IMPORTANCE-DRIVEN TRANSFER FUNCTION DESIGN FOR VOLUME VISUALIZATION OF MEDICAL IMAGES

CAI LILE
B.Eng., University of Science and Technology of China

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Cai Lile
17 Nov. 2015

Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Ong Sim Heng, for letting me be one of his students, and guiding me along my Ph.D. journey. Both this thesis and my research publications would not have been possible without his guidance, patience and understanding.

I would also like to thank my co-supervisor, Prof. Chui Chee Kong, for his mentorship, advice and encouragement. Without his support, I would not have achieved so much in the road of pursuing my Ph.D..

My thanks also go to Nguyen Phu Binh, for providing his insightful comments and advice on my research work. I appreciate every inspiring discussion with you and feel lucky to have the opportunity to work with you.

Thanks to all group members, especially Tay Wei Liang, Wen Rong, Agus, Duan Bin, and Chen Xuan, for your friendship and assistance in the eye-tracking experiment. Special thanks to the lab manager, Francis, who has helped me a lot since I came to the Vision and Machine Learning Lab.

I wish to thank my dear friends, Liu Tianfei, Chen Bichao, Wang Yuxiang, Yuan Jun, Zhao Ke, and Shen Chengyao, for making my days in Singapore so colorful and unforgettable. I will always miss the times when we had meat together at YIH.

Last but not the least, I would like to thank my parents who raised me and always support me in all my pursuits.

Summary

Volume visualization aims to identify and visualize meaningful structures in a volumetric dataset. In direct volume rendering, this goal is achieved by a transfer function (TF), which "transfer" voxel properties (e.g., intensity, gradient) to optical properties (e.g., color, opacity). The non-intuitive nature of TF space combined with the extreme flexibility and lack of guidance and constraints makes the specification of the TF a highly challenging task in volume rendering. This motivates the extensive study of TF design. In this thesis, the TF design problem in volume visualization of medical images is investigated and several importance-driven TF design methods are proposed to better identify and highlight important structures in medical volumetric datasets.

First, an automatic 2-D TF design method based on the matching of visibility distributions is proposed. Instead of assigning opacity directly from voxel intensity and gradient magnitude values, the opacity TF is automatically derived by minimizing the Jensen-Shannon divergence between the observed visibility distribution and an importance-based target visibility distribution, which ensures important regions can achieve a predefined amount of visibility in the rendered image. When the method was tested on several medical volumetric datasets, the key structures within the volume were clearly visualized with minimal user intervention.

In the second work, the TF domain is extended to higher dimensions in order to perform more sophisticated classification of volumetric objects, and a two-level clustering approach is proposed to address the higher-dimensional TF design problem. Based on two-level clustering, a novel volume exploration scheme with top-down navigation is proposed. Interesting structures in the volume are discovered interactively by the user selecting clusters to visualize and modifying the clustering results when necessary. The proposed approach allows users to flexibly create a visualization by only rendering the structures deemed important.

The third work moves away from manipulating TFs in the TF space and proposes a rule-enhanced TF design method that utilizes a small amount of segmentation data labelled in the spatial domain. This is motivated by the intuitive idea that the segmentation data of the user-specified target tissue contains information to characterize the target tissue, and this information can be encoded by a set of rules. The rule-enhanced TFs are able to separate and highlight important

volumetric objects in challenging datasets, which would otherwise be difficult to achieve by traditional methods.

Finally, an eye tracking experiment is conducted to study visual attention deployment in volume visualization. The visualization images generated by the above three methods as well as various state-of-the-art methods are used as stimuli and the eye movement data of participants free-viewing these images are recorded. This allows for the investigation of how visual attention is deployed in different visualizations, which can provide insights for visualization experts to create visualizations that guide viewer attention to important regions more effectively.

Contents

List of Figures viii
List of Tables x
List of Algorithms xi
List of Abbreviations xii

1 Introduction **1**
1.1 Background 1
1.1.1 Volume Visualization of Medical Images 1
1.1.2 Importance-Driven Visualization 2
1.1.3 Transfer Function Design 3
1.2 Scope and Motivation 4
1.3 Thesis Organization 5

2 Literature Review **6**
2.1 Importance-Driven Visualization 6
2.1.1 Focus+Context Visualization 6
2.1.2 Importance-Driven Rendering Methods 7
2.2 Transfer Function Design 8
2.2.1 Data-Centric Transfer Function Design 8
2.2.2 Image-Centric Transfer Function Design 11
2.3 Self-Organizing Map 12
2.3.1 Visualization of the SOM 12
2.3.2 Cluster Detection on the SOM 13
2.4 Rule-based Methods in Artificial Intelligence 13
2.5 Visual Attention in Visualization 14
2.5.1 Visual Attention Processing and Modeling 14
2.5.2 Guiding Attention in Visualization 15
2.5.3 Eye Tracking Based Measurement of Attentional Deployment 16
2.6 Summary 17

3 Automatic 2-D Transfer Function Design Using Visibility Distributions **18**
3.1 System Overview 19
3.2 Method 19
3.2.1 Opacity Assignment Based on Matching of Visibility Distributions 19
3.2.2 Color Assignment Based on Projective Mapping 26
3.3 Experimental Results 27

3.3.1 Parameter Settings 31
3.4 Discussion 33
3.5 Summary 33

4 Semiautomatic Multidimensional Transfer Function Specification Using Two Level Clustering 35
4.1 System Overview 36
4.2 Method 36
4.2.1 First-Level Clustering 37
4.2.2 Second-Level Clustering 38
4.2.3 Overview Generation 43
4.2.4 Volume Exploration 44
4.3 Experimental Results 46
4.4 Discussion 53
4.4.1 Parameter Choices 53
4.4.2 Comparison with Other Transfer Function Design Methods 54
4.5 Summary 56

5 Rule-Enhanced Transfer Function Generation for Medical Volume Visualization 60
5.1 System Overview 61
5.2 Method 61
5.2.1 Rule Generation 61
5.2.2 Rule Selection 62
5.2.3 Rule Evaluation 64
5.2.4 Rule-Enhanced Transfer Function Generation 66
5.3 Experimental Results 67
5.4 Discussion 71
5.5 Summary 74

6 An Eye-Tracking Based Study on Visual Attention Deployment in Volume Visualization 76
6.1 Method 77
6.2 Experimental Results 79
6.3 Discussion 85
6.4 Summary 87

7 Conclusions and Future Work 89
7.1 Future Work 90

Bibliography 92
List of Publications 106

List of Figures

1.1 Direct volume rendering pipeline. 2

3.1 System diagram of the proposed TF design method. 19
3.2 Flowchart of the proposed opacity definition scheme. 20
3.3 The proposed color assignment scheme. 27
3.4 Influence of target visibility distribution on the DTI Brain dataset. 28
3.5 Influence of threshold for zero target visibility on the Visible Male Head dataset. 29
3.6 Volume rendering results on the Pig dataset. 30
3.7 Volume rendering results on the CT Head dataset. 30
3.8 Convergence of Newton's method on the Tooth dataset. 32

4.1 System diagram of the proposed method. 37
4.2 Second-level clustering on the Tooth dataset by the normalized cut. 40
4.3 Cluster modification by weight adjustment. 43
4.4 The proposed color assignment scheme for overview generation. 44
4.5 Overviews rendered with different α and β values. 45
4.6 The proposed overview-guided exploration scheme. 46
4.7 Visualization of the Tooth dataset. 47
4.8 Visualization of the Engine dataset. 48
4.9 Visualization of the Foot dataset. 49
4.10 Visualization of the Tomato dataset. 50
4.11 Visualization of the Carp dataset. 51
4.12 Visualization of the Visible Female Feet dataset. 57
4.13 Visualization of the Pig dataset. 58
4.14 The corresponding volumetric objects of the clusters produced by the two-level clustering approach. 59

5.1 Framework of the proposed method. 61
5.2 Refine rule-evaluation result by connected component analysis. 66
5.3 Visualization of the Visible Male Head dataset. 68
5.4 Visualization of the Aneurysm dataset. 69
5.5 Visualization of the MRI Woman Head dataset. 70
5.6 Visualization of the Brain Tumor dataset. 71
5.7 Visualization of the CT Head dataset. 72

5.8 The maximum fitness value of each generation vs number of generations plot for the four evaluation datasets. 74

6.1 Our experimental setup for the eye-tracking study. 78
6.2 Fixation map generation pipeline. 79
6.3 Attention distribution in volume visualization images. 80
6.4 Comparing visual attention distribution for default vs rule-enhanced visualization. 81
6.5 Comparing visual attention distribution for different visualizations of the Visible Male Head dataset. 82
6.6 Ranking the visualization results generated in our own works for the Visible Female Feet dataset according to attention percentages. 83
6.7 Ranking the visualization results generated in our own works for the Tooth dataset according to the attention percentage within the AOI. 84
6.8 Ranking the visualization results generated in our own works for the Carp dataset according to the attention percentage within the air bladder region. 84
6.9 Comparing the expected attention distribution predicted by computational saliency models with ground truth fixation data. 87

List of Tables

3.1 Parameters and time costs of the evaluation datasets. 31

4.1 Parameters of the evaluation datasets. 52
4.2 Time costs of the evaluation datasets. 53

5.1 Time costs and parameters of the evaluation datasets. 73

6.1 Ranking the visualization results of the Visible Female Feet dataset according to the attention percentage within the AOI. 83
6.2 Ranking the visualization results of the Tooth dataset according to the attention percentage within the AOI. 83
6.3 Ranking the visualization results of the Carp dataset according to the attention percentage within the air bladder region. 84
6.4 Ranking the visualization results of the Carp dataset according to the attention percentage within the head region. 85
6.5 The final ranking of the Carp dataset visualization results according to the proposed quality metric. 85

List of Algorithms

1 Cluster Modification by Weight Adjustment 42
2 Rule-Evaluation Result Refining Based on Connected Component Analysis 65

List of Abbreviations

AOI	Area Of Interest
CT	Computed Tomography
CUDA	Compute Unified Device Architecture
DTI	Diffusion Tensor Imaging
DVR	Direct Volume Rendering
GA	Genetic Algorithm
GPU	Graphics Processing Unit
IGM	Intensity-Gradient Magnitude
JS	Jensen-Shannon
KL	Kullback-Leibler
LH	Low-High
LLE	Locally Linear Embedding
MDS	Multi-Dimensional Scaling
MRI	Magnetic Resonance Imaging
NBG	Non-Background
NN	Neural Network
PCA	Principal Component Analysis
PCP	Parallel Coordinate Plot
SOM	Self-Organizing Map
SVM	Support Vector Machine
TF	Transfer Function

Chapter 1

Introduction

1.1 Background

1.1.1 Volume Visualization of Medical Images

Visualization has become an important technique to perform data analysis for a variety of data-intensive applications. Volume visualization or volume rendering is a specific area of visualization that displays volumetric data as 2-D images by projecting the entire 3-D data to 2-D screens. Volumetric data is a set of samples (also called voxels) representing one value (i.e., scalar field data) or multiple values (i.e., vector field data) at a 3-D location. Such data can come from many different disciplines and sciences, such as medical imaging and computational fluid dynamics. Volume visualization techniques take the volumetric data as input and produce a visualization that can assist the user in understanding and analyzing the underlying problem.

Medical imaging is one of the primary sources of volumetric data. Examples of these medical imaging systems include Magnetic Resonance Imaging (MRI) and Computed Tomography (CT). Such imaging systems are used to scan internal structures of the human body and generate a stack of images or slices. While slice-by-slice viewing of these images allows for the assessment of the morphologies of anatomic and pathological structures, it is nevertheless time-consuming and does not facilitate mentally reconstructing the complex 3-D structures, and hence many aspects relevant to diagnosis and treatment planning may be missed. Volume visualization, which represents the stack of images as a whole, can present to the user a more intuitive view of the data and potentially address the drawbacks of traditional 2-D slice viewing.

The primary methods used in volume visualization are direct volume rendering (DVR), which creates visualizations by directly mapping volume data points to composited image elements, and surface rendering, which first converts volumetric data into intermediate geometric representations which are then rendered by conventional polygon rendering algorithms. DVR avoids making the binary classification decision in surface rendering, i.e., whether a surface passes through the current voxel, and hence is more efficient for datasets where a reliable

extraction of surfaces is difficult. Also, due to the growing power of graphics cards, loading the entire volume into the graphics card memory and rendering it directly is possible. On the other hand, the increasing resolution of medical volumetric data makes the generation of intermediate geometric representation computationally expensive and less efficient (as the generated geometric primitives may be smaller than a pixel on the screen). For these reasons, DVR has gained more popularity in medical visualization than surface rendering.

In DVR, the volumetric dataset is regarded as consisting of particles that simultaneously emit and absorb light. The pipeline of DVR is displayed in Fig. 1.1. The operations in the pipeline consist of interpolation, classification, shading and compositing. Interpolation is used to generate new samples that lie between the actual voxels. The sampled values in volumetric data are typically measurement of some physical properties that cannot be directly used for rendering. Optical properties (i.e., opacity, color, etc.) need to be assigned to each sample of the dataset. The process of assigning optical properties is often called "classification" and is performed by a transfer function (TF). Shading is used to improve the visual appearance of objects by applying illumination models. Finally, optical properties of the sampled points along a ray are accumulated via a composition function to determine the pixel value on the 2-D image. Popular composition algorithms include ray-casting, splatting, shear-warp, and 3-D texture-mapping [MHB+00].

