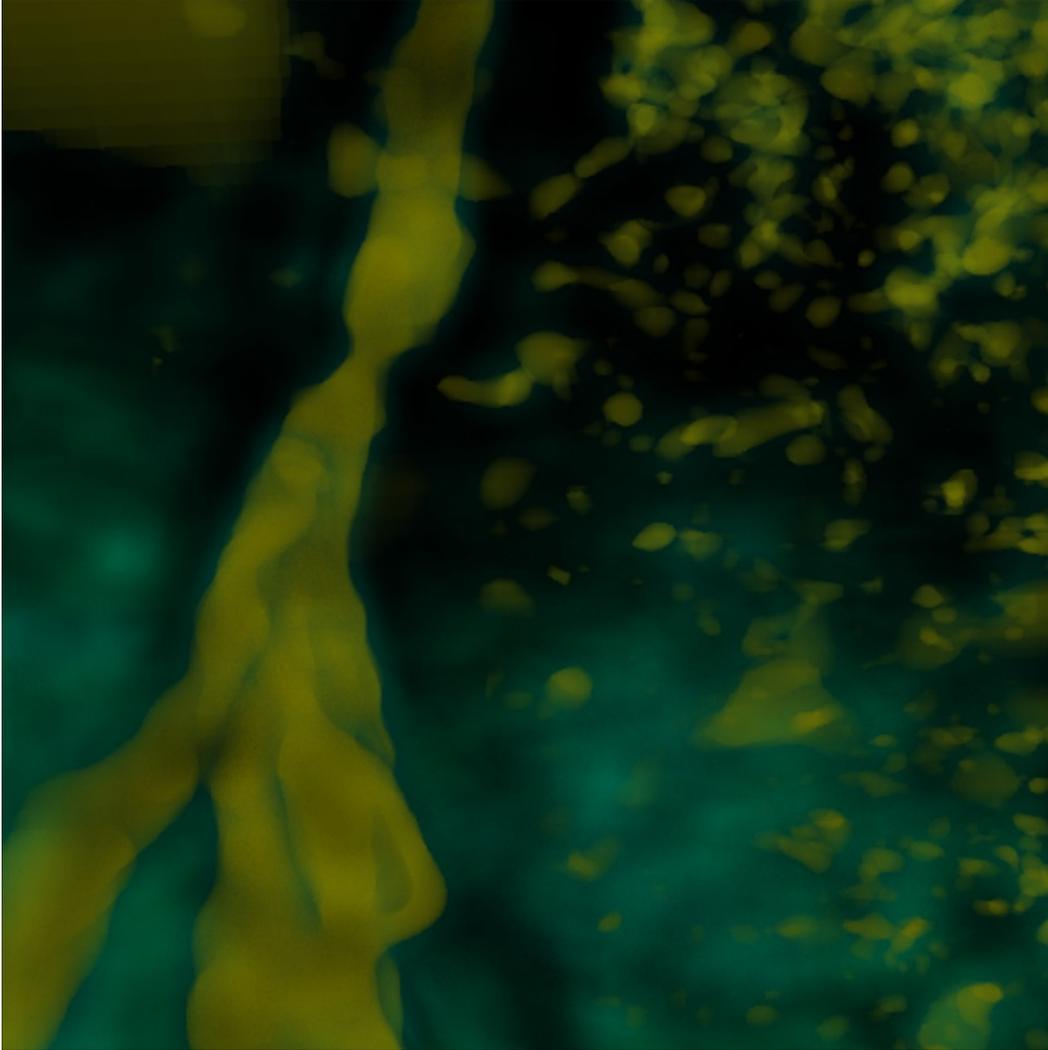


Figure 5.38: Coronary stenosis revealed by $\langle TFBB, YFCO \rangle$.

spectrum, but also does not take differences between blur levels into account. Based on our results, after comparing these two methods with the reference we can conclude that RFBO+ performed better than YFCO when combined only with TFBB.

5.3.5.4 $\langle TFBB, DoF, NCC \rangle$

The results from comparing $\langle TFBB, DoF, NCC \rangle$ with the reference reveals something that we asked in Chapter 4. The question was whether combining two blurring methods can be confusing for the viewer. According to our

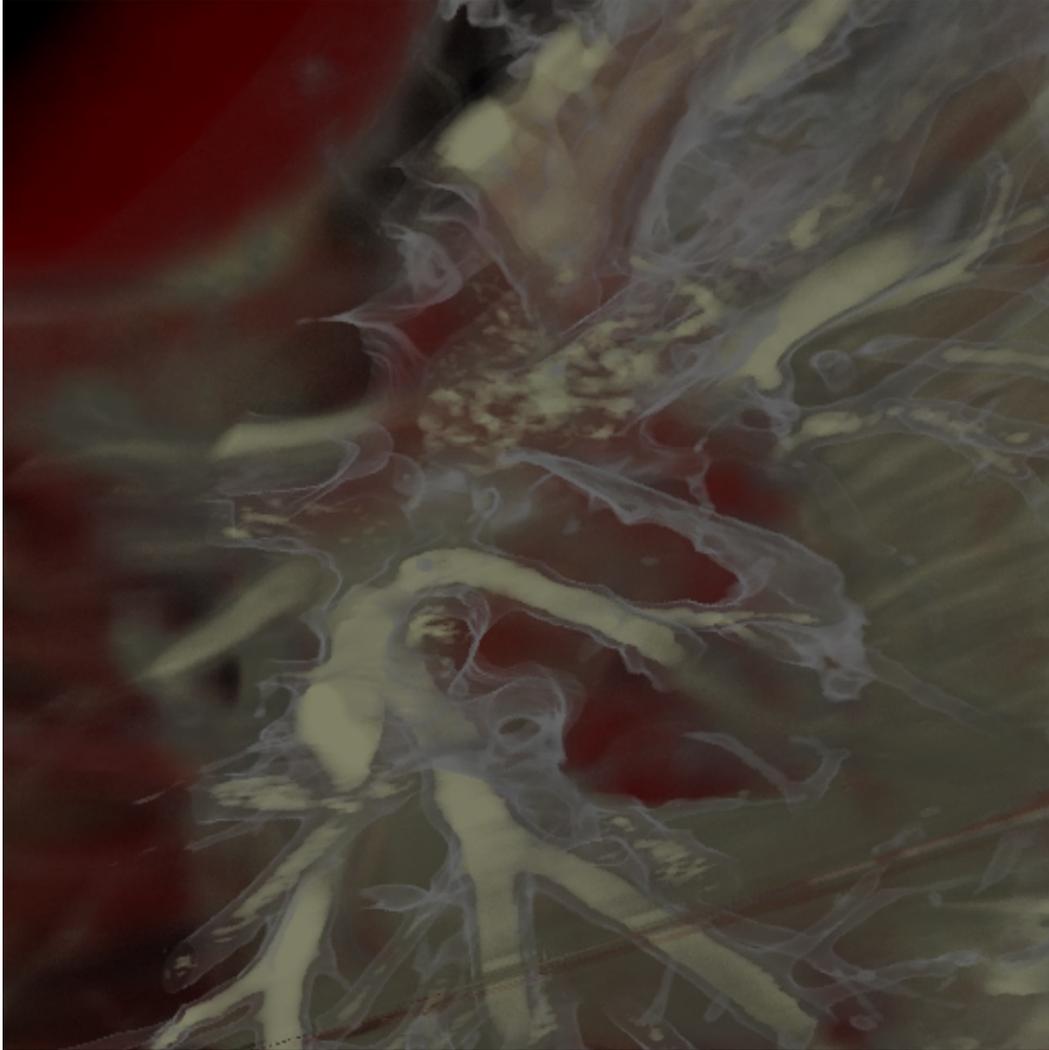


Figure 5.39: Pulmonary abnormality revealed by $\langle TFBB, DoF, NCC \rangle$.

results there is not a definitive yes or no answer to this question. In case of $\langle TFBB, DoF, NCC \rangle$ where no special chroma-coding is used, number of correct assessments, number of false negatives, time to assess, and time of correct assessment are significantly better than the reference. However, number of false positives, and distance of marked location tests were significantly worse than the reference.

Figure 5.39 shows the pulmonary abnormality present in a patient's CT rendered with $\langle TFBB, DoF, NCC \rangle$. We chose to show pulmonary CT as an example since it is usually cluttered and harder to assess. Despite the fact

that TFBB and DoF successfully reveal the region of interest and remove both foreground and background distractions, it is harder to find the abnormality when compared to images rendered with FBCC such as Figure 5.37.