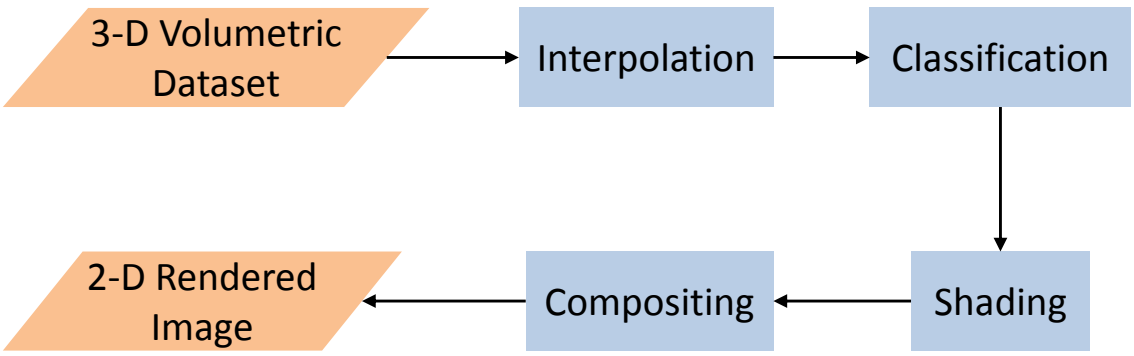


Figure 1.1: Direct volume rendering pipeline.

1.1.2 Importance-Driven Visualization

When projecting 3-D volumetric data to 2-D screens, any occlusion that is present makes the interior parts of the volume difficult to be viewed clearly in the rendered image. Volume clipping [WEE03], also known as cutaway views, is the most straightforward way to address this problem, which works by cutting away parts of the outer hull using simple geometries such as clipping planes or convex objects. However, such cutaway views may remove possibly important context information and do not facilitate the understanding of spatial relationships between structures in the volume. Readers may refer to Fig.4 in [BGKG05] for examples of

cutaway views vs context preserving visualization.

Importance-driven visualization is an alternative solution to the occlusion problem, where an additional dimension, importance, is taken into account to decide the optical properties of the volumetric data in such a way that visual prominence is given to important data-parts. The importance value can be either manually assigned to each part of the volume or directly defined and computed based on some data statistics. Importance-driven visualization avoids the binary selection of cutting away parts as done in volume clipping, since the unwanted parts can be assigned a small importance value and rendered sparsely in the final image.

Importance-driven visualization is essentially implementing the focus+context paradigm. Hauser [Hau06] generalized the definition of "focus+context visualization" as "the uneven use of graphics resources (space, opacity, color, etc.) for visualization with the purpose to visually discriminate data-parts in focus from their context, i.e., the rest of the data". Importance-driven visualization achieves the "focus+context" display by allocating the graphics resources based on data importance. It can be realized in different stages of the rendering pipeline, since each stage involves the allocation of certain graphics resource. For example, Haidacher [Hai07] applied a more complex shading method to important parts so that the lighting effect of these parts is more realistic. The magnification lens [WZMK05], which is used to magnify regions of interest for close inspection, works by modifying the direction of casted rays in the composition stage.

1.1.3 Transfer Function Design

As introduced in Section 1.1.1, TFs play the role of classification in the rendering pipeline by defining a mapping from sample values to optical properties. Mathematically, a TF can be expressed by the following formula:

$$t(f_i(x, y, z), i \in [1, \ldots, n]) \rightarrow \mathbf{o}, \tag{1.1}$$

where t denotes the TF. Functions $f_i, i \in [1, \ldots, n]$ are a set of different functions applied on a 3-D location (x, y, z), each of which returns a feature value for that location. The number of these functions n is essentially the dimension of the TF domain. The vector $\mathbf{o}$ represents a bundle of optical properties. The most important optical properties are opacity, denoted as α, and color, denoted as $\mathbf{c}$. The most simple and common 1-D TF assigns optical properties solely based on the sample's scalar value and can be expressed as:

$$t_{1\mathrm{D}}(s(x, y, z)) \rightarrow (\alpha, \mathbf{c}), \tag{1.2}$$

where s represents a function that returns the scalar value for a sample location (x, y, z). The most common 2-D TF adds the gradient magnitude as a second dimension and can be

expressed as:

$$t_{2D}(s(x,y,z), \|\frac{\partial s(x,y,z)}{\partial(x,y,z)}\|) \rightarrow (\alpha, \mathbf{c}). \tag{1.3}$$

TF design is usually conducted in the TF feature space, where the frequency distribution of the feature values of all sample points comprising the volume is displayed in order to support the user in selecting regions of interest. Figure 2 in [KKH02] presents an example of TF specification, where the scatter plot of all pairs of scalar value (the horizontal axis) and gradient magnitude (the vertical axis) is shown as the TF feature space. Voxels falling into the user-specified regions are rendered accordingly. Figure 2(a) in [KKH02] is essentially a 1-D TF that only depends on scalar value, since the assigned optical properties do not vary along the vertical axis. It can be seen that 2-D TF can provide better separability of volumetric structures than 1-D TF.

Figure 2 in [KKH02] also implies the trial-and-error nature in TF design, since the user-specified region can be placed at any position of the TF space and can be of any shape. Kindlmann [Kin02] outlined three reasons that make TF design a difficult and tedious task:

1. TFs are inherently non-spatial. It can be difficult to isolate a volumetric object in the TF domain even though the object may be easily identifiable in the spatial domain, since the feature values that characterize the object may overlap with other objects in the TF domain.
2. TFs have an enormous number of degrees of freedom in which the user can get lost.
3. The usual interfaces for TF design are generally not constrained or guided by the dataset in question. The TF domain is explored only by observing changes in the volume rendering as a result of incremental adjustments.

The problem is compounded by the ever-increasing size and complexity of volumetric datasets. Therefore, many efforts have been put into improving the efficiency and effectiveness of TF design in the past two decades.

1.2 Scope and Motivation

This thesis focuses on investigating importance-driven TF design methods for volume visualization of medical images. The goal is to assign optical properties in such a way that important data-parts are prominently visible in the rendered image. TFs play a fundamental role in volume rendering. If important parts cannot be distinguished from less important parts by a TF in the classification stage, more sophisticated techniques have to be applied in other stages of the rendering pipeline in order to visually bring out the important parts. Moreover, the effectiveness of these techniques can be greatly affected by the classification made by the TF. In the extreme cases, if all volumetric data is assigned the same opacity

and color by a TF, visually highlighting important parts in later stages of the rendering pipeline would be much more difficult. Therefore, it is highly desired to develop TF design methods that can effectively separate important data-parts from the less important ones.

1.3 Thesis Organization

The thesis is organized as follows. Chapter 2 presents a literature review on the relevant techniques. First, various techniques related to importance-driven visualization are described, followed by a review of TF design methods developed in the last two decades. We also survey the related work on two computational intelligent methods, i.e., the self-organizing map and rule-based methods, which are utilized in the following chapters to develop our TF design methods. The final section of this chapter reviews the work on modeling and measurement of visual attention deployment as well as various techniques for guiding attention in visualization.

Chapter 3 introduces an automatic 2-D TF design method using visibility distributions. The proposed method automatically assigns higher visibility values to more important regions of the volume by matching the observed visibility distribution to an importance-based target visibility distribution.

Chapter 4 extends the TF domain beyond 2-D and proposes a two-level clustering approach to address the higher-dimensional TF design problem. Higher-dimensional TFs can perform more sophisticated classification of volumetric data and thus result in better separated volumetric objects. This enables users to flexibly create a visualization by only rendering their interested structures.

Chapter 5 presents a rule-enhanced TF design method. Instead of conducting TF design in the TF space as done in Chapters 3 and 4, the proposed method in this chapter employs a set of rules, which are learned from a small amount of segmentation data labelled in the spatial domain, to distinguish the target tissue from other tissues, thus allowing important structures of the volume to be more effectively separated and highlighted.

Chapter 6 studies visual attention deployment in volume visualization with an eye-tracking experiment. The visualization images generated by the methods introduced in Chapters 3, 4 and 5, and various state-of-the-art methods are used as stimuli. Eye movement data of participants free-viewing these visualization images are recorded and analyzed to reveal how attention is deployed in different visualizations, which can help visualization experts to create visualizations that are more effective at guiding viewer attention.

Chapter 7 concludes the thesis and proposes some promising future work.

Chapter 2

Literature Review

2.1 Importance-Driven Visualization

2.1.1 Focus+Context Visualization

Focus+context visualization seeks to give visual prominence to the focus object whereas retain context information to support user navigation. A lot of focus+context techniques have been investigated in the visualization community. These techniques can be categorized by the type of graphics resources that are unevenly allocated between the data-parts in focus and the context. Some commonly used graphics resources are summarized as below.

Space Early focus+context techniques were based on the uneven allocation of the limited 2-D screen space. This was usually achieved by using various distortion lenses. The *fisheye view* [Fur86] selected information for display based on an *a-priori* importance and the distance of each data item to the focus point. Wang et al. [WZMK05] present several distortion lenses, including magnification lenses and sampling-rate-based lenses, to magnify the features of interest. The *exploded views* [BG06] can also be considered as an uneven allocation of space, where the focus object was always displayed in the center and the parts belonging to the context were forced to move away to reveal the focus object.

Opacity The uneven allocation of opacity is another commonly used method to realize focus+context visualization. Zhou et al. [ZDT04] introduced distance as a factor for volume feature enhancement. They proposed to modulate the opacity of a sample point by the distance between the sample point and the focal region center. Bruckner et al. [BGKG06] proposed a context-preserving rendering model based on lighting-driven feature classification. They modulated a sample's original opacity by the shading intensity at each sample position. Krüger et al. [KSW06] proposed another context-preserving rendering method, which used several importance measures based on curvature and distance to control the visibility of context regions via opacity modulation.

Color The portions of the data that are in focus can be discriminated from their context by assigning them a more saturated, eye-catching color. In WEAVE [GRW+00], a system for visually linking 3-D and statistical visualizations of human cardiac simulation data, data-parts selected by the user were drawn in bright colors consistently across all the visualization types whereas all the rest was displayed in gray. Apart from hue and saturation, color lightness can also be utilized to highlight the object of interest. Hauser [Hau06] suggested to display the selected object in increased color lightness and opacity in order to provide feedback to the user during 3-D object selection in virtual environment.

Rendering Styles Another method of visually distinguishing between focus and context regions is to reduce the amount of details in context regions by simplifying the rendering style. The two-level volume rendering [HMBG01] selectively applied different rendering techniques to different parts of a 3-D volume. To achieve focus+context visualization, the objects in focus can be rendered using DVR, while the context was depicted using reduced rendering styles, such as non-photorealistic rendering that only displayed object contours [CMH+01].

Apart from the above-mentioned visual features, focus+context visualization can also be realized by the uneven use of other graphics resources, such as resolution and frequency (see [Hau06] for a more comprehensive review).

2.1.2 Importance-Driven Rendering Methods

Importance as a parameter or measure has been investigated in various computer graphics rendering techniques. Despite employing the term *importance* in different contexts and meanings, all these methods consider importance of the underlying data to improve the behavior of traditional methods. In global illumination, importance sampling strategies have been widely used for path tracing, where more rays are sampled into those directions from which the light will contribute mostly to the reflected radiance [Jen95, Hd14]. Importance functions have been used in radiosity to describe the fraction of emitted radiosity that reaches the eye [SAS92]. Importance functions were also used to identify important attributes of the image that should be preserved in the halftoning process [SB98]. Tang et al. employed importance criterions to guide the determination of the sample set for texture compression. In information visualization, Hao et al. [HDKS05] generated the layout for time series based on data importance. In flow visualization, importance measures were used to control the visual appearance of particles to reduce occlusions and visual clutter [BKKW08]. Matvienko and Kruger [MK13] used gradient magnitude as an importance measure in their proposed quality metric for dense vector field evaluation.

Importance-driven methods have also been studied in volume visualization. Viola et al. [VKG04, VKG05] introduced importance-driven volume rendering techniques for automatic

focus+context display of volumetric data. They assigned an importance factor to each object of the volumetric data to describe its visibility priority. An additional rendering step called *importance compositing* was used to assign each part of an object a specific level of sparseness, which determined that part's visibility on the rendered image. In a later work, Viola et al. [VFSG06] proposed an information-theoretic framework to estimate the characteristic viewpoint for an object of interest, where importance distribution was incorporated in the viewpoint mutual information computation. Haidacher [Hai07] manipulated several optical properties, i.e., opacity, shading and silhouette enhancement, by the importance value of each sample point when visualizing the volumetric data in interventional imaging. Wang et al. [WYM08] conducted block-wise importance analysis for time-varying volumes and performed clustering on the importance curves to better uncover and visualize the important aspects of time-varying data. de Moura Pinto [dMPF10] proposed importance-aware composition, where sample importance was taken into consideration in the front-to-back composition process to modify the accumulated opacity. Wang and Kaufman [WK12] proposed an automatic color design method for DVR, where the attentiveness of an object's color was determined by its importance. Schulte zu Berge et al. [SzBBKN14] introduced point predicates for feature classification and assigned importance factors to each predicate to allow for focus+context visualization of 3-D ultrasound volumes.