Although this method of visualization has its benefits, it is prone to creating positive uncertainty. That is, it causes normal images to be assessed as having abnormality. This is much safer than the opposite where the patients with abnormalities are assessed as normal. However, it does not mean too many false positives do not have its draw backs. It creates more traffic for hospitals, it wastes resources, and put patient through possibly receiving more radiation. Therefore, it is better to avoid using two blurring methods simultaneously where there is no chroma-coding present as the sole visualization method to rely on. It is, however, possible to take advantage of this method when it can provide additional insight alongside other methods.

5.3.5.5 $\langle TFBB, DoF, RFBO+ \rangle$

Analyzing the performance of $\langle TFBB, DoF, RFBO+ \rangle$ is not trivial. The reason is the fact that this visualization method despite creating more false positives compared to the reference, performs significantly better than the reference in many other aspects of clinical diagnosis. $\langle TFBB, DoF, RFBO+ \rangle$ significantly increases the number of correct assessments, and correct markings, while significantly reducing number of false negatives, combined missed abnormalities, time of assessment, and time of correct assessment.

Figures 5.40 and 5.41 are rendered using this visualization method. The former provides a rather unusual view of a pulmonary embolism. A top down view of the pulmonary arteries is generally very cluttered and hard to assess. However, TFBB, DoF, and RFBO+ manage to eliminate a lot of clutter and provide a useful context and a clear view or the embolism. Figure 5.41 also

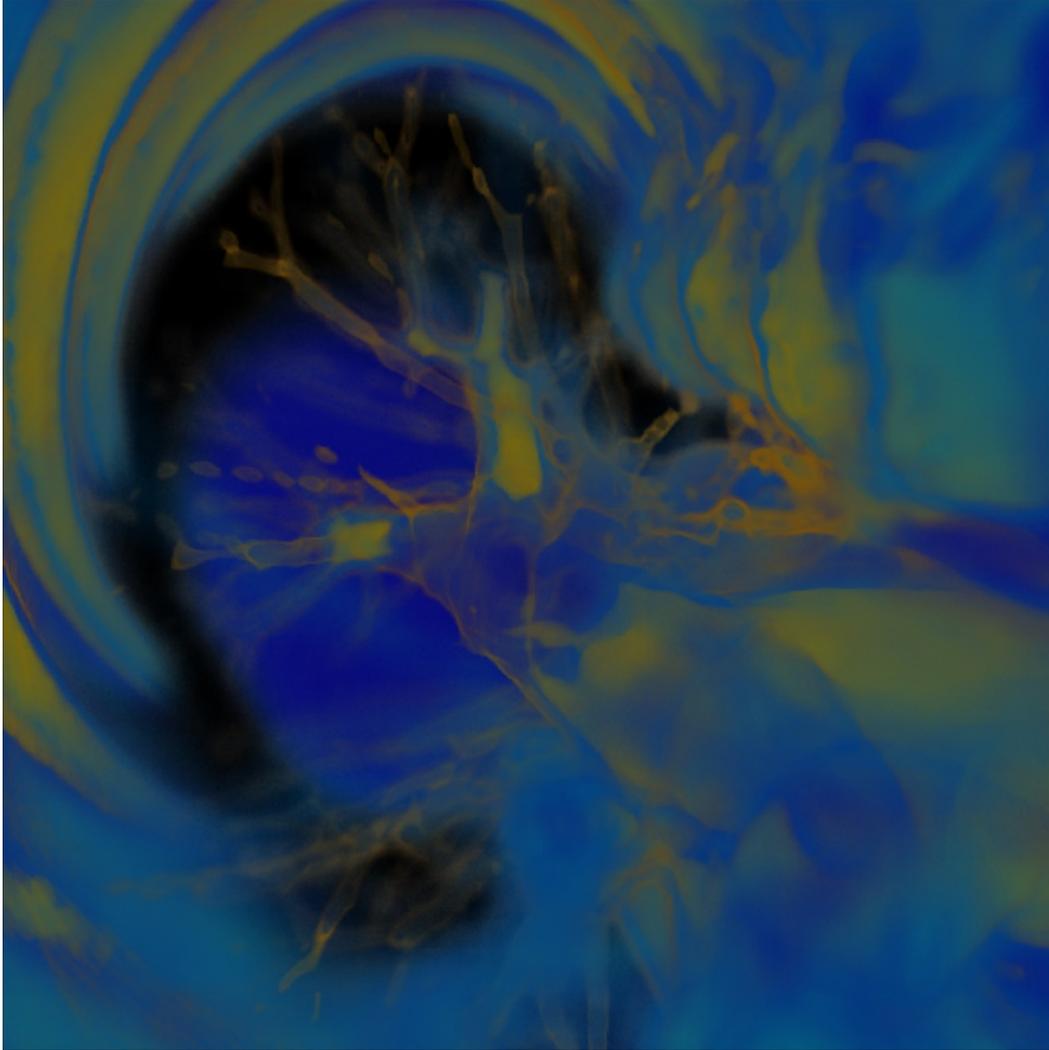


Figure 5.40: Pulmonary embolism revealed by $\langle TFBB, DoF, RFBO+ \rangle$ even from the extremely angled camera positions.

reveals the abnormality clearly in a cluttered view.

Considering the fact that the only difference between $\langle TFBB, DoF, RFBO+ \rangle$ and $\langle TFBB, RFBO+ \rangle$ is the addition of DoF we can conclude that DoF when combined with TFBB and RFBO+ can potentially create visual artifacts that physicians might mistakenly consider abnormalities. Blur discrimination, perception, and depth ordering has been researched by Mather *et al.* [35]. They state that region blur is more effective than border blur for human perception. Moreover, they show that human visual system

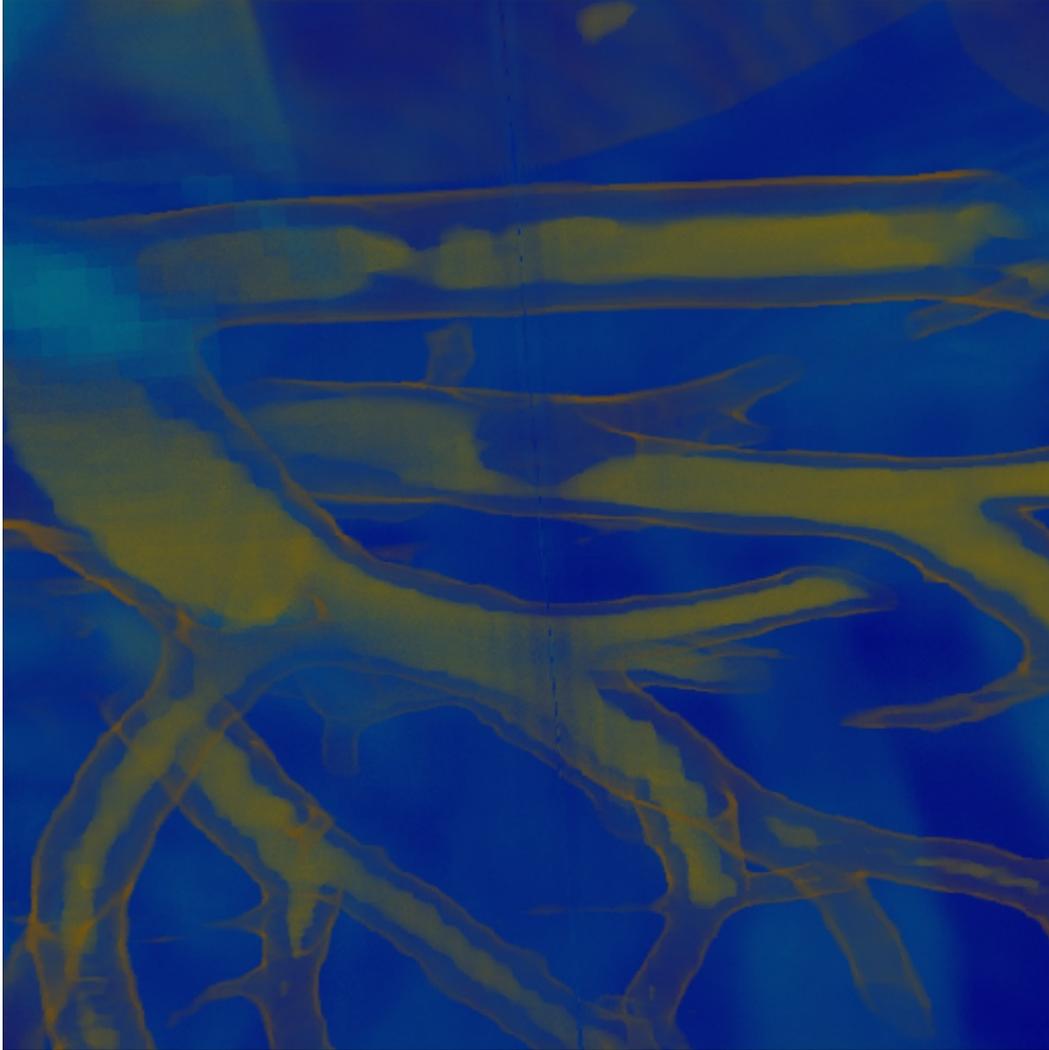


Figure 5.41: Large thrombus revealed by $\langle TFBB, DoF, RFBO+ \rangle$.

has a limited ability to discriminate different levels of blur.