2.2 Transfer Function Design

There are various ways to categorize TFs. Depending on the dimension of the function domain, TFs can be categorized as 1-D TFs, where scalar value is the only variable used to derive optical properties, and multidimensional TFs, which use other attributes such as gradient magnitude and second order derivative [Kin02]. TFs can also be categorized as data-centric and image-centric TFs [PLB+01]. Data-centric methods derive optical properties based on the analysis of the volumetric data itself, while image-centric methods analyze the properties of the rendered images and use it as a basis to generate other (and potentially better) TFs.

2.2.1 Data-Centric Transfer Function Design

1-D and 2-D Transfer Functions

A TF can be simply decided by the scalar values of the voxels. Much research has focus on identifying important isovalues of the volume. Bajaj et al. [BPS97] proposed the *contour spectrum* to assist the user in selecting relevant isovalues. Fujishiro et al. [FAT99] identified the critical isosurfaces based on a 3-D field topology analysis of the volume. Pekar et al. [PWH01] proposed an automatic method to identify dominant intensity transition values

based on maxima detection in cumulative Laplacian-weighted gray value histograms. Weber et al. segmented a volume by decomposing the contour tree of the volume [WDC+07].

1-D TFs have inherent difficulties in visualizing datasets where one data value is associated with multiple volumetric objects. This limitation can be overcome by extending the dimension of the TF domain, thus giving rise to 2-D TFs. The most widely used 2-D TF space is the intensity-gradient magnitude (IGM) histogram [KD98], where gradient magnitude is introduced as the second dimension to allow for better separation of object boundaries. More sophisticated features, such as curvature [HKG00], classification certainty [LLY06], distance to a reference object [TPD06], feature size [CM08, WK09], ambient occlusion [CM09a] and shape [PRMH10], have been proposed as the second dimension to construct a 2-D TF space together with the scalar value. The inclusion of these features provides interesting new separability of volumetric structures. Instead of always using scalar value as one dimension of the TF space, Šereda et al. [vBSG06] introduced the LH (Low-High) space that showed the lower and higher intensities of materials forming the boundaries. Selver and Güzeliş [SG09] performed feature detection on a *Volume Histogram Stack*, which was a 2-D histogram constructed by aligning the 1-D histograms of each slice of a volumetric dataset. Haidacher et al. [HPB+10] introduced the statistical TF space defined by the mean value and standard deviation of data values in a local neighborhood. To guide the selection of salient isovalues, Bruckner and Möller [BM10] proposed the isosurface similarity maps that depicted the similarity between two isosurfaces by mutual information. Maciejewski et al. [MJW+13] introduced the *abstracted attribute space*, where statistical properties found within the histogram bins were overlaid with the conventional 1-D and 2-D histograms to provide more information and cues to the user in selecting areas of interest. Guo et al. [GLY14] represented the TFs designed by a group of users with a 2-D space called *transfer function map*, which provided clues for good TFs in the form of clusters.

Apart from investigating novel attributes to construct the TF space, another important direction in 2-D TF design is to improve the efficiency of manipulating this space. The traditional design method of line ramping for 1-D TFs and polygon drawing for 2-D TFs are often tedious and laborious. Pre-dividing the TF space into meaningful sub-regions can make the exploration process more efficient. Various clustering or segmentation techniques can be applied to divide the volume into a set of clusters or segments that users can interact with, avoiding directly manipulating the TF space. Šereda et al. [vVG06] introduced a hierarchical clustering framework in LH space. Maciejewski et al. [MCWE09] divided the IGM feature space into several regions based on non-parametric kernel density estimation. Röettger et al. [RBS05] computed the spatial barycenter and variance of each bin in the IGM histogram, and performed clustering based on spatial information. Nguyen et al. [NTCO12] applied mean shift clustering to over-segment volume boundaries and then performed hierarchical clustering to group similar voxels. Wang et al. [WCZ+11] trained a Gaussian mixture model on the IGM space and a TF was designed by modulating the resulting Gaussian ellipses.

Cheuk et al. [IVJ12] applied the normalized cut algorithm to hierarchically segment the IGM histogram and introduced an information-theoretic measure to guide the exploration. Wang et al. [WZJ+12] proposed to segment the IGM histogram into a set of valley cells using the theory of Morse complex.

Higher-Dimensional Transfer Functions

The discriminative power of a TF can be further improved by increasing its dimension beyond two. However, 3-D and higher dimensional TF spaces are difficult to visualize, making the design of an effective user interface all but impossible. To address this problem, the community has developed various methods, including those based on cluster space, machine learning, parallel coordinate plot (PCP) and dimensional reduction.

Cluster space-based methods performed classification in the high dimensional feature space and transformed the volumetric data into a cluster space representation that can be manipulated by the user. Clustering algorithms that have been used to classify high dimensional volumetric data include ISODATA (Iterative Self-Organizing Data Analysis Technique) [TM04], k-means [CR08], density estimation [LVLRR08] and modified dendrogram [WZK12]. Machine learning-based methods took advantage of some well-established machine learning algorithms to "learn" a classification function that mapped voxel properties into uncertainty of whether the given voxel belonged to the material of interest. The uncertainty was then mapped to opacity during rendering. Neural networks (NNs) and support vector machines (SVMs) [TLM03, TLM05] were two examples of these learning algorithms. Soundararajan and Schultz [SS15] conducted a comparative study of several supervised classification techniques, namely, Gaussian naive Bayes classifiers, k nearest neighbor, SVMs, NNs, and random forests, with respect to probabilistic TF design. PCP represents a high dimensional point by a polyline passing through the coordinate axis of each dimension. It treats each dimension uniformly and can reveal the relationship/correlation of neighboring coordinates. The TF can be designed on the PCP by axis brushing, which selected a certain range in one dimension, and angular brushing, which selected line segments with similar slopes between two neighboring axes [ZK10, GXY11].

Dimensional reduction techniques have been widely used in high dimensional TF design. These techniques automatically project the high-dimensional data to a 2-D space, so that TF design techniques for low dimensional space can be applied. Kim et al. [KSC+10] employed the isomap, locally linear embedding (LLE) and principal component analysis (PCA) to perform dimensional reduction for multi-channel volumes. Zhao and Kaufman [ZK10] applied LLE coupled with PCP in their TF design framework to represent the TF concisely. Guo et al. [GXY11] embedded multi-dimensional scaling (MDS) plot of the high-dimensional data into the PCP plot to provide flexible feature classification without context switching. Zhou and Hansen [ZH13] employed Fastmap to performed dimensional

reduction for multivariate volume visualization. The self-organizing map (SOM) [Koh90] was another popular dimensional reduction technique that can be used for TF design. It was first suggested by Pinto and Freitas [PF07], where a TF was composed by blending several user-specified 2-D Gaussian TFs. Khan et al. [KKG13, KKG15] employed a spherical SOM to train the volumetric data and a TF was obtained by direct selection and grouping of SOM neurons.

Visibility-Driven Transfer Functions

Visibility histograms were proposed by Carlos and Ma [CM09b, CM10] to provide users immediate feedback on the visibility of particular scalar values. Based on this feedback, users can then adjust the opacity TF to maximize the visibility of interesting scalar value ranges. Wang et al. [WZC+11] extended visibility histograms to feature visibility, which measured the visibility of each feature (instead of each scalar value) in the rendered image. Ruiz et al. [RBBV11] proposed an automatic framework for TF generation based on the minimization of informational divergence between the visibility distribution captured by the viewpoints and a target distribution selected by the user.

Transfer Functions with Semantics

TFs that are designed based on semantics instead of directly manipulating the low-level volumetric attributes have been studied. Salama et al. [SKK06] applied principal component analysis on the parametric models of 2-D TFs so that the TF template can be adjusted with a single parameter. Fuzzy sets and fuzzy logic rules were employed by Rautek et al. [RBG07] to define a semantic layer that mapped semantic values of volumetric attributes to semantic values of visual styles for illustrative volume rendering. Lindholm et al. [LLL+10] introduced spatially conditioned TFs that allowed for selectively visualizing only a portion of a feature based on its spatial relation to other features. Schulte zu Berge et al. [SzBBKN14] introduced the concept of point predicates for 3-D ultrasound volume visualization, which allowed for the evaluation of both low-level information like intensity and gradient and high-level information such as anatomical models at each sampling point.

2.2.2 Image-Centric Transfer Function Design

TF design by manipulating the TF space is very unintuitive. Image-centric methods address this problem by allowing users to directly manipulate the rendered images to obtain desired visualizations. The first image-centric TF design method was proposed by He et al. [HHKP96], where genetic algorithm was employed to generate desired TFs based on users' evaluation of the rendered images. The *Design Galley* was another image-centric method, where an evolutionary strategy was applied to find the broadest selection of rendered images

[MAB+97]. Ma [Ma99] represented the volume data exploration process with an *Image Graph* that users can interact with to review a previous visualization session or to create new renderings. Wu and Qu [WQ07] proposed an image-centric TF design method that allowed users to design a TF based on multiple rendered images using editing operations (e.g., fusing, blending and deleting). A stroke-based method for 1-D TF design was introduced by Ropinski et al. [RPSH08], where the intensity range for a feature of interest was identified based on the user-drawn strokes on the rendered image. Guo et al. [GMY11] introduced the WYSIWYG (What You See Is What You Get) system which enabled users to manipulate the visual properties of the volume by brushing strokes on top of the rendered image. Spatial information was incorporated in the WYSIWYG system in a later work [GY13], which allowed features with similar scalar values to be differentiated according to their spatial locations.

2.3 Self-Organizing Map

The self-organizing map (SOM) [Koh90] is an unsupervised learning algorithm that projects data points onto a lower-dimensional space (usually 2-D space) while preserving the original topological relations. The map is initialized randomly, but after training, the map can "self-organize" itself to represent the input data distribution. This property makes it a popular tool for data mining. It has been used for data visualization, clustering, classification and vector quantization [PH08].

2.3.1 Visualization of the SOM

When visualizing a SOM, different methods distinguish themselves by the kind of information that is visualized on the map grids. For U-matrix [Ult93], the average Euclidian distance between each neuron and its immediate neighboring neurons was visualized. For P-matrix [Ult03a], the density value in the input space measured at each neuron's weight vector was visualized. The distance-based U-matrix and the density-based P-matrix can be combined to form the so-called *U*-matrix* [Ult03b] that can better describe clusters. Another visualization method that took into account both the local distance information and local density information was the *Clusot surface* [BR01], which generated the surface based on the Gaussian probability curve defined at each SOM neuron. Data topology was integrated into the SOM visualization by Tasdemir and Merényi [TM09], where adjacent neurons were connected by lines the width of which was proportional to the connectivity strength between the two neurons. De Runz et al. [DRDH12] visualized a SOM by first conducting PCA on the weight vectors and then mapping the three first components to color triplets.

2.3.2 Cluster Detection on the SOM

The visualization of a SOM can be regarded as a qualitative manner to present the information contained in the SOM. When the number of SOM neurons is large, quantitative methods are needed in order to better reveal its cluster structures. Clusters on the SOM can be detected by applying well-established clustering algorithms to the SOM weight vectors, such as hierarchical agglomerative clustering and k-means [VA00]. Kiang [Kia01] proposed another agglomerative clustering method for the SOM neurons, where two neighboring groups whose merging resulted in minimum global variance were merged in each iteration . Wu and Chow [WC04] proposed a clustering validity index that considered not only inter-cluster distance but also intra-cluster and inter-cluster density for the hierarchical agglomerative clustering of SOM.

Detecting clusters on a SOM can also be conducted on the visualization of the map instead of the neurons. A semi-automatic segmentation algorithm was proposed by [OM04], which applied a region-growing algorithm to the U-matrix visualization of the SOM to determine the frontiers of each visually identified cluster. The approach was extended to operate on the U*-matrix for better separability in low density regions and a heuristic was proposed to automatically set the threshold parameter [MU+05]. In [BBR08], a recursive flooding algorithm was applied to detect clusters on the Clusot surface.