It is probable that physicians are misled by the borders of blurred region created by DoF. These areas might potentially create the illusion of embolisms or thrombosed arteries if the physician is not experienced with images rendered with depth-of-field. It is possible that the point, where an artery that is extended in depth leaves the focused region and enters the blurred region, is mistakenly considered an abnormality by some testers who were not trained with this special visualization technique.

We believe the more plausible factor contributing to this problem is the fact

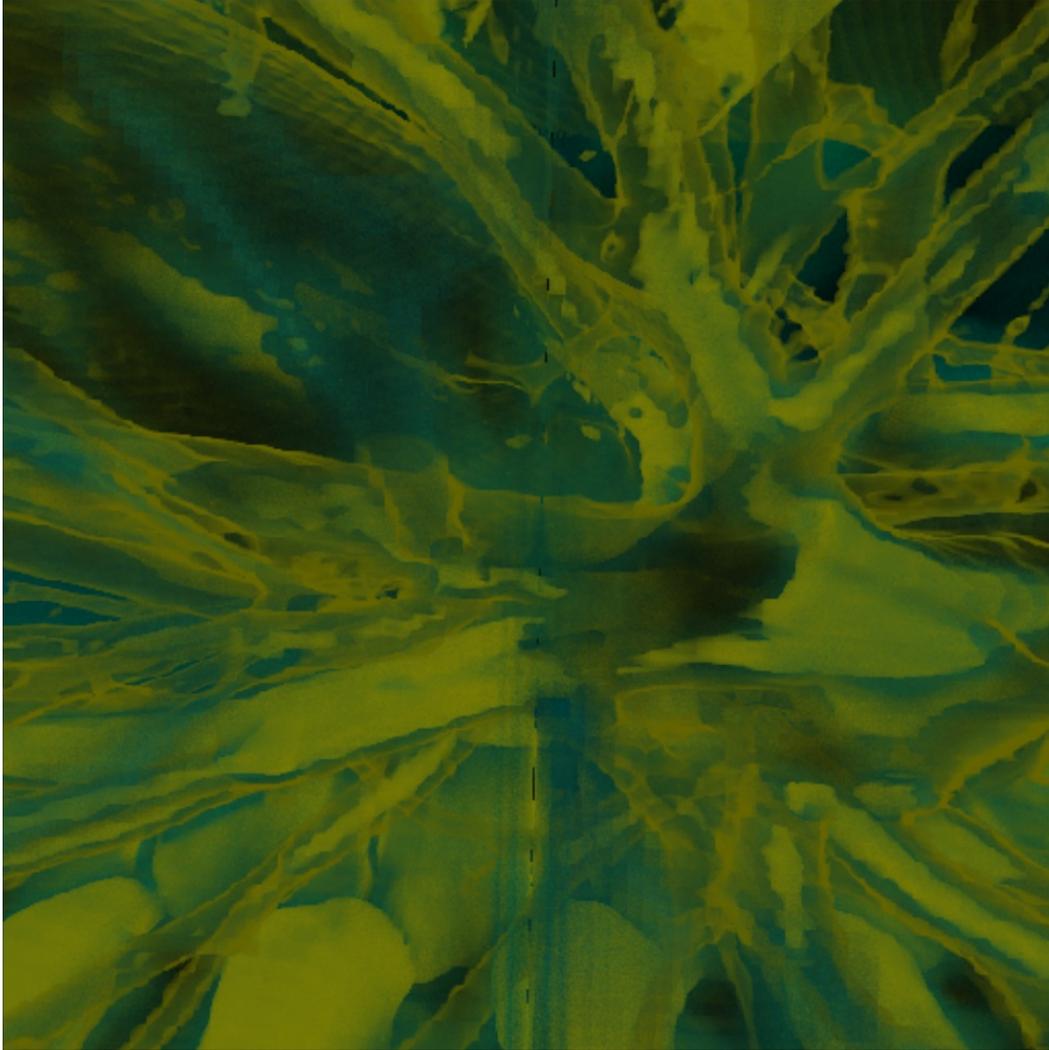


Figure 5.42: Pulmonary embolism revealed by $\langle TFBB, DoF, YFCO \rangle$. Even in the most cluttered views it is possible to detect abnormalities.

that our study is being done with still images and not an interactive real-time rendering interface. In our study participants cannot modify the focused region or lens parameters so they have no means of changing depth of focus to check the borders of depth-of-field blur. However, in reality our method provides real-time rendering so the user can change lens parameters and depth of focus. This will provide radiologists to easily distinguish an abnormality from the border of the blurred region and virtually eliminate such problems.

One of the pros of TFBB is that the transition between blurred and focused

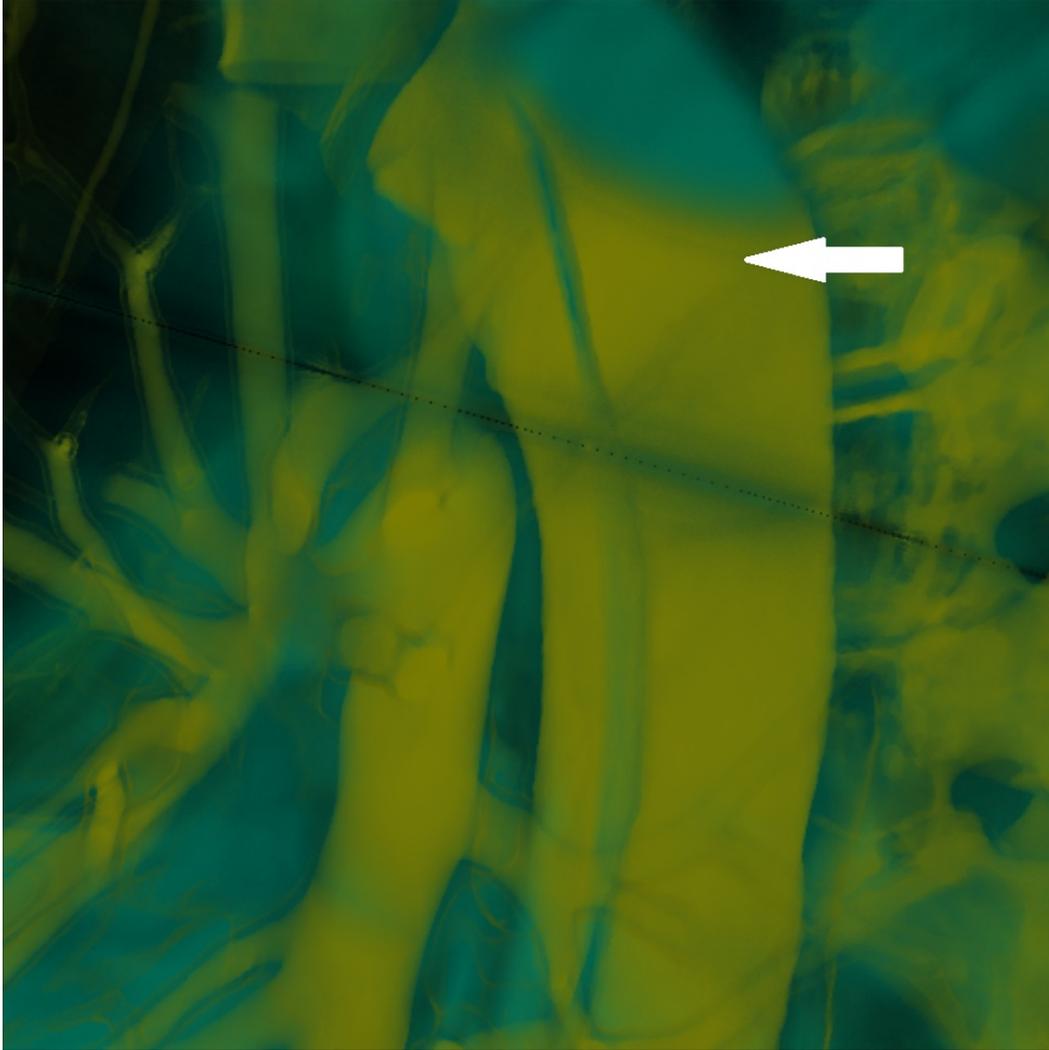


Figure 5.43: Aortic dissection revealed by $\langle TFBB, DoF, YFCO \rangle$.

areas does not exist when using only TFBB. Unlike DoF, TFBB does not create such artifacts since it keeps one or more materials entirely in focus and completely blurs other materials. Therefore, there will be no points where a single material such as the artery or the contrast agent transitions from being in focus to being out of focus. We recommend that both $\langle TFBB, DoF, RFBO+ \rangle$ and $\langle TFBB, RFBO+ \rangle$ to be used together for a more concrete assessment.

5.3.5.6 $\langle TFBB, DoF, YFCO \rangle$

$\langle TFBB, DoF, YFCO \rangle$ performs even better than $\langle TFBB, DoF, RFBO+ \rangle$ when compared to the reference. It significantly increases number of correct assessments, and number of correct markings, while it significantly reduces number of false negatives, combined missed abnormalities, time of assessment, and time of correct assessment. There is no significant difference between $\langle TFBB, DoF, YFCO \rangle$ and the reference in terms of false positives.

Something that is interesting to note here is the fact that $\langle TFBB, RFBO+ \rangle$ performed better than the $\langle TFBB, YFCO \rangle$. However, adding DoF resulted in $\langle TFBB, DoF, YFCO \rangle$ performing better than the $\langle TFBB, DoF, RFBO+ \rangle$ when compared to the reference. This can be caused by the effect color and its interplay with blur on human perception. This supports our suggestion that these modes of visualization need to be used alongside each other to be most effective.

Figures 5.42 and 5.43 show two images rendered using $\langle TFBB, DoF, YFCO \rangle$. As our first image we have chosen to show an image rendered from a very cluttered and hard to assess view of a pulmonary embolism. The dark region to the left of the center of Figure 5.42 clearly shows the embolus vessel. It is worth noting that detecting such an abnormality using the reference is almost impossible. It is also very challenging for such an abnormality to be assessed using traditional 2D slices of CT volume.

Figure 5.43 clearly shows an aortic dissection from the CT data from a patient. We can see in Figure 4.4a the same data rendered using the reference where the abnormality is virtually invisible.

5.3.5.7 Overall Comparison

All of the six methods that we compared against the reference significantly enhanced the average number of correct assessments. This is one of the impor-

tant aspects of clinical diagnosis that we wished to enhance with our proposed methods. Moreover, all of our visualization methods significantly reduced the number of false negative assessments compared to the reference. This aspect of clinical diagnosis, as discussed previously, is of utter importance. False negatives directly put the patient's health at risk.

In addition to the previous two enhancements, all of our visualization methods significantly enhanced the time of assessment. This result combined with the previous two results means that our methods manage to successfully create a positive confidence in decision making. This confidence is the result of less cluttered and clearer views. Four of our methods significantly enhanced the number of correct markings as well as reducing the number of combined missed abnormalities while there was no significant difference between the other two and the rest of our methods as well as the reference.

5.4 Conclusion

In this chapter, we presented our study. In this study we gathered more evidence to support our claim that TFBB, FBCC, their variations, and their combinations with each other as well as DoF can improve many aspects of clinical diagnosis. We have discussed the design, limitations, and achievements of our study. Based on our results we can confidently say that our proposed methods have a positive visible impact on clinical diagnosis using diagnostic imaging.

Our study can be used as a preliminary step towards more in depth analysis of our suggested visualization techniques and more sophisticated studies to be conducted in future. We hope that this will pave the way for better visualization, improved patient diagnosis, and consequently enhanced patient treatment and quality of life.

Chapter 6

Conclusion

Medical visualization is an integral part of every day clinical diagnosis. Many researchers have proposed methods to improve the information that is conveyed to the physician through this vital tool. We have argued throughout this thesis that there is still plenty of room for improvement.

The first problem that we have tackled in this work was the lack of a robust and flexible real-time algorithm to create depth-of-field in direct volume rendering. As discussed in Chapter 2 and Chapter 3, depth-of-field is a natural phenomenon that our visual system rely on for understanding depth. This phenomenon is absent in direct volume rendering due to the simplifications that make real-time direct volume algorithms such as ray casting possible.

Our proposed algorithm created depth-of-field by putting a kernel on each sampling point on each ray in ray casting algorithm while a permutation-based partial sampling of the kernel creates the desired effect. In this way we have managed to create real-time depth-of-field while avoiding aliasing and maintaining acceptable quality.

The proposed blurring method proved to have higher potential than only creating depth-of-field effect. It can blur any portion of the volume in real-time. We have utilized this capability of our algorithm in conjunction with transfer functions to create transfer function based blurring (TFBB). This

new visualization method takes advantage of blur's ability to direct viewer's attention to the object of interest as discussed in Chapter 2, Chapter 4 and by Robert Kosara in semantic depth-of-field [18]. TFBB is compatible with all different methods used to create transfer functions.

TFBB provides physicians with a new mode of visualization that was not previously available. Using TFBB, it is possible to keep the material of interest in focus and blur all the material that we are not interested in. In this way a lot of unwanted clutter can be blurred and the physician can focus on the material of interest. Blurred material provides context and makes it easier for the physician to understand spatial orientation of what is being examined.

Color has also been used to direct viewer's attention. We have used color in conjunction with blurring to create focus-based chroma-coding FBCC. This new method can further help physicians by directing their attention, providing contrast, and neutralizing clutter in an image. Since FBCC relies on focus information, it needs to be used at the same time as blurring methods such as TFBB, DoF, or a combination of both. These combinations as well as different choices of color can create a huge number of different visualizations. We have presented a very small subset of these visualizations in this thesis.

In order to validate these newly created visualization methods we have conducted a usability study. The effect of our visualization methods on various aspects of clinical diagnosis has been isolated and studied. Based on our results we concluded that our newly created methods significantly improve several aspects of clinical diagnosis. We have also discussed the limitations of our study and concluded that a more comprehensive study in future can provide a solid proof so our proposed visualization methods can be used commercially.

Bibliography

- [1] Brian A. Barsky and Todd J. Kosloff. Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers*, ICCOMP'08, pages 999–1010, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).
- [2] Christian Boucheny, Georges-Pierre Bonneau, Jacques Droulez, Guillaume Thibault, and Stephane Ploix. A perceptive evaluation of volume rendering techniques. *ACM Trans. Appl. Percept.*, 5(4):23:1–23:24, feb 2009.
- [3] Stefan Bruckner and Eduard Gröller. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1344–1351, nov 2007.
- [4] Kenneth J. Ciuffreda, Bin Wang, and Balamurali Vasudevan. Conceptual model of human blur perception. *Vision Research*, 47(9):1245 – 1252, 2007.
- [5] Andy Cockburn, Amy Karlson, and Benjamin B Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2, 2008.
- [6] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, jan 1986.
- [7] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 137–145, New York, NY, USA, 1984. ACM.
- [8] Joe Demers. *Depth of Field: A Survey of Techniques*. GPU Gems. Addison-Wesley Professional, 2004.
- [9] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '88, pages 65–74, New York, NY, USA, 1988. ACM.
- [10] Gideon Ehrlich. Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. ACM*, 20(3):500–513, jul 1973.

- [11] R. A. Fisher. Xv.the correlation between relatives on the supposition of mendelian inheritance. *Transactions of the Royal Society of Edinburgh*, 52:399–433, 1 1919.
- [12] Enrico Gobbetti, Fabio Marton, and José Antonio Iglesias Guitián. A single-pass GPU ray casting framework for interactive out-of-core rendering of massive volumetric datasets. *Vis. Comput.*, 24(7):797–806, jul 2008.
- [13] Pascal Grosset, Mathias Schott, Georges-Pierre Bonneau, and Hansen Charles. Evaluation of depth of field for depth perception in DVR. In *IEEE Pacific Visualization 2013*, Sidney, Australia, 2013.
- [14] Markus Hadwiger, Joe M. Kniss, Christof Rezk-salama, Daniel Weiskopf, and Klaus Engel. *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006.
- [15] Robert T. Held, Emily A. Cooper, James F. O’Brien, and Martin S. Banks. Using blur to affect perceived distance and size. *ACM Trans. Graph.*, 29(2):19:1–19:16, apr 2010.
- [16] Christopher Johnson and Charles Hansen. *Visualization Handbook*. Academic Press, Inc., Orlando, FL, USA, 2004.
- [17] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *In IEEE Symposium on Volume Visualization*, pages 79–86, 1998.
- [18] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS’01)*, INFOVIS ’01, pages 97–, Washington, DC, USA, 2001. IEEE Computer Society.
- [19] Robert Kosara, Silvia Miksch, Helwig Hauser, Johann Schrammel, Verena Giller, and Manfred Tscheligi. Useful properties of semantic depth of field for better f+c visualization. In *Proceedings of the Symposium on Data Visualisation 2002, VISSYM ’02*, pages 205–210, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [20] Martin Kraus and Magnus Strengert. Depth-of-field rendering by pyramidal image processing. In *Proceedings Eurographics 2007*, 2007.
- [21] Kevin Kreeger and Arie Kaufman. Interactive volume segmentation with the pavlov architecture. In *Proceedings of the 1999 IEEE symposium on Parallel visualization and graphics, PVGS ’99*, pages 61–68, Washington, DC, USA, 1999. IEEE Computer Society.
- [22] J. Krivanek, J. Zara, and K. Bouatouch. Fast depth of field rendering with surface splatting. In *Computer Graphics International, 2003. Proceedings*, pages 196–201, 2003.
- [23] J. Kruger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS’03)*, VIS ’03, pages 38–, Washington, DC, USA, 2003. IEEE Computer Society.