2.4 Rule-based Methods in Artificial Intelligence

In artificial intelligence, rule-based expert systems are designed to represent the knowledge and reasoning process of human experts in domain-specific problems [BS+84]. The knowledge of a human expert is captured primarily in the form of *if-then* rules. These rules, together with a collection of facts, constitute the knowledge base of the system, from which the inference engine of the system deduces the information and relationships needed to provide answers and predictions. Dendral [BSF68], the first expert system, automated the process of identifying unknown organic molecules for organic chemists by reasoning from the knowledge of atomic mass numbers and valence rules. MYCIN, an expert system derived from Dendral [Sho76], used a simple inference engine and a knowledge base of around 600 rules to identify the bacteria causing severe infections and to recommend antibiotics. Expert systems have been successfully applied in many fields, such as geological site evaluation for mineral exploration [DGH79], computer system customization [BOBS89], route planning [VSBO11] and macromolecular crystallography data reduction [Win09]

Rule-based methods are also employed in data mining. Association rule mining is a popular and well researched method for discovering frequent patterns or relations among sets of items in a large database. The problem with mining association rules is to generate all association rules that have support and confidence greater than the user-defined minimum

values. Pioneering work was done by Agrawal et al. [AIS93] for market basket analysis, where the candidate itemsets were generated on-the-fly during the pass as the transaction data was being read. The drawback of such method was that it would incur the generation and counting of a large number of small candidate itemsets. The Apriori algorithm [AS+94] offered a more efficient method to generate the candidate itemsets by using only those that had minimum support in the previous pass to create the candidate itemsets for the current pass. The FP-growth algorithm [HPY00] removed the requirement for candidate itemset generation by compressing a large database into a frequent-pattern tree (FP tree) structure. Apart from the basic Boolean association rule learning, its variants, such as multiple-level rule learning that took into consideration the taxonomy of the database items [HF95], and quantitative association rule learning that dynamically discretized numeric attributes [SA96] have also been studied. Association rule mining has been applied in various fields such as web usage mining [SCDT00], bioinformatics [CH03] and customer relationship management [NXC09],

2.5 Visual Attention in Visualization

Human visual attention is an important consideration in visualization design. Understanding how viewers deploy their attention in a visual scene allows visualization experts to properly manage visual features for different parts of the visualization based on the amount of attention they are expected to receive, thus improving the efficiency and effectiveness of the visualization in conveying information to the viewer.

2.5.1 Visual Attention Processing and Modeling

At the early stage of visual processing, a set of low-level features, e.g., orientation, color and luminance, are processed rapidly and in parallel across the entire visual field [TG80]. These features are encoded in different topographical maps based on the center-surround mechanisms [IK01], while another topographical map, termed the saliency map, combines the information from individual feature maps into one global measure of conspicuity [KU87]. The preattentive processing ability enables subjects to catch certain properties of a visual scene without moving eyes to scan the scene. Beyond this preattentive processing stage, the perception system employs a serial computational strategy to process visual information due to limited computational capacity. Multiple stimuli in the visual scene compete for further processing and control of behaviour. Attention is an emergent property of many neural mechanisms to resolve this competition and to select relevant stimuli for detailed analysis. Attentional deployment is influenced by bottom-up saliency. For instance, the most conspicuous location in the preattentively-computed saliency map has a pop-out effect and automatically attracts attention. This form of attention is fast, bottom-up driven

and task-independent. Apart from bottom-up saliency, top-down cognitive factors, such as memory and prediction, can also influence selective visual attention. This form of attention is slower, top-down controlled and task-dependent.

Much research has been carried out towards computational modelling of visual attention. These models usually embrace the idea of a saliency map that topographically encodes for stimulus conspicuity across the visual scene. The Itti saliency model [IKN98, IK00] was built based on Treisman's Feature Integration Theory [TG80], where multiscale feature maps processed in parallel across the visual field were combined into a single topographical saliency map and a Winner-Take-All neural network with "inhibition of return" determined the focus of attention by locating the current most conspicuous location of the saliency map. Many saliency map-based visual attention models have been proposed since the introduction of the Itti model. Bruce and Tsotsos [BT07] proposed the Attention by Information Maximization (AIM) saliency model based on information theoretic metrics. Zhang et al. [ZTM+08] introduced a Bayesian framework to predict bottom-up saliency based on natural image statistics. Judd et al. [JEDT09] applied support vector machine to learn a model of saliency from human eye tracking data. Hou et al. [HHK12] proposed the *Image Signature* that isolated the foreground information of an image in the spectrum domain and facilitated the detection of salient regions by an inverse transform. Zhang and Sclaroff [ZS13] computed an attention map from a Boolean map based on Gestalt principle and generated a saliency map by combining the set of resulting attention maps. Vig et al.[VDC14] performed guided search for optimal saliency features among a large number of convolutional neural networks. The MIT Saliency Benchmark website [BJB+] provides a comprehensive list of up-to-date saliency models and their performance on the benchmark datasets. A recent survey on computational modelling of visual attention can be found in [MMKI14].

2.5.2 Guiding Attention in Visualization

Many techniques have been proposed in the visualization community to direct viewer attention to target regions in an image. The various focus+context techniques reviewed in Section 2.1.1 serve as good examples, which can guide visual attention to the focus region by enhancing certain visual features of the focus elements while suppressing those of the context elements. The importance-driven visualization techniques reviewed in Section 2.1.2 are another good example. These techniques give visual prominence to important data-parts in the visualization so that they can more easily attract viewer attention. Kim and Varshney [KV06] proposed a saliency-guided enhancement operator for volume visualization. Their technique took as input a user-specified saliency field that assigned a saliency value to each voxel of the volume. An emphasis field was computed from the saliency field based on the Itti saliency model [IKN98]. This emphasis field was then used to guide the modulation of the visual appearance by locally changing a voxel's brightness or color saturation. Different

from the above static visual attraction techniques, Waldner et al. [WLMB+14] proposed the *Attractive Flicker*, which added dynamic changes to the focus, to guide attention in dynamic narrative visualizations. Dynamic targets can draw attention more effectively than static ones, especially with increasing distance from the viewer's fixation point.

2.5.3 Eye Tracking Based Measurement of Attentional Deployment

Eye movements are closely related to shifts of attention, since humans move their eyes to align locations of interest with the foveas. Eye tracking can be used to measure where people deploy their attention within a visual scene. The two most relevant eye movements are fixations (i.e., pause of the visual gaze on a single location) and saccades (i.e., rapid movements between two fixations). Salvucci and Goldberg [SG00] described and compared various algorithms to identify fixations from raw recording data. Common analysis metrics based on these two eye movements are fixation duration, saccadic amplitude, number of fixations within area of interest and transitional properties of scanpaths [HNA+11]. Eye tracking has been widely used in areas such as neuroscience, psychology, computer science to study the underlying behavior [Duc07]. Saliency models reviewed above usually employ human fixation data as ground truth for evaluation and comparison.

In visualization, eye tracking has been used as a form of human-computer interaction. For example, the gaze-directed volume rendering proposed by Levoy and Whitaker [LW90] varied the rendering resolution according to the user's gaze position returned by an eye tracker. Okoe et al. [OAJ14] used eye-tracking as an interactive input to produce visual changes that made network reading tasks more efficient. Another application scenario of eye tracking in visualization is to evaluate and compare different visualization techniques. Tory et al. [TAK+05] employed eye gaze analysis to study the arrangement of 2-D and 3-D views in a single display. Goldberg and Helfman [GH10] conducted an eye tracking experiment to compare the effectiveness of bar, line and spider graphs in information visualization. They further compared the various versions of radial and linear graphs in value lookup tasks based on eye tracking analysis [GH11]. Toker et al. [TCSC13] used eye tracking to investigate the impact of users' cognitive abilities on visualization processing. More recently, a comparative eye tracking study on different trajectory techniques for node-link visualization was conducted by Netzel et al. [NBW14]. In scientific visualization, Kim and Varshney [KV06] conducted an eye-tracking based user study to validate the effectiveness of their enhancement technique for volume visualization. Ho et al. evaluated different 2-D flow visualization methods based on users' visual behavior [HYL+15].

2.6 Summary

This chapter begins with the review of importance-driven visualization techniques, which include focus+context visualization techniques that highlight the objects in focus from their context by allocating more graphics resources to them, and importance-driven rendering methods, where importance is used as an additional dimension to improve the behavior of traditional methods. Next, various TF design methods are reviewed and categorized as data-centric and image-centric methods. Data-centric methods are further categorized as 1-D, 2-D and higher-dimensional design methods according to the dimension of the TF. The focus of 1-D TF design is to identify important isosurfaces in the volume, while the two major directions in 2-D TF design are to investigate novel attributes to construct the TF space and to improve the efficiency of manipulating this space by pre-dividing it into meaningful sub-regions. Higher-dimensional TF methods are discussed with respect to the different approaches taken to process the high-dimensional data, which can be categorized as methods based on cluster space, machine learning, PCP and dimensional reduction. Two novel classes of TF design methods that introduce interesting new concepts, i.e., visibility and semantics, into the design are described.

Apart from importance-driven visualization and TF design, this chapter also reviews the work related to the visualization and cluster detection for the SOM. Rule-based methods in artificial intelligence, including rule-based expert systems and association rule mining, are briefly discussed. The final section of this chapter is devoted to the related work on visual attention in visualization. This section first gives an introduction on how visual attention is processed in the human perception system, followed by a survey on computational modelling of visual attention. Visualization techniques aiming to direct viewer attention to desired regions of a visual scene are discussed. The section concludes with a review on the eye-tracking methodology for visual attention measurement and its application in visualization.

Chapter 3

Automatic 2-D Transfer Function Design Using Visibility Distributions

The numerous degrees of freedom in TF specification and the non-intuitive, position-independent nature of the TF space [KKH02] make the design of a desirable TF a laborious trial-and-error process. Moreover, most interaction widgets require domain knowledge of TF space for effective manipulation, but clinicians may not possess such knowledge. An automated TF design technique to reveal as much information of the volume to the viewer, or an intuitive framework for transferring their medical domain knowledge to manipulate TFs, would increase the ease of generating good and relevant visualizations.

In this chapter, we introduce an automatic method for obtaining meaningful volume renderings based on visibility distributions. Visibility-driven TFs were first introduced by Carlos and Ma [CM09b, CM10], where visibility distributions were used to provide feedback on the current opacity TF so that users can adjust the TF to obtain sufficient visibility for the object of interest. In a later work, Ruiz et al. [RBBV11] proposed an automatic framework for TF generation based on the minimization of informational divergence, or Kullback-Leibler divergence, between the observed visibility distribution and a user-selected target distribution. In this work, we propose to employ Jenson-Shannon divergence [Lin91] to match two visibility distributions and to define target visibility distributions based on information theoretic measures. The proposed method automatically assigns higher visibility values to more important regions, which ensures they are not occluded in the resulting image and thus presents a more informative visualization result to the user. Also, an automatic color assignment scheme based on projective mapping is proposed, which allows for better differentiation of distinct structures in the volume. The visibility distribution based opacity design method, together with the projective mapping based color assignment scheme, is shown to produce satisfactory visualization results on various volumetric datasets. Lastly, we describe an intuitive framework for applying domain knowledge to modulate the visual parameters, allowing clinicians to more easily obtain their desired visualizations. Compared with state-of-the-art methods, our method requires minimal user intervention, while also providing clinicians with a more intuitive manner to manipulate the TF space.

3.1 System Overview

Figure 3.1 presents an overview of our automatic TF design method for medical visualization. There are two major modules in our system responsible for opacity assignment and color assignment, respectively. Given a volume dataset, a preprocessing module first computes the 2-D intensity and gradient magnitude (IGM) histogram. The opacity of each bin is decided by an optimization process which matches the observed visibility distribution to a target visibility distribution; the target visibility distribution is defined based on data importance, which is either set automatically based on data statistics, or set based on predetermined profiles relevant for the medical task. The IGM histogram is then segmented into several regions, each approximating a spatially connected structure, by the normalized cut algorithm [SM00]. Subsequently, the center of each segment is mapped to the CIELAB color space to determine a unique color for each region. The resulting opacity TF and color TF are then applied to render the volume and generate the visualization result.

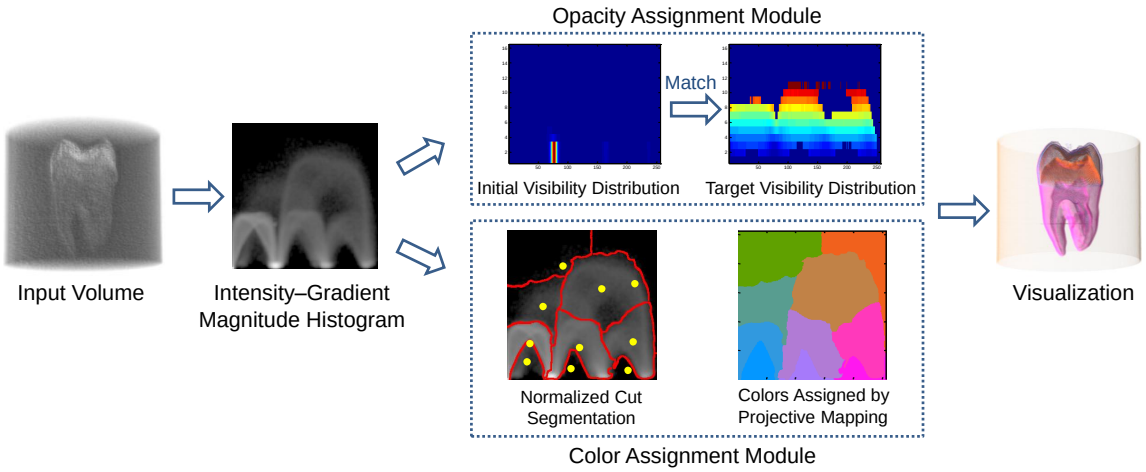


Figure 3.1: System diagram of the proposed TF design method.