- [24] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 451–458, New York, NY, USA, 1994. ACM.
- [25] Sarang Lakare and Arie Kaufman. Light weight space leaping using ray coherence. In *Proceedings of the conference on Visualization '04*, VIS '04, pages 19–26, Washington, DC, USA, 2004. IEEE Computer Society.
- [26] Byeonghun Lee, Jihye Yun, Jinwook Seo, Byonghyo Shim, Yeong-Gil Shin, and Bohyoung Kim. Fast high-quality volume ray casting with virtual samplings. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1525–1532, nov 2010.
- [27] Kefei Lei and John F. Hughes. Approximate depth of field effects using few samples per pixel. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '13, pages 119–128, New York, NY, USA, 2013. ACM.
- [28] Marc Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, jul 1990.
- [29] Wei Li and Arie E. Kaufman. Texture partitioning and packing for accelerating texture-based volume rendering. In *Graphics Interface*, pages 81–88, 2003.
- [30] Wei Li, Klaus Mueller, and Arie Kaufman. Empty space skipping and occlusion clipping for texture-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 42–, Washington, DC, USA, 2003. IEEE Computer Society.
- [31] H. Lieberman. A multi-scale, multi-layer, translucent virtual space. In *Information Visualization, 1997. Proceedings., 1997 IEEE Conference on*, pages 124–131, Aug 1997.
- [32] Huei-Yung Lin and Kai-Da Gu. Photo-realistic depth-of-field effects synthesis based on real camera parameters. In *Proceedings of the 3rd international conference on Advances in visual computing - Volume Part I*, ISVC'07, pages 298–309, Berlin, Heidelberg, 2007. Springer-Verlag.
- [33] Nikolai L. Manev. Sequences generating permutations. *Applied Mathematics and Computation*, 216(3):708 – 718, 2010.
- [34] George Marsaglia and Arif Zaman. Some portable very-long-period random number generators. *Comput. Phys.*, 8(1):117–121, jan 1994.
- [35] George Mather and David RR Smith. Blur discrimination and its relation to blur-mediated depth perception. *PERCEPTION-LONDON-*, 31(10):1211–1220, 2002.
- [36] Michael Meißner, Michael C. Doggett, Johannes Hirche, Urs Kanus, and Wolfgang Straßer. Efficient space leaping for raycasting architectures. In *Volume Graphics'01*, 2001.

- [37] Klaus Mueller and Roger Crawfis. Eliminating popping artifacts in sheet buffer-based splatting. In *Proceedings of the conference on Visualization '98, VIS '98*, pages 239–245, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
- [38] Manjushree Nulkar and Klaus Mueller. Splatting with shadows. In *Proceedings of the 2001 Eurographics conference on Volume Graphics, VG'01*, pages 35–50, Aire-la-Ville, Switzerland, Switzerland, 2001. Eurographics Association.
- [39] R. Osborne, H. Pfister, H. Lauer, N. McKenzie, S. Gibson, W. Hiatt, and T. Ohkami. Em-cube: An architecture for low-cost real-time volume rendering. pages 131–138, 08/1997 1997.
- [40] OsiriX. OsiriX DICOM Repository. <http://www.osirix-viewer.com/datasets/>. Accessed: 2013-07-31.
- [41] N. Pears, Y. Liu, and P. Bunting. *3D Imaging, Analysis and Applications*. SpringerLink : Bücher. Springer London, 2012.
- [42] Vladimir Pekar, Rafael Wiemker, and Daniel Hempel. Fast detection of meaningful isosurfaces for volume data visualization. In Thomas Ertl, Kenneth I. Joy, and Amitabh Varshney, editors, *IEEE Visualization*. IEEE Computer Society.
- [43] Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler. The volumepro real-time ray-casting system. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*, pages 251–260, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [44] Przemyslaw Rokita. Generating depth-of-field effects in virtual reality applications. *IEEE Comput. Graph. Appl.*, 16(2):18–21, mar 1996.
- [45] Timo Ropinski, Frank Steinicke, and Klaus Hinrichs. *Visually Supporting Depth Perception in Angiography Imaging*, volume 4073 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006.
- [46] M. Schott, A.V.P. Grosset, T. Martin, V. Pegoraro, S.T. Smith, and C.D. Hansen. Depth of field effects for interactive direct volume rendering. In H. Pfister H. Hauser and J. J. van Wijk, editors, *Proceedings of Eurographics/IEEE Symposium on Visualization 2011*, volume 30, 2011.
- [47] Mathias Schott, Vincent Pegoraro, Charles D. Hansen, Kevin Boulanger, and Kadi Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Comput. Graph. Forum*, 28(3):855–862, 2009.
- [48] J. P. Schulze, M. Kraus, U. Lang, and T. Ertl. Integrating pre-integration into the shear-warp algorithm. In *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics, VG '03*, pages 109–118, New York, NY, USA, 2003. ACM.
- [49] Robert Sedgewick. Permutation generation methods. *ACM Comput. Surv.*, 9(2):137–164, jun 1977.

- [50] N. Shareef, D. L. Wang, and D. L. Yagel. Segmentation of medical images using LEGION. *IEEE Trans. Medical Imaging*, 18(1):74–91, jan 1999.
- [51] AmirAli Sharifi and Pierre Boulanger. Using stochastic sampling to create depth-of-field effect in real-time direct volume rendering. In *Proceedings of the 2014 Graphics Interface Conference, GI '14*, pages 77–85, Toronto, Ont., Canada, Canada, 2014. Canadian Information Processing Society.
- [52] AmirAli Sharifi and Pierre Boulanger. Enhancing visual perception and directing viewer’s attention in interactive direct volume rendering. In *Information Visualisation (iV), 2015 19th International Conference on*, pages 536–542, July 2015.
- [53] Irvin Sobel. Neighborhood coding of binary images for fast contour following and general binary array processing. *Computer Graphics and Image Processing*, 8(1):127–135, 1978.
- [54] Richard A Steenblik. The chromostereoscopic process: A novel single image stereoscopic process. In *OE LASE'87 and EO Imaging Symp (January 1987, Los Angeles)*, pages 27–34. International Society for Optics and Photonics, 1987.
- [55] Yubo Tao, Hai Lin, Feng Dong, and Gordon Clapworthy. 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*, pages 418–422, Washington, DC, USA, 2011. IEEE Computer Society.
- [56] Yubo Tao, Hai Lin, Feng Dong, and Gordon Clapworthy. 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*, pages 418–422, Washington, DC, USA, 2011. IEEE Computer Society.
- [57] Takashi Totsuka and Marc Levoy. Frequency domain volume rendering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques, SIGGRAPH '93*, pages 271–278, New York, NY, USA, 1993. ACM.
- [58] I. Viola, M. Feixas, M. Sbert, and M.E. Groller. Importance-driven focus of attention. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):933–940, Sept 2006.
- [59] Ming Wan, Steve Bryson, and Arie Kaufman. Boundary cell-based acceleration for volume ray casting. *Computers & Graphics*, 22(6):715–722, 1998.
- [60] Leonard C. Wanger, James A. Ferwerda, and Donald P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Comput. Graph. Appl.*, 12(3):44–51, 54–58, may 1992.
- [61] Lee Alan Westover. Interactive volume rendering. In *Proceedings of the 1989 Chapel Hill workshop on Volume visualization, VVS '89*, pages 9–16, New York, NY, USA, 1989. ACM.

- [62] Lee Alan Westover. Footprint evaluation for volume rendering. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '90, pages 367–376, New York, NY, USA, 1990. ACM.
- [63] Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. PhD thesis, Chapel Hill, NC, USA, 1991. UMI Order No. GAX92-08005.
- [64] Craig M. Wittenbrink, Thomas Malzbender, and Michael E. Goss. Opacity-weighted color interpolation, for volume sampling. In *Proceedings of the 1998 IEEE symposium on Volume visualization*, VVS '98, pages 135–142, New York, NY, USA, 1998. ACM.
- [65] S.E. Wixson. The display of 3d mri data with non-linear focal depth cues. In *Computers in Cardiology 1990, Proceedings.*, pages 379–380, Sep 1990.
- [66] R. Yagel and Z. Shi. Accelerating volume animation by space-leaping. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pages 62–69, oct 1993.
- [67] Xuan Yu, Rui Wang, and Jingyi Yu. Real-time depth of field rendering via dynamic light field generation and filtering. *Computer Graphics Forum*, 29(7):2099–2107, 2010.
- [68] Caixia Zhang and Roger Crawfis. Volumetric shadows using splatting. In *Proceedings of the conference on Visualization '02*, VIS '02, pages 85–92, Washington, DC, USA, 2002. IEEE Computer Society.
- [69] Lin Zheng, Yingcai Wu, and Kwan-Liu Ma. Perceptually-based depth-ordering enhancement for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):446–459, mar 2013.
- [70] Steven W. Zucker and Robert A. Hummel. A three-dimensional edge operator. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (3):324–331, 1981.

Appendix A

Usability Study Consent Form

In this appendix we have included the information sheet and the consent form given to the participants to read and sign before the study.

PARTICIPANT INFORMATION SHEET

Study title:

Assessment of newly developed synthetic visual cues on visual perception in CT.

Principal Investigator: AmirAli Sharifi
PhD Candidate, Department of Computing Science
University of Alberta
Telephone: 780 953-6261
Email: asharifi@ualberta.ca

BACKGROUND AND PURPOSE:

Medical imaging methods such as CT-scan or MRI provide physicians with underlying anatomy and structure of the patient. Traditionally the result of such imaging techniques were visualized as a set of 2D slices. Clinical diagnosis based on such images required many years of training and it was prone to errors. Direct Volume Rendering (DVR) was later used to visualize the data acquired from imaging devices such as CT in 3D. Although DVR enhances visualization, the result of 3D rendering is still a 2D image projected on 2D screen such as a monitor screen. Consequently, correct and effective visual perception still remains a challenge to this day.

Our proposed visualization techniques are enhancements to currently used visualization methods. They aim to provide the user of the system with a less cluttered and more informative visualization of the medical imaging data by using blurring and coloring parts of the image.

Objectives:

1. To determine if our method can reduce the time of detection of the defect by the specialist.
2. To determine if our method can reduce the error in diagnosis.

PROCEDURES:

If you choose to participate in this research study, you will be viewing 3D renderings of CT images of various patients' cardiovascular system in random order, some of which are rendered with our visual perception enhancements. You will view these images on a single 2D screen (computer monitor). You will be asked to indicate whether or not you see a vascular abnormality in each image and if your answer is positive you will be asked to mark the center of the of the disease area on the image using a standard computer mouse. You will be asked to fill a short

Version 1: January 2015

questionnaire after you have finished the session. The session should take around 60 minutes in total. For each image you have a limited time and you can rest between images.

BENEFITS: There is no benefit to you from participating in this study.

RISKS: There is no risk to you from participating in this study.

CONFIDENTIALITY:

The questionnaire data and your interaction data collected from you during this study will not identify you by name when we analyze the data. Only a coded number will be used. Your name will not be disclosed outside the research clinic, unless required by law. Any report published as a result of this study will not identify you by name. No demographic information other than your occupation will be collected.

By participating in this study, you are giving permission for the study investigator to administer the questionnaire to you and analyze the results. After the study is done, we will still securely store your research data for 5 years before destroying.

VOLUNTARY PARTICIPATION AND FREEDOM TO WITHDRAW:

Participation in the study is voluntary. If you choose not to participate, there will be no consequences. If you choose to participate in the study, you can withdraw from the study at any time. You do not have to give a reason for withdrawing. If you withdraw from the study, we will also remove your data.

COSTS:

There is no cost to you.

ADDITIONAL CONTACTS:

If you have any concerns about any aspect of this study you may contact the Ethics Office at 780-492-2615. This office has no affiliation with study investigators.

If you have any questions or concerns about the study activities please contact.

AmirAli Sharifi, principal investigator	Phone 780 953-6261
Michelle Noga, supervisor	Pager 780-445-1999

Consent Form

Study title: Assessment of newly developed synthetic visual cues on visual perception in CT.

Investigator: AmirAli Sharifi
Telephone: 780 953-6261

Do you understand that you are participating in a research study? Yes No

Have you read and received a copy of the attached Information Sheet? Yes No

Do you understand the benefits and risks involved in taking part in this research study? Yes No

Did you have an opportunity to ask questions and discuss this study? Yes No

Do you understand that you are free to refuse to have your participation in the study or withdraw from the study at any time? You do not have to give a reason. Yes No

Has the issue of confidentiality been explained to you? Yes No

Do you understand who will have access to the information you provide? Yes No

This study was explained to me by: _____

I agree to take part in this study. Yes No

Signature

Date

Witness

Printed Name

Printed Name

I believe that the person signing this form understands what is involved in the study and voluntarily agrees to participate.

Signature of Investigator or Designee

Date

Appendix B

Test Taker Instructions

In this appendix we have included the instructions given to the participants before the study. This sheet provided a brief overview of the rendering and visualization techniques used in the study. This instructions were given to the participant a few days before the study. It was also available during the actual study so the participant could consult it whenever they needed to.

Dear Volunteer,

Thank you for choosing to participate in this usability study. This sheet includes information about the study and what to expect. Please read this carefully and completely before starting the test.

Primary and Secondary Questions:

There are 140 images in this study. Each image may or may not contain an abnormality. The primary question for an image will ask you about the presence or absence of a specific abnormality. If your answer to the primary question is positive then a second question will ask you to mark the abnormality by a single left-click of the mouse. If the answer to your primary question is negative then you go to the next image.

Note: if you initially answered positively to primary question for an image and while looking to mark the abnormality changed your mind, you can still press “no” and your answer will be changed to “no” and you will proceed to next image. You can do this before you run out of time for the image.

Please read all questions carefully as they differ from one image to another.

Timer:

The primary question for the image is shown before you see the image and it is not timed. As soon as you press the “show image” button the image is shown and the 15 second timer for the current image will start and it ends either by running out of time or by you completely answering the questions for the current image. There is no timer between images so you can take as much rest as you want between images. After you finish with an image, the primary question for the next image is shown.

Consent Form:

Please read and sign the consent form.

Rendering Artifacts:

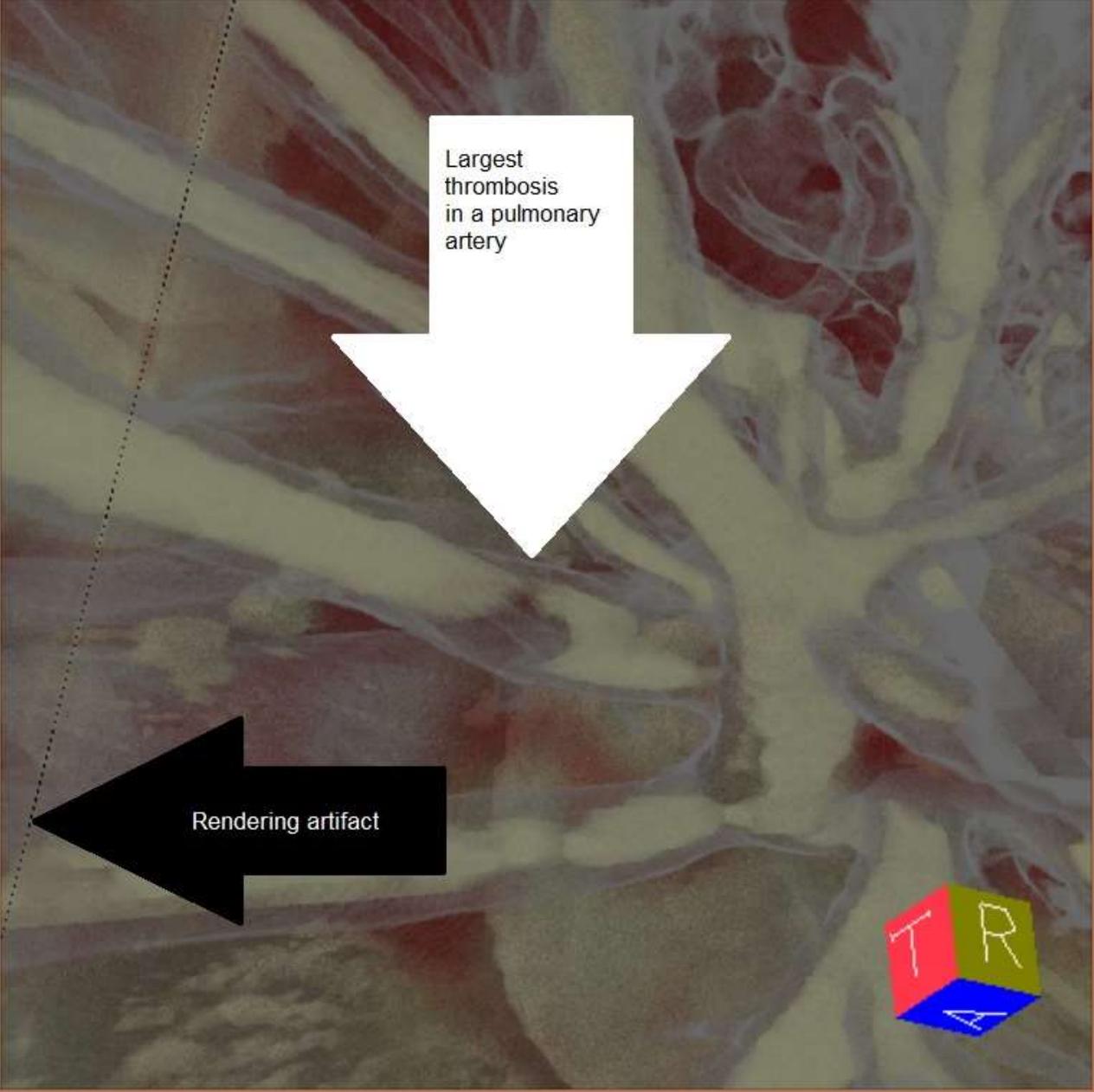
There is a line that appears in some of the image and it is a rendering artefact so please ignore it. The following pages present some examples of such rendering artifacts.