3.2 Method

3.2.1 Opacity Assignment Based on Matching of Visibility Distributions

Opacity is the most important visual property in volume rendering. It decides the transparency of each voxel in the rendered image. In the realm of automatic design methods, opacity can be simply decided by intensity and modulated by gradient magnitude. The drawback of this method is that although the intention is to highlight high-intensity and high-gradient regions, such regions would still be invisible in the rendered image due to the presence of occlusions within the volume. We propose to account for these occlusions by

deciding opacity in such a way that important regions are ensured to achieve a predefined amount of visibility in the final image. Even if interesting regions are located at the very inner part of a volume and greatly occluded, our method can still make them readily visible in the rendered image, as it would modify the opacity so that these regions achieve the predefined amount of visibility.

Figure 3.2 displays the flowchart of our opacity definition scheme. Given a volume, the target visibility distribution is computed based on each bin's importance, which is quantitatively measured by its intensity, gradient magnitude and self-information [C.E48]. Users can modify the target visibility distribution using their prior knowledge. An initial TF, which is set proportional to intensity, is applied to assign opacity to each voxel. The observed visibility distribution is computed from a set of sampled viewpoints. The optimizer then iteratively modifies the TF until the stopping criterion is satisfied. In the following section, we discuss how to compute the observed visibility distribution of a volume, define the target visibility distribution and measure the divergence between the observed visibility distribution and the target visibility distribution. Finally, we provide the mathematical details on minimizing the divergence by Newton's method.

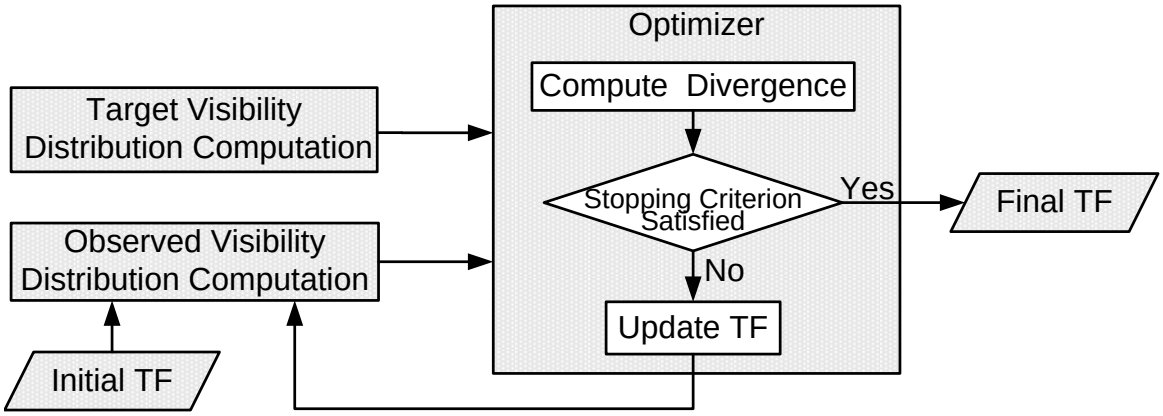


Figure 3.2: Flowchart of the proposed opacity definition scheme.

Observed Visibility Distribution

The visibility of a sample represents the contribution of the sample to the final image in terms of opacity. In the front-to-back composition of ray-casting algorithm, the opacity is accumulated over the sample points along a ray projected from viewpoint v by the following equation:

$$\alpha_{\text{out}} = \alpha_{\text{in}} + (1 - \alpha_{\text{in}})\alpha(z), \tag{3.1}$$

where α_{in} and α_{out} are the total accumulated opacity before and after the ray hit the current sample point z, and $\alpha(z)$ is the opacity of z. We define a sample point's visibility on the

rendered image as its contribution to the accumulated opacity. Let $\mathrm{vis}(z|v)$ denote the visibility of z from viewpoint v, then it can be computed by the follow equation:

$$\mathrm{vis}(z|v) = (1 - \alpha_{\mathrm{in}})\alpha(z). \tag{3.2}$$

Eq. (3.2) implies that the visibility of a sample point depends on its own opacity (i.e., $\alpha(z)$) as well as the remaining transparency before it (i.e., $(1 - \alpha_{\mathrm{in}})$).

Summing the visibility values of all voxels belonging to the same histogram bin b from viewpoint v obtains the visibility $\mathrm{vis}(b|v)$:

$$\mathrm{vis}(b|v) = \sum_{z \in \mathrm{bin}_b} \mathrm{vis}(z|v). \tag{3.3}$$

The conditional probability $p(b|v)$ is defined as the normalized visibility of bin b given viewpoint v:

$$p(b|v) = \frac{\mathrm{vis}(b|v)}{\sum_{b \in \mathcal{B}} \mathrm{vis}(b|v)}. \tag{3.4}$$

$p(b|v)$ can be regarded as the probability that the samples in bin b are visible in the rendered image. The higher the probability is, the more visible bin b is in the rendered image.

When multiple viewpoints are taken into consideration, the overall visibility of bin b is computed by summing $\mathrm{vis}(b|v)$ over all the viewpoints:

$$\mathrm{vis}(b) = \sum_{v \in \mathcal{V}} \mathrm{vis}(b|v). \tag{3.5}$$

The unconditional probability $p(b)$ is then computed as the normalized overall visibility of bin b:

$$p(b) = \frac{\mathrm{vis}(b)}{\sum_{b \in \mathcal{B}} \mathrm{vis}(b)}. \tag{3.6}$$

The volume's observed visibility distribution $p(B)$ is formed from $p(b)$ by computing all the bins' visual probabilities. From the visibility distribution of a volume, we can identify which feature is highly visible on the rendered image and which is not. By modifying the visibility distribution to approach the desired one, we can generate an informative TF that assigns more visibility to important features.

Target Visibility Distribution

Automatic Approach In our system, the target visibility distribution serves as an automatic guide in assigning opacity. Generally, regions of high intensity and gradient magnitude are important and should be assigned higher visibility. Apart from intensity and gradient magnitude, we also measure the importance of a region by its information amount.

Information can be measured based on the quality or quantity of data [HBKG08]. Quality assessments are subjective and user input is necessary to specify the important regions. When such prior knowledge is unavailable, we need to rely on the quantity, i.e., frequency, of the data to measure the information. According to information theory, events with low occurrence carry more information than events with high occurrence because people have more uncertainty about them. The *self-information*, which is used to capture this property of event occurrence, is defined as:

$$I(e) = -\log p(e), \tag{3.7}$$

where $p(e)$ is the probability of event e. If we regard the entire volume as a random variable and each histogram bin as one of its possible values, the probability of each bin can be estimated from its occurrence in the histogram. With this probability, the information amount, or self-information, contained in each bin can be measured by Eq. (3.7). In the context of medical data, bins containing higher self-information correspond to features with smaller size (e.g., border areas or tissue segments), which are always of user interest, and bins containing lower self-information can be interpreted as larger regions with the same value (e.g., the background or air), which do not provide much useful information about the volume to the user. Bins carrying more information are more important and should be highlighted in the rendered image. This motivates the following target visibility distribution definition strategies:

$$\begin{aligned} (a)\ q(b) &= \frac{I(b)\cdot \mathrm{int}(b)}{\sum_{i\in\mathcal{B}} I(i)\cdot \mathrm{int}(i)}, \\ (b)\ q(b) &= \frac{I(b)\cdot \mathrm{gmag}(b)}{\sum_{i\in\mathcal{B}} I(i)\cdot \mathrm{gmag}(i)}, \end{aligned} \tag{3.8}$$

where $q(b)$ is the target visibility value of bin b, $I(b)$ is the self-information of bin b, $\mathrm{int}(b)$ is the intensity of bin b, and $\mathrm{gmag}(b)$ is the gradient magnitude of bin b. Users can select the target visibility distribution based on their visualization interests. For example, if the user wants to highlight high-intensity region, the self-information can be multiplied by intensity; if the user is more interested in boundaries, the self-information can be multiplied by gradient magnitude.

Compared to the target distribution strategies proposed by Ruiz et al. [RBBV11], where the target visibility of each bin is proportional to intensity or gradient weighted by bin occurrence, our scheme favors bins with higher information, i.e., lower occurrence. Our rational is that, if the target visibility is weighted by bin occurrence, large-sized features, which usually correspond to background or unimportant structures (like the material between the tooth and the container in the Tooth dataset), would obtain high visibility in the rendered image, making it difficult to clearly observe inner structures. On the contrary, if we weight the target distribution by self-information, a large feature would be assigned a low target

visibility value and thus be rendered with a very small opacity, making it possible to reveal more important features in the volume. We will illustrate the influence of target visibility distributions in Section 3.3.

It should be noted that, if the number of voxels falling into a bin is very low, the self-information of this bin would be very high by definition and would be assigned a large target visibility value. To balance the visibility distribution among structures of different sizes, we set the target visibility value of bins with occurrence below certain threshold to be zero.

Incorporation of Prior Knowledge In medical applications, clinicians usually have domain knowledge on the corresponding value range for a particular tissue. This prior knowledge can be incorporated in defining the target visibility distribution such that the tissue of interest can be highlighted in the rendered image. The prior knowledge is represented with an intensity level that the user would like to highlight, and a deviation value that controls how much context around the selected intensity should be provided in the visualization. A Gaussian function is defined based on these two values:

$$g(b) = \exp\left(-\frac{(\text{int}(b) - \text{int}_u)^2}{2\sigma^2}\right), \tag{3.9}$$

where $\text{int}(b)$ is the intensity of bin b, int_u is the user-selected intensity level, and σ is the deviation. The target visibility defined in Eq. (3.8) is then multiplied by $g(b)$ (and normalized again) in order to assign a weightage to the automatic generated target visibility distribution. This allows the creation of visualization results that better meet users' needs.

Jensen-Shannon Divergence

After obtaining the observed visibility distribution and target visibility distribution, we need to find a measure to quantify the distance between them. A natural choice is the Kullback-Leibler (KL) divergence, also called informational divergence, which is defined by

$$D_{\text{KL}}(P\|Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}, \tag{3.10}$$

where $P = \{p_1, p_2, ..., p_n\}$ is the observed visibility distribution and $Q = \{q_1, q_2, ..., q_n\}$ is the target visibility distribution. However, there are some issues with KL divergence measure that make it less than ideal [BS05, WS11]. First, it is not a true metric as it is asymmetric and does not obey the triangle inequality. Second, KL divergence is undefined if $q_i = 0$ and $p_i \neq 0$. Third, KL divergence does not offer any nice upper bounds. In matching two visibility distributions, we may need to set some target visibility values to be zero, so that uninteresting regions can be rendered totally invisible in the final image. The symmetric Jensen-Shannon (JS) divergence is a more suitable candidate to match two visibility distributions, which is

defined by:

$$\begin{aligned} D_{\mathrm{JS}}(P\|Q) &= D_{\mathrm{JS}}(Q\|P) \\ &= \frac{1}{2}(D_{\mathrm{KL}}(P\|M) + D_{\mathrm{KL}}(Q\|M)), \end{aligned} \tag{3.11}$$

where $M = \frac{P+Q}{2}$. JS divergence can also be expressed in terms of entropy:

$$D_{\mathrm{JS}}(P\|Q) = H\left(\frac{1}{2}P + \frac{1}{2}Q\right) - \frac{1}{2}\left(H(P) + H(Q)\right), \tag{3.12}$$

where H represents the entropy of a probabilistic variable. The JS measure does not have the zero target visibility problem and is bounded by $0 \leq D_{\mathrm{JS}}(P\|Q) \leq 1$ (using the base 2 logarithm). In [BS05], JS divergence is used to measure the similarity of two viewpoints.

Visibility Distribution Matching Method

In this section, we discuss how to employ Newton's method to minimize the JS divergence between the observed visibility distribution and a target visibility distribution.

Newton's method is a second-order optimization method that achieves a faster converging rate than gradient descent. Given an objective function $f(\mathbf{x})$, Newton's method finds the solution to minimize $f(\mathbf{x})$ iteratively by applying the following updating rule:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - sH^{-1}(f(\mathbf{x}_t))\nabla f(\mathbf{x}_t), \tag{3.13}$$

where s is the step size, H is the Hessian matrix and ∇ is the gradient operator.