Color Modes:

There are three different color modes used to produce images in the study:

- No color coding
- Yellow-Cyan
- Yellow-Blue

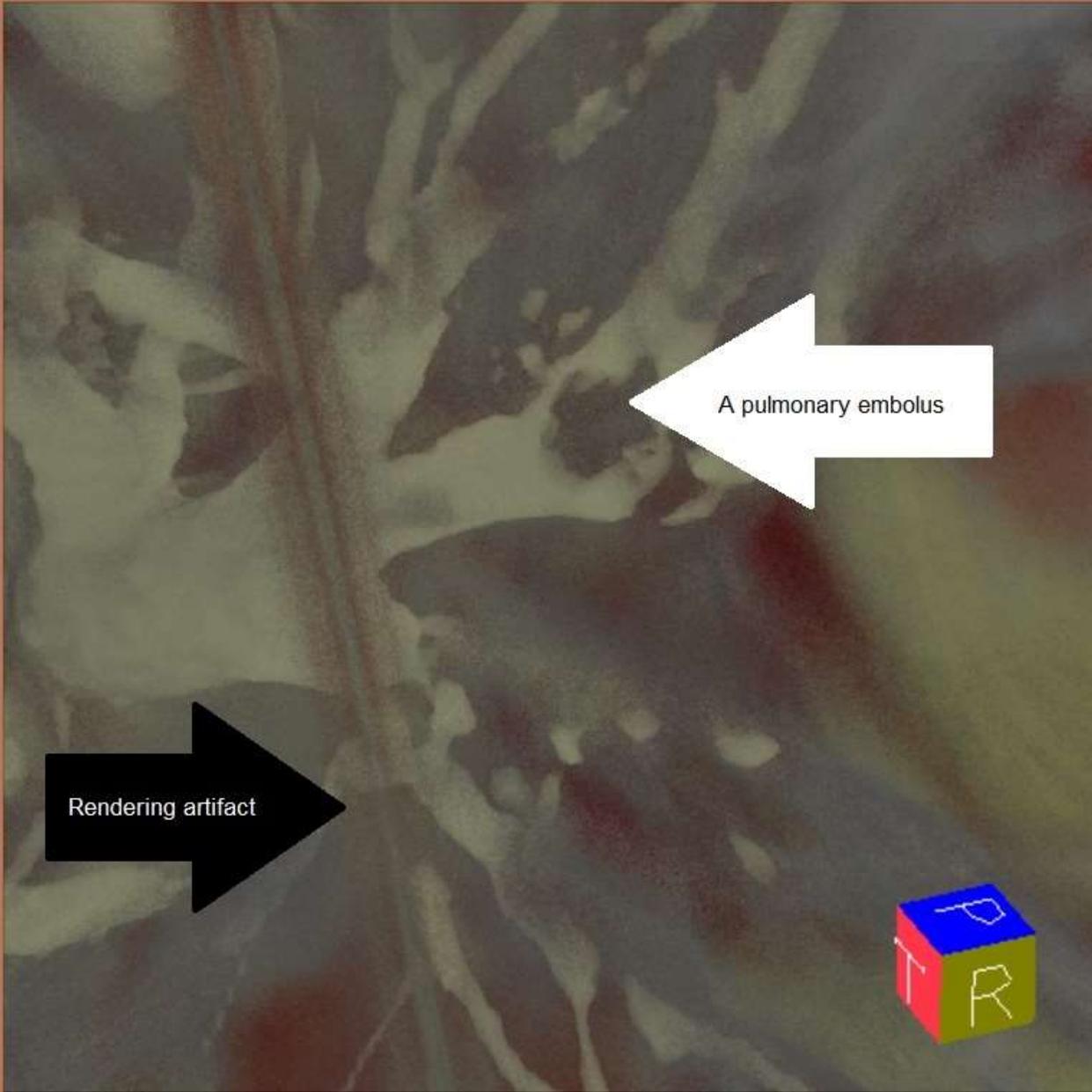
In the no color coding mode the colors will resemble natural colors of tissues. In the other two color coding modes yellow is the color of focus and cyan or blue are the color of out of focus areas. Some examples of the images using different color coding modes with different abnormalities are provided below.