In our system, each histogram bin is assigned the same opacity value. Thus, the visual probability of bin i has the following form according to Eqs. (3.2)-(3.6):

$$p_i = \frac{e_i\alpha_i}{E}, \tag{3.14}$$

where α_i is the opacity value assigned to bin i, e_i is the total remaining transparency (i.e., $(1-\alpha_{\mathrm{in}})$ in Eq. 3.2) of bin i over all the sampled viewpoints, $E = \sum_{j=1}^{n} e_j\alpha_j$ is the normalized factor, and n is the total number of bins. In order to apply Newton's method to minimize the JS divergence, the analytic forms of the first and second order derivatives of $D_{\mathrm{JS}}(P\|Q)$, i.e., $\frac{\partial D_{\mathrm{JS}}(P\|Q)}{\partial \alpha_i}$ and $\frac{\partial^2 D_{\mathrm{JS}}(P\|Q)}{\partial \alpha_i \partial \alpha_j}$, $i, j = 1, 2, \cdots, n$, are needed. We use the entropy summation form of JS divergence (i.e., Eq. (3.12)) to deduce the derivatives. Here, $H(Q) = -\sum_{i=1}^{i=n} q_i \log q_i$ is independent of α_i, hence its derivative $\frac{\partial H(Q)}{\partial \alpha_i} = 0$. Therefore,

$$\frac{\partial D_{\mathrm{JS}}(P\|Q)}{\partial \alpha_i} = \frac{\partial H\left(\frac{1}{2}P + \frac{1}{2}Q\right)}{\partial \alpha_i} - \frac{1}{2}\frac{\partial H(P)}{\partial \alpha_i}. \tag{3.15}$$

For the first term in Eq. (3.15):

$$\begin{aligned}\frac{\partial H\left(\frac{1}{2}P+\frac{1}{2}Q\right)}{\partial \alpha_i} &= -\frac{\partial \sum_{j=1}^{n}\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)\log\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)}{\partial \alpha_i} \\ &= -\sum_{j=1}^{n}\left(\frac{1}{2}\frac{\partial p_j}{\partial \alpha_i}\log\left(\frac{1}{2}p_j+\frac{1}{2}q_j\right)+\frac{1}{2}\frac{\partial p_j}{\partial \alpha_i}\right).\end{aligned} \tag{3.16}$$

We assume that $e_j, j = 1, 2, \cdots, n$ and E are independent of α_i, since the opacities remain the same except for samples in bin i. Thus:

$$\frac{\partial p_j}{\partial \alpha_i} \approx \begin{cases} e_i/E, & j = i \\ 0, & \text{otherwise} \end{cases}. \tag{3.17}$$

Therefore,

$$\frac{\partial H(\frac{1}{2}P+\frac{1}{2}Q)}{\partial \alpha_i} \approx -\frac{1}{2}\frac{e_i}{E}\left(\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)+1\right). \tag{3.18}$$

The derivative of the entropy term $H(P)$ in Eq. (3.15) is:

$$\begin{aligned}\frac{\partial H(P)}{\partial \alpha_i} &= -\frac{\partial \sum_{j=1}^{n} p_j \log p_j}{\partial \alpha_i} \\ &= -\sum_{j=1}^{n}\left(\frac{\partial p_j}{\partial \alpha_i}\log p_j+\frac{\partial p_j}{\partial \alpha_i}\right) \\ &\approx -\frac{e_i}{E}(\log p_i+1).\end{aligned} \tag{3.19}$$

Combining Eqs. (3.18) and (3.19), we obtain the first derivative of $D_{\mathrm{JS}}(P\|Q)$:

$$\begin{aligned}\frac{\partial D_{\mathrm{JS}}(P\|Q)}{\partial \alpha_i} &\approx \frac{1}{2}\frac{e_i}{E}\left(\log p_i-\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)\right) \\ &= \frac{1}{2}\frac{p_i}{\alpha_i}\left(\log p_i-\log\left(\frac{1}{2}p_i+\frac{1}{2}q_i\right)\right).\end{aligned} \tag{3.20}$$

For Newton's method, we use Eqs. (3.17) and (3.20) to obtain the second derivative of $D_{\mathrm{JS}}(P\|Q)$:

$$\frac{\partial^2 D_{\mathrm{JS}}(P\|Q)}{\partial \alpha_i \partial \alpha_j} \approx \begin{cases} \frac{1}{2}\left(\frac{p_i}{\alpha_i}\right)^2\left(\frac{1}{p_i}-\frac{1}{p_i+q_i}\right), & j = i \\ 0, & \text{otherwise} \end{cases}. \tag{3.21}$$

With these derivatives, we can apply Newton's method to solve the optimization problem. Note that in Eq. (3.17), $e_j, j = 1, 2, \cdots, n$, and E are assumed to be independent of α_i in order to obtain a simplified close-form solution. Our experiments showed that the Newton's method can still converge quickly despite this assumption.

3.2.2 Color Assignment Based on Projective Mapping

Assigning different colors to different structures of the volume would allow them to be more intuitively discriminated by the viewer. Cheuk et al. [IVJ12] has shown that segments produced by the normalized cut algorithm on the IGM histogram represent meaningful volumetric structures. Assigning colors based on each segment instead of each bin of the IGM histogram would make it easier to identify the same structure and to distinguish different structures. The next problem is how to assign a unique color to each of the segments. Generally, coloring of multivariate data can be done by mapping the data into a color space. Kaski et al. [KVK00] proposed a non-linear projection method to project high-dimensional data to the CIELAB space, which is a perceptually uniform color space where the perceptual color difference is proportional to Euclidean distance between two colors in the space. We also adopt CIELAB as the target color space and only one plane of the space with fixed lightness value is considered. Ideally, the relative positions and distances of the original points should be preserved by the projected points as much as possible, so that the assigned colors can reflect the similarity of the volumetric structures in the transfer function space. The invariant properties of projective transformations, e.g., incidence, concurrency, collinearity, can help to achieve this goal. This motivates us to use a projective transformation to relate the two planes.

In the first step of our color assignment scheme, the normalized cut is applied to segment the log-scale IGM histogram into a predefined number of segments. Each segment i is represented by the segment centroid (xc_i, yc_i):

$$(xc_i, yc_i) = \frac{\sum_{j \in \text{seg}_i} (x_j, y_j) \log \text{occ}_j}{\sum_{j \in \text{seg}_i} \log \text{occ}_j}, \tag{3.22}$$

where occ_j is the occurrence of bin j, and (x_j, y_j) is the coordinates of bin j. From the segment centroid, a minimal bounding box containing all the center points is computed. Next, the four corners of the minimal bounding box are mapped onto the CIELAB space using a projective transformation. The four corners of the CIELAB space with lightness set to 60 are chosen as the target for the mapping to maximize the color separation in the final visualization. The homography is estimated from these four pairs of matching points using the direct linear transformation algorithm [HZ03]. Each segment is then assigned a color based on its projected position. Since the projected quadrilateral is slightly larger than the feasible CIELAB space, points that are projected beyond the space are relocated to the nearest point in the feasible region. The proposed color assignment scheme is illustrated in Fig. 3.3:

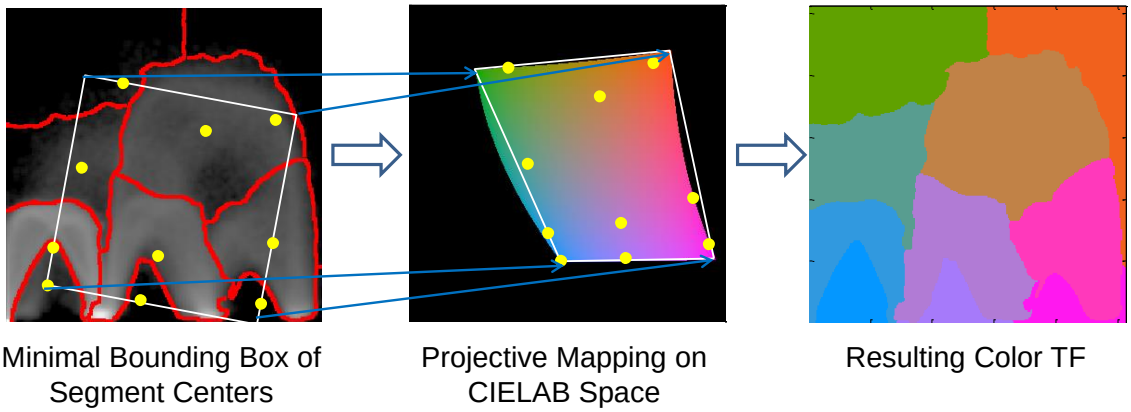


Figure 3.3: The proposed color assignment scheme. Yellow circles denote segment centroid.

3.3 Experimental Results

Influence of target visibility distribution Different target distributions result in different visualizations. The influence of target visibility distribution is illustrated by the DTI Brain dataset (8-bit Diffusion Tensor MRI) in Fig. 3.4. Figures 3.4(a) and (b) are the rendering results using the proposed target visibility distributions. It can be seen that when the target visibility is proportional to $I(b) \cdot \text{int}(b)$, regions with high intensity are rendered with higher opacity; when the target visibility is proportional to $I(b) \cdot \text{gmag}(b)$, boundaries are emphasized.

Figures 3.4(c) and (d) are the visualization results generated with target visibility weighted by bin occurrence as proposed by Ruiz et al. [RBBV11]. It is seen that under this strategy, large-size features (e.g., the background), are assigned high target visibility values and thus are still clearly visible in the rendered image, making it difficult to observe more important features (e.g., the brain).

Figures 3.4(e) and (f) are the visualization results rendered with opacity set proportional to intensity or gradient magnitude. Compared with the results in Figs. 3.4(a) and (b), it can be seen that the visibility distribution-based method makes the inner structures more prominently visible and produces clearer contrast between the cortex and the surrounding material.

Influence of threshold for zero target visibility A small threshold for setting a bin's target visibility to zero would cause small structures to be assigned a large target visibility value. As a result, other structures in the volume would be assigned lower visibility values. This would generate a transparent result with only a few small features highlighted. Figure 3.5 demonstrates the influence of threshold for zero target visibility on the Visible Male Head

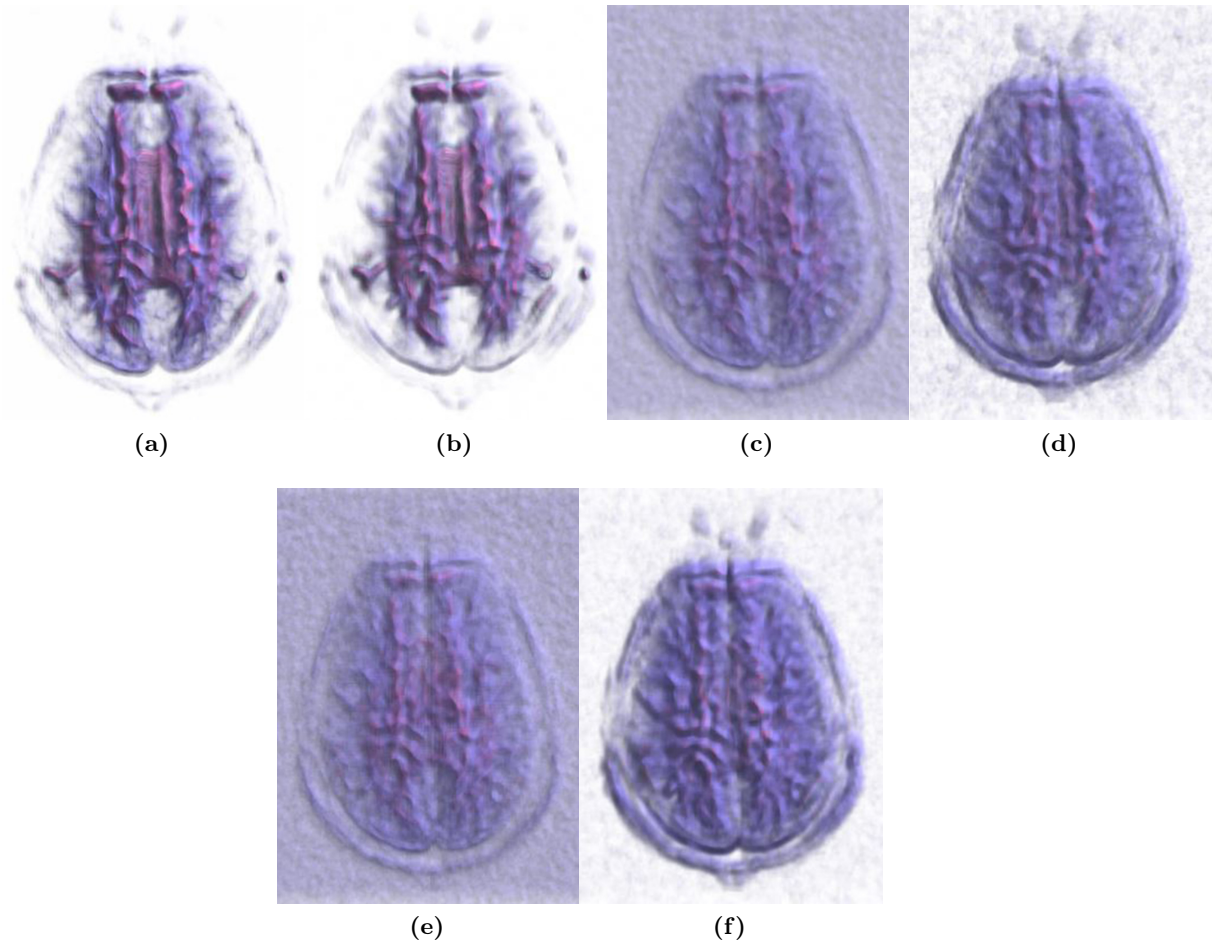
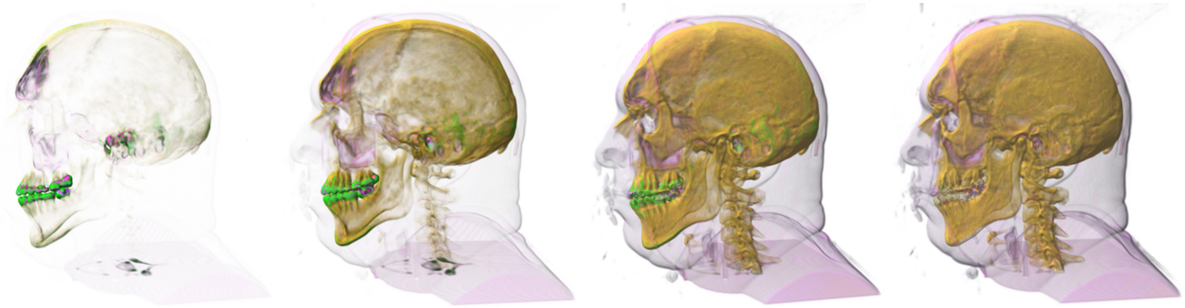