Largest thrombosis in a pulmonary artery

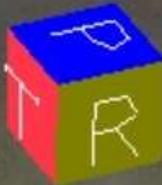
Rendering artifact

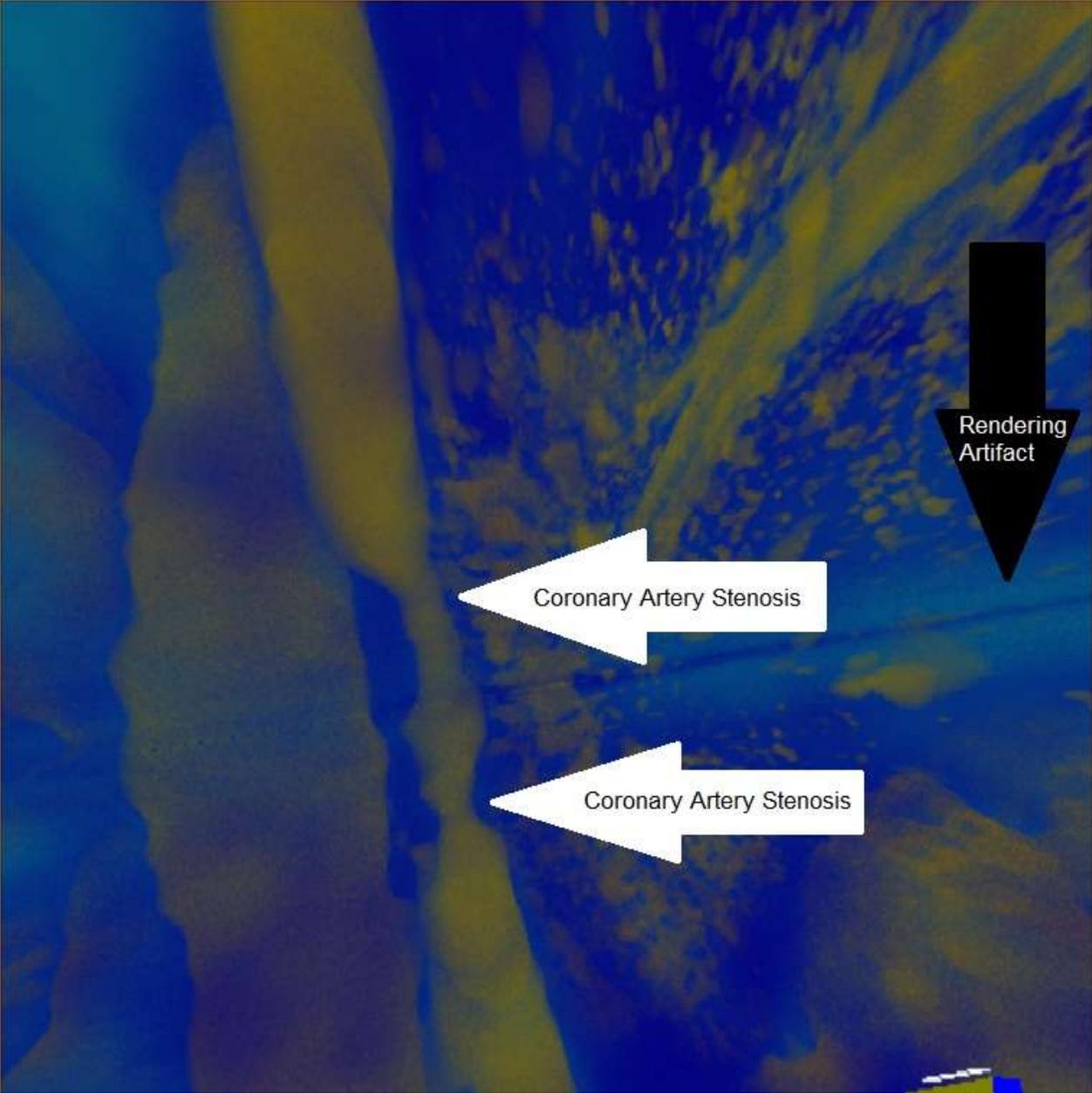




A pulmonary embolus

Rendering artifact

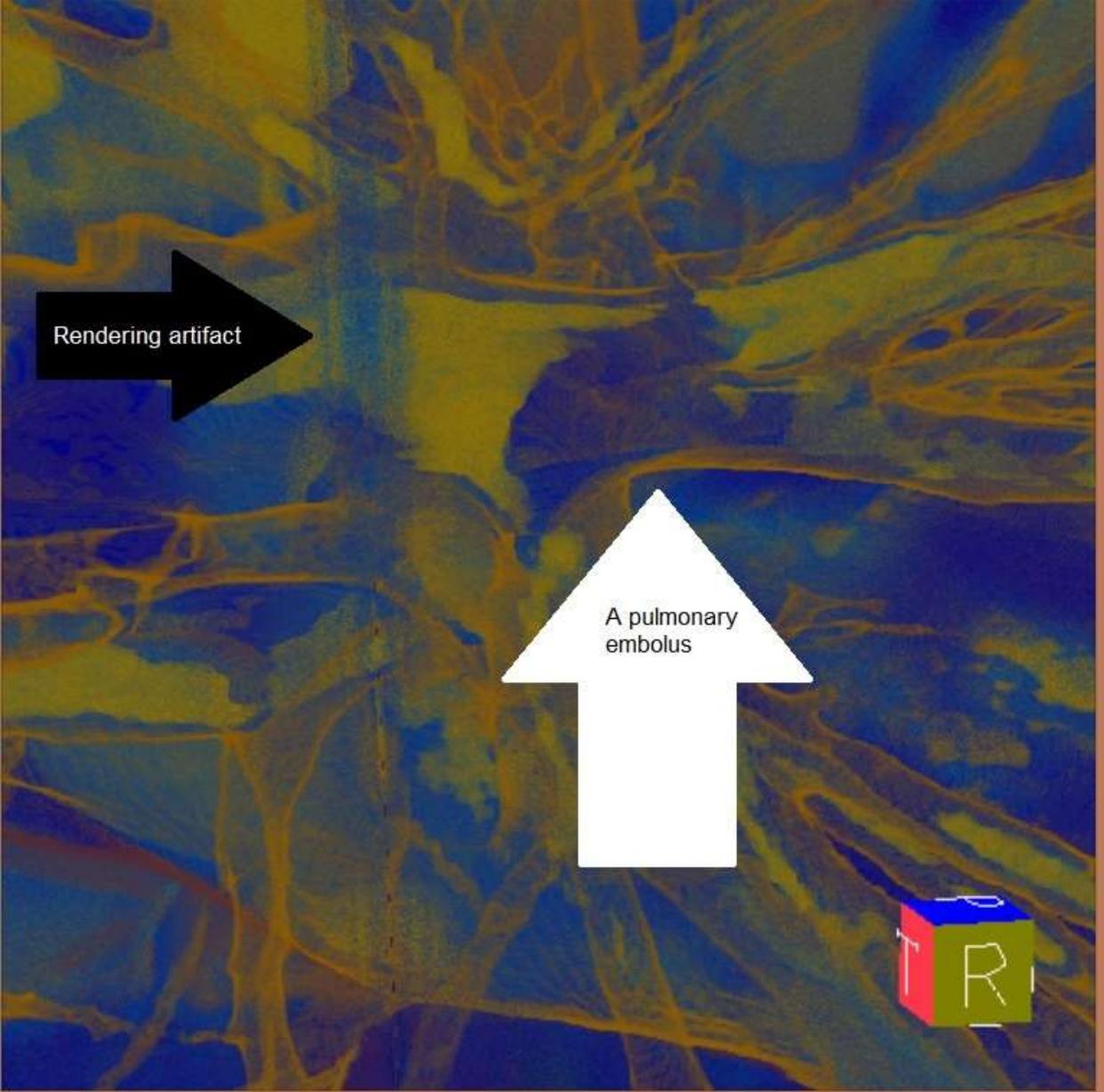




Coronary Artery Stenosis

Coronary Artery Stenosis

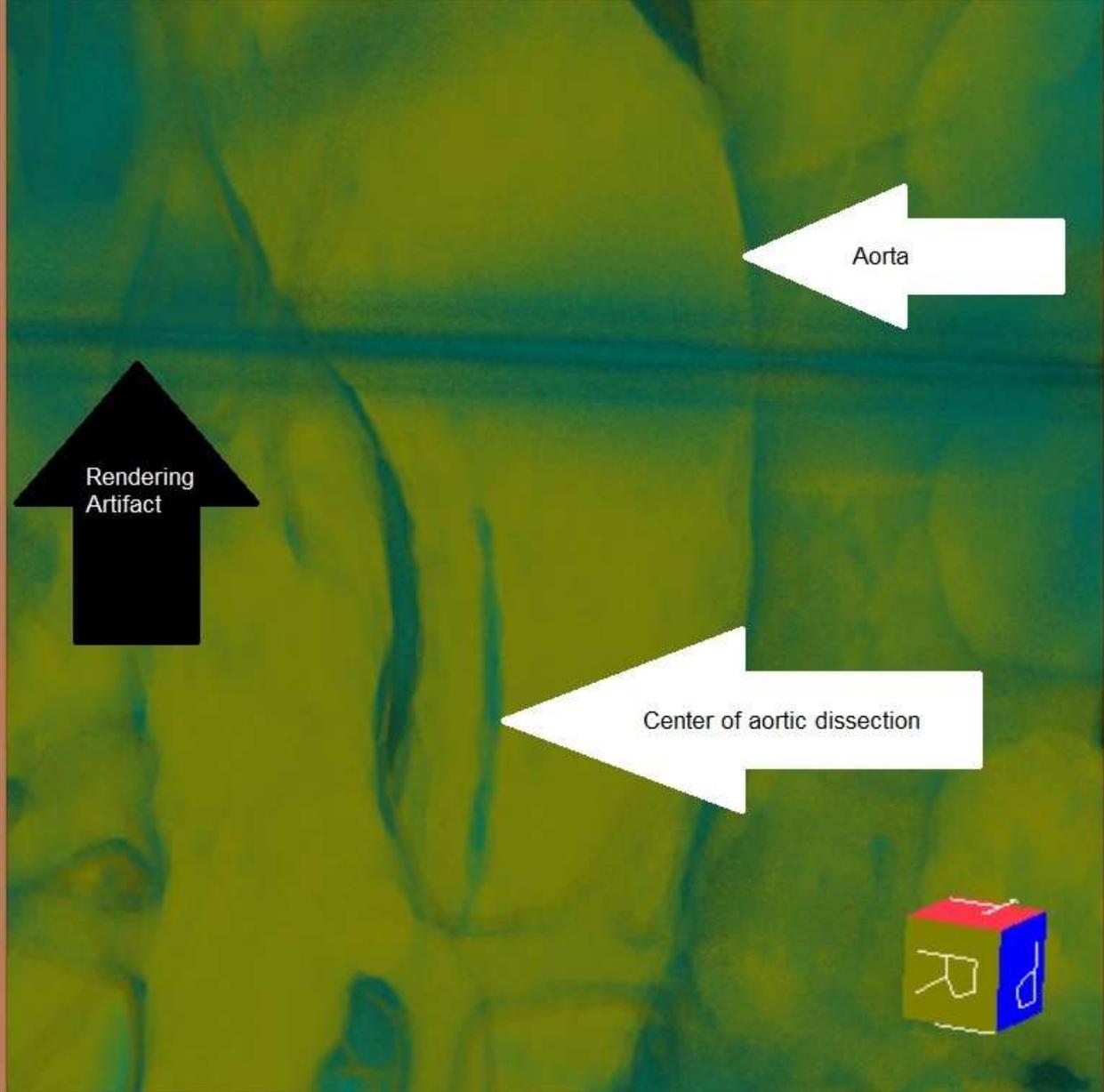
Rendering
Artifact



Rendering artifact

A pulmonary embolus





This is a normal image with an arrow indicating the artery in question.



Rendering artifact