Figure 3.4: Influence of target visibility distribution on the DTI Brain dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \text{int}(b)$; (b) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \text{gmag}(b)$; (c) Rendered with target visibility distribution $q(b) \propto \text{occ}(b) \cdot \text{int}(b)$; (d) Rendered with target visibility distribution $q(b) \propto \text{occ}(b) \cdot \text{gmag}(b)$; (e) Rendered with opacity set proportional to $\text{int}(b)$; (f) Rendered with opacity set proportional to $\text{gmag}(b)$.

dataset (16-bit CT). It can be seen that when the threshold is small (Figs. 3.5(a) and (b)), the target visibility assigned to small structures (e.g., teeth, otic bones, frontal and maxillary sinuses) is high, resulting in insufficient target visibility left for other structures, such as skull and backbone. By increasing the threshold, we can achieve a more balanced visibility distribution (Fig. 3.5(c)). However, this threshold cannot be too large, otherwise some detailed structures would be forced to be invisible (e.g., the teeth tip in Fig. 3.5(d)), which is not desired. In our experiment, we found that a value between $10^{-6} \times$ volume size (i.e., the total number of voxels in the volume) and $10^{-5} \times$ volume size allowed a balancing visibility distribution for most datasets.

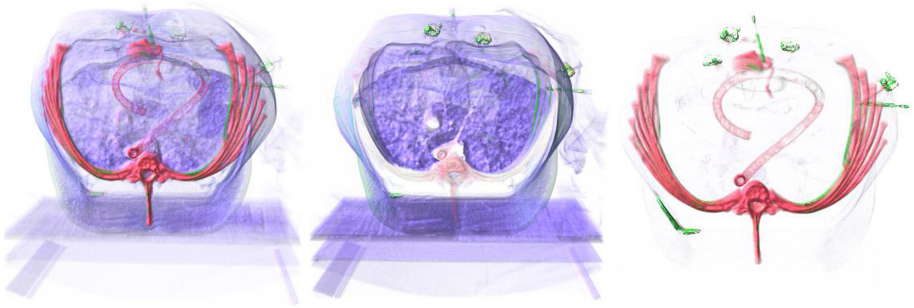


(a) (b) (c) (d)

Figure 3.5: Influence of threshold for zero target visibility on the Visible Male Head dataset. (a) Threshold $= 10^{-7} \times$ volume size; (b) Threshold $= 10^{-6} \times$ volume size; (c) Threshold $= 10^{-5} \times$ volume size; (d) Threshold $= 10^{-4} \times$ volume size.

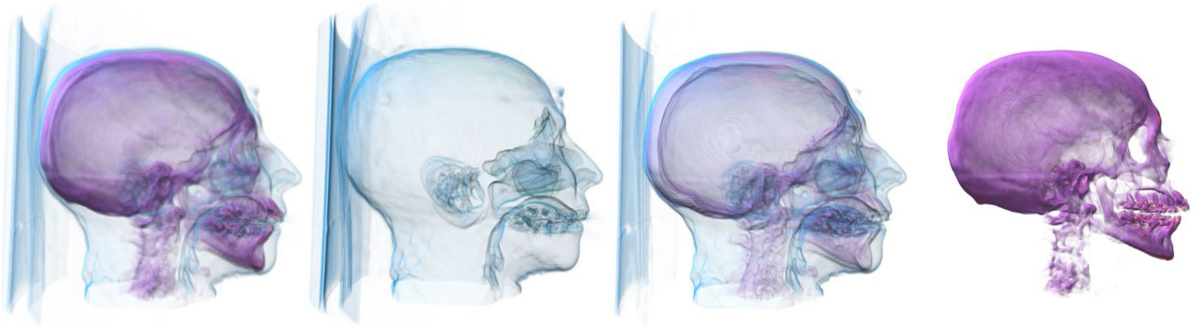
Specification of target visibility distribution using prior knowledge Medical domain knowledge can be applied to define target visibility distributions to achieve different visualization outcomes. Figure 3.6 shows three visualizations for the Pig dataset (16-bit CT) generated with different prior knowledge settings. By modulating the contributions of different bins in the target visibility distribution, the visualization can be focused on soft tissue (Fig. 3.6b), or on bone and regions with contrast agent (Fig. 3.6c). Figure 3.7 shows four visualizations for the CT Head dataset (16-bit CT). It can be seen that different user-selected values would result in visualizations emphasizing skin (Fig. 3.7b), brain (Fig. 3.7c) and bone (Fig. 3.7d). In our experiment the intensity is quantized into 256 levels, i.e., $\text{int}_u = 0, 1, 2, \cdots, 255$.

Our algorithm was implemented in CUDA C to use the GPU for the expensive visibility computation and volume rendering operations. For the normalized cut algorithm, we used the Matlab code provided by [SM04]. The experiment platform was an Nvidia GTX680 graphics card and Intel i7 2600 CPU. Table 3.1 lists the corresponding parameters and time costs for our test datasets. Our system took around 10 seconds to visualize a $256 \times 256 \times 128$ dataset. The time cost of our method can be greatly reduced by more efficient implementation of the involved algorithms, such as the normalized cut segmentation and histogram computation.



(a) (b) (c)

Figure 3.6: Volume rendering results on the Pig dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \text{int}(b)$; (b) Rendered with prior knowledge $\text{int}_u = 20, \sigma = 20$; (c) Rendered with prior knowledge $\text{int}_u = 100, \sigma = 20$.



(a) (b) (c) (d)

Figure 3.7: Volume rendering results on the CT Head dataset. (a) Rendered with target visibility distribution $q(b) \propto I(b) \cdot \text{gmag}(b)$; (b) Rendered with prior knowledge $\text{int}_u = 30, \sigma = 20$; (c) rendered with prior knowledge $\text{int}_u = 80, \sigma = 20$; (d) Rendered with prior knowledge $\text{int}_u = 170, \sigma = 20$.

Considering a user may take several minutes to get a satisfactory result by manually masking regions of interest in the TF space, our method effectively visualizes a volume with minimal user effort.

Dataset	Resolution	Target Visibility Distribution	Threshold for Zero Target Visibility	Optimization	Normalized Cut
Tooth	$256 \times 256 \times 161$	$q(b) \propto I(b) \cdot \text{gmag}(b)$	$10^{-5} \times$ volume size	2.557 s	6.170 s
DTI Scan	$128 \times 128 \times 58$	$q(b) \propto I(b) \cdot \text{int}(b)$	$10^{-5} \times$ volume size	0.481 s	5.552 s
		$q(b) \propto I(b) \cdot \text{gmag}(b)$		0.493 s	
Visible Male Head	$128 \times 256 \times 256$	$q(b) \propto I(b) \cdot \text{gmag}(b)$	$10^{-5} \times$ volume size	3.220 s	8.142 s
Pig	$256 \times 256 \times 128$	$q(b) \propto I(b) \cdot \text{int}(b)$	$10^{-6} \times$ volume size	1.844 s	8.037 s
CT Head	$256 \times 256 \times 113$	$q(b) \propto I(b) \cdot \text{gmag}(b)$	$10^{-5} \times$ volume size	4.155 s	8.167 s

Table 3.1: Parameters and time costs (in seconds) of the evaluation datasets.

3.3.1 Parameter Settings

There are several parameters involved in our system which control the tradeoff between the rendering quality and the processing time required. These parameters have been preset and fixed in our experiments.

Number of viewpoints Computing over all possible viewpoints is computationally expensive, and [RBBV11] showed that increasing the number of viewpoints (beyond a small number) resulted in little visual improvement. Therefore, only a small number of viewpoints (i.e., 6), uniformly distributed on each face of a unit cube, were used to compute the observed visibility distribution.

Bin configurations Different bin configurations were used for opacity specification and color assignment. For opacity specification, we used 256 intensity bins and 16 gradient bins. The number of histogram bins directly determines the dimension of visibility distribution. To balance histogram resolution and computational cost, more bins were assigned to quantities with a larger dynamic range (i.e., intensity) and fewer bins to quantities with a smaller dynamic range (i.e., gradient magnitude, whose values are typically distributed in the low end). For color assignment, the intensity and gradient magnitude were quantized into 256 bins each; the IGM histogram is thus represented as a 256×256 image for the normalized cut segmentation.

Stopping criterion for Newton's method In our experiments, early termination of the optimization process was performed after 10 iterations. Typically, the JS divergence was below 0.1 after 10 iterations. Further iterations reduced the divergence slowly without significant improvement in visualization quality. Figure 3.8 demonstrates the convergence of Newton's method on the Tooth dataset (16-bit CT). It can be seen that the JS divergence dropped quickly in the first several iterations, but did not decrease much in the following iterations; regarding the visual quality, a small number of iterations sufficed to produce reasonably good results. Therefore, considering the tradeoff between visual quality and computational cost, an early termination strategy was adopted.

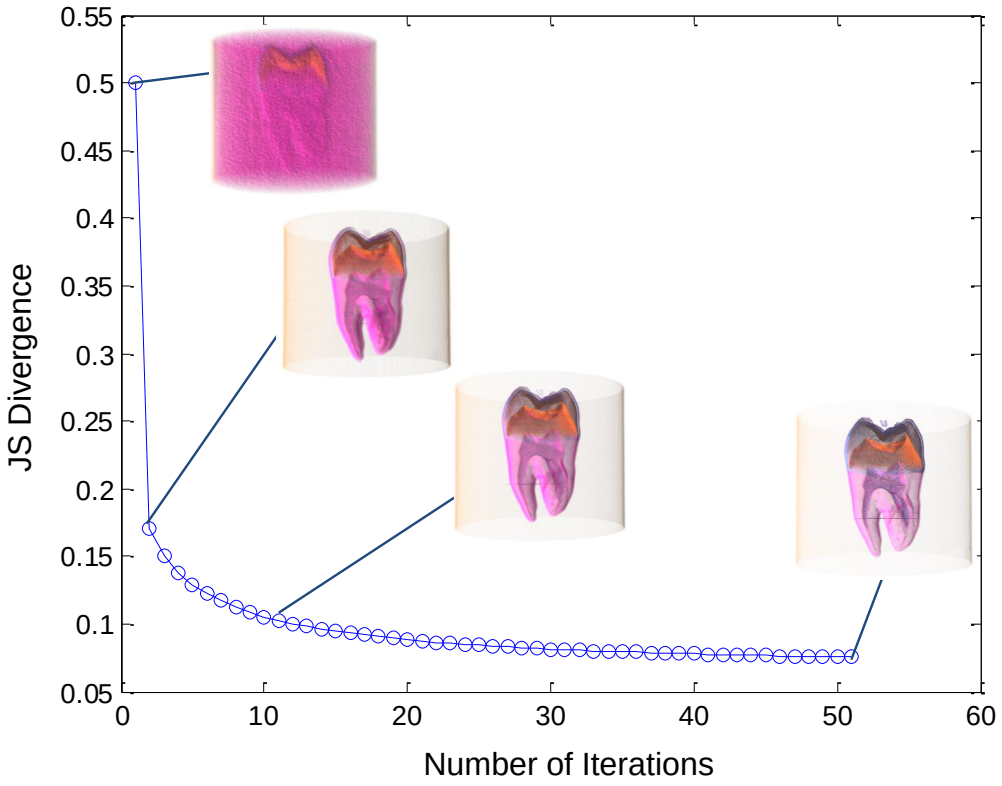


Figure 3.8: Convergence of Newton's method on the Tooth dataset.

Step size for Newton's method For the step size of Newton's method, a tradeoff between the convergence rate and accuracy has to be considered. As Newton's method is a second-order optimization method and usually converges quickly, a small step size 0.05 was chosen. This allows the algorithm to find a minimum more accurately while retaining an acceptable converging speed, as illustrated by Fig. 3.8.

Number of segments of the normalized cut In general, the more complex the volume is, the more segments we should use. In our experiment, we did not attempt to tune this parameter and fixed it to 10.

3.4 Discussion

In this section, we discuss how our framework can be applied to obtain visualizations fulfilling different medical visualization objectives, and how the visualization interface can be designed to present visualization parameters in terms of concepts intuitive to clinicians.

Scale of visualized structures Smaller structures in the volume may be structures of medical interest (tumors or blood clots), but small structures may also be the result of imaging noise. A parameter that allows the clinician to select the scale of visualized structures will be useful for medical visualization. As shown in Fig. 3.5, the threshold for zero target visibility controls the minimum size of structures that are visible. Lowering the threshold assigns a higher target visibility to small regions, thus increasing their visibility relative to larger structures. For medical visualization interfaces, this can be intuitively presented as a slider which determines the value of the threshold for zero target visibility.

Type of tissues Different biological tissues have different imaging characteristics. For example, bone has a much higher CT value than soft tissues. The automatic target visibility distributions may result in a visualization dominated by bone, while deemphasizing other soft tissues in the volume. If the desired visualization target is not bone, the automatic-generated target visibility distribution can be weighted by a Gaussian function centered at the user-selected value, as demonstrated in Figs. 3.6 and 3.7. One way to include this in a simplified interface is a slider which determines the position of the user-selected intensity level; this allows the visual contributions to be varied between soft tissues (which tend to have low CT values) and hard tissues (which tend to have high CT values).

3.5 Summary

In this work, we proposed an approach to derive an opacity TF by minimizing the JS divergence between the target visibility distribution and the observed visibility distribution. The target visibility distribution was defined based on data importance, which was quantitatively measured by the intensity, gradient magnitude and self-information of a bin. Users may incorporate their prior knowledge on important regions by weighting the automatically generated target visibility distribution with a Gaussian function centered at a user-specified intensity level. The essence of the algorithm was to use visibility distribution to guide the

design of a TF and assign an opacity value to a bin based on its target visibility, rather than directly based on its intensity and gradient magnitude. This allowed the problem of occlusion to be alleviated in the TF design process.

For the color TF, we applied the normalized cut algorithm on the IGM histogram to divide the histogram into several regions. The segment centers were computed and mapped using projective transformation to the CIELAB color space, and each segment was assigned a color determined by its projected coordinate. The normalized cut algorithm was shown to produce meaningful segments on the IGM histogram, with each segment roughly representing a structure in the 3-D volume. When the proposed method was evaluated on several medical volumetric datasets, the key structures within the volume were clearly visualized with minimal user intervention.

Chapter 4

Semiautomatic Multidimensional Transfer Function Specification Using Two Level Clustering

In Chapter 3, an automatic 2-D TF design method was described. While the proposed method allowed for the generation of an informative volume visualization with minimal user intervention, its accuracy of volumetric object classification is limited by the number of attributes used in the TF domain. By extending the TF domain into higher dimensions, the classification ability of a TF can be improved [TM04, WZK12]. However, 3-D and higher dimensional TF spaces are non-intuitive to visualize and manipulate, thus making the design of an effective user interface difficult.

In this chapter, we address the multidimensional TF design problem by a two-level clustering approach, where the first-level clustering serves to project the high-dimensional feature data onto a low-dimensional space, and the second-level clustering produces a manageable number of clusters that facilitates the detection and separation of meaningful volumetric objects. In the first-level clustering, the dimensional reduction ability and topology preserving nature of the self-organizing map (SOM) are exploited to transform the high-dimensional feature space to a 2-D embedding that is familiar and easy to understand to most users. Traditional TF design on a 2-D TF space is often achieved by direct manipulation of widgets in a bottom-up approach [KKH01, PF07, KKG15], where the user masks regions of interest in the TF space and the voxels falling into these regions are rendered accordingly to provide feedback on current TFs. However, as the user can place a widget of any size at any position in the TF space, such an exploration scheme is largely a tedious trial-and-error process. Motivated by the need to provide reasonable constraints and guidance to the process of exploring the volume, we propose to perform a second-level clustering on the SOM neurons and to provide top-down navigation for exploration by an informative volume overview. Guided by the volume overview, interesting structures in the volume are discovered interactively by the user selecting clusters to visualize and modifying the clustering results when necessary. Our interface keeps track of each interesting structure discovered, which not only enables users to inspect individual structures closely, but also allows them to compose the final visualization by fusing the structures deemed important.

The main contribution of this chapter is a novel two-level clustering approach to address

the multidimensional transfer function design problem, unlike previous approaches that employed one-level clustering [PF07, KKG15] and were mainly concerned with 2-D TF design [MCWE09, WCZ+11, WZJ+12]. The proposed two-level clustering approach is shown to perform volume structure classification better than previous methods for various datasets. Another contribution is an automatic method to generate an overview rendering of the volume. Based on the two-level clustering and volume overview, we propose a novel volume exploration scheme with top-down navigation, which reduces the trial-and-error efforts in a purely bottom-up approach.

4.1 System Overview

An overview of our method is presented in Fig. 4.1. In the preprocessing step, a simple region growing algorithm is applied to remove background voxels. This helps to prevent the SOM from being occupied by high-occurrence background voxels as well as to reduce the computational cost of the following processing steps. A multi-dimensional feature vector is extracted for each non-background voxel. We take into consideration a total of eight features in our experiment, namely, intensity, gradient, second order directive along gradient direction [KKH02] and five first order textural features, i.e., mean, standard deviation, entropy, skewness and fourth order moment [BK02], which are computed from a local histogram at each non-background voxel's neighborhood. A two-level clustering is then performed on the extracted feature vectors. First-level clustering is conducted by the SOM, which projects high-dimensional feature vectors to a 2-D topology preserving embedding. This not only allows us to leverage on the discriminative power of high-dimensional feature descriptors, but also converts the volume data into a compact and easy-to-understand 2-D form. Second-level clustering is conducted by the normalized cut algorithm [SM00], which over-segments the 2-D SOM into semantically related sub-regions. Each of these sub-regions corresponds to a cluster that potentially represents a meaningful structure in the volume.

In the volume exploration stage, an overview of the volume is automatically generated to provide the user a quick and informative look into the volume. The overview also serves to guide the user in inspecting the clusters; if the user wishes to inspect a particular structure shown in the overview, he or she can find the clusters in the corresponding colored region of the map, thus avoiding an exhaustive search of the TF space. A final visualization of the volume is composed by combining the structures deemed important by the user.

4.2 Method

In this section, we describe in detail the two-level clustering approach and the overview-guided exploration scheme.

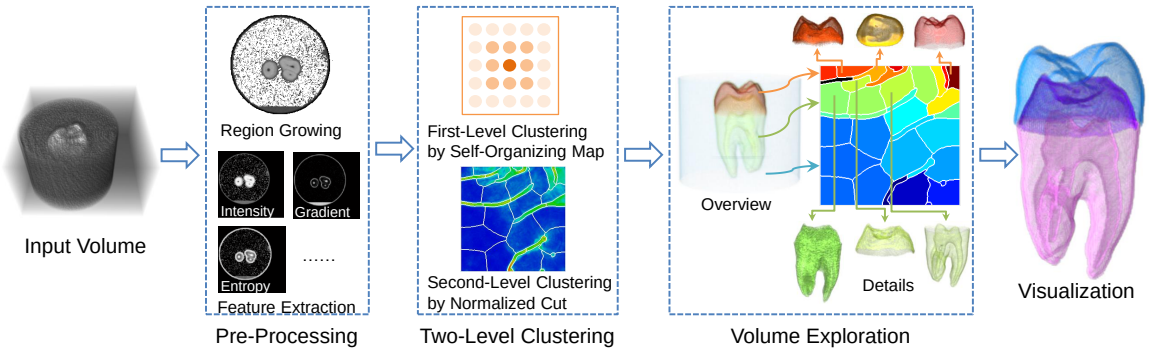


Figure 4.1: System diagram of the proposed method.

4.2.1 First-Level Clustering

The dimension reduction ability and topology preserving nature of the SOM make it a suitable tool to represent the volume data in 2-D space. While other dimension reduction techniques, such as PCA [KSC+10], LLE [ZK10, KSC+10], MDS [GXY11], have also been applied in TF design, they do not lead to a compact semantic map as SOM does.

To facilitate the design of a TF interface that is familiar to most users, we employ a rectangular map that contains a 2-D array of neurons. The neurons are related to each other by their locations in the map and each neuron is associated with a weight vector that has the same dimension as the input training vector. The map is built through an iterative unsupervised learning process comprising the following steps:

1. Initialization: Initialize the weight vectors of the SOM neurons. Random initialization or other more sophisticated initialization schemes [SLC99] can be used for this step.

2. Sampling: Randomly choose an input vector $\mathbf{x}$ from the training set.

3. Best matching unit (BMU) selection: Calculate the Euclidean distances between the input vector and the weight vectors of all neurons. BMU or the winner neuron is the neuron for which the Euclidean distance is smallest, i.e.,

$$\text{BMU} = \arg \min_{i \in [1,\ldots,n]} ||\mathbf{x} - \mathbf{w}_i(t)||, \tag{4.1}$$

   where $\mathbf{w}_i(t)$ is the weight vector of neuron i at time t and n is the number of neurons.

4. Weight update: Update the weights of BMU and other neurons with the following equation:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) h_{i,\text{BMU}}(t)(\mathbf{x} - \mathbf{w}_i(t)), \tag{4.2}$$

where

$$\eta(t) = \eta_0 \exp(-\frac{t}{T}), \tag{4.3}$$

$$h_{i,\mathrm{BMU}}(t) = \exp(-\frac{d^2_{i,\mathrm{BMU}}}{2\sigma^2(t)}), \tag{4.4}$$

$$\sigma(t) = \sigma_0 \exp(-\frac{t}{\tau}), \tag{4.5}$$

$$\tau = \frac{T}{\log \sigma_0}. \tag{4.6}$$

Here, $\eta(t)$ is the time-varying learning rate where η_0 is the initial learning rate, T is the total number of learning epochs, $h_{i,\mathrm{BMU}}$ is the time-varying neighborhood function, where $d_{i,\mathrm{BMU}}$ is the Euclidean distance between the integer 2-D coordinates of neuron i and the winner neuron, and $\sigma(t)$ is the time-varying neighborhood size where σ_0 is the initial size.

5. If the stopping criterion is met, stop; otherwise, go to 2.

The learning rate $\eta(t)$ and neighborhood size $\sigma(t)$ decrease monotonically with time, which means that the influence of the BMU on its neighbors decreases with time and is crucial for the map to converge. The neighborhood function $h_{i,\mathrm{BMU}}(t)$ takes a Gaussian form, implying that the further a neuron is from the BMU, the less its weight will be affected. This feature of SOM were inspired by neurobiological evidence for lateral interactions, where a neuron that is firing tends to excite neurons in its intermediate neighborhood more than those that are farther away. This neighborhood cooperation allows the weight vectors of the map to follow the distribution of the input space and leads to a topological ordering in the sense that neurons that are adjacent in the lattice will tend to have similar weight vectors. The topology preserving nature of SOM results in a semantic map where neurons are spatially organized in a meaningful way, and thus facilitates the generation of semantically related clusters.

4.2.2 Second-Level Clustering

SOM Neurons Clustering by Normalized Cut

The number of neurons contained in the trained SOM is much larger than the number of meaningful structures or sub-structures contained in the volume. Typically, each neuron represents a small portion of a meaningful structure and a group of neighboring neurons combine to represent a complete structure. This motivates us to conduct a second-level clustering on the SOM neurons so that users can interact with a manageable number of clusters instead of a large number of SOM neurons.

We adopted the normalized cut as our second-level clustering algorithm. Compared to other

clustering or segmentation algorithms such as k-means and the watershed algorithm, the normalized cut is more sophisticated and can produce more robust results. The normalized cut represents the set of points to be clustered as a weighted undirected graph $G = (V, E)$, where vertices V are the input points, and an edge is formed between each pair of nodes. The edge connecting nodes i and j is associated with weight $w(i, j)$, which is a function of the similarity between the two nodes. The normalized cut algorithm finds a cut to bipartition the graph into two disjoints sets A and B by minimizing the following disassociation measure:

$$\mathrm{Ncut}(A, B) = \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(B, V)}, \tag{4.7}$$

where $\mathrm{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v)$, $\mathrm{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t)$ and $\mathrm{assoc}(B, V) = \sum_{u \in B, t \in V} w(u, t)$.

Minimizing Eq. (4.7) exactly is NP-complete. By relaxing the problem in the real value domain, an approximate solution can be obtained by solving a generalized eigenvalue problem:

$$(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda \mathbf{D}\mathbf{y}, \tag{4.8}$$

where $\mathbf{W}$ is the affinity matrix of the graph and $\mathbf{D}$ is a diagonal matrix with each element being the sum of elements of the corresponding row in $\mathbf{W}$. The eigenvector $\mathbf{y}$ with second smallest eigenvalue is then quantized to bipartition the graph and the resulting segments can be recursively sub-divided to create more segments if necessary.

We regard the clustering of SOM neurons as an image segmentation problem, where the SOM is first represented by a U-matrix [Ult03a] and then fed into the normalized cut algorithm. The U-matrix has the same size as the SOM, with each element storing the average distance between a neuron and its connected neighbors. On the U-matrix, regions with similar neurons appear as "valleys" and the boundaries between two "valleys" appear as "ridges". A cluster by definition is a group of similar neurons, thus corresponding to "valleys", while cluster boundaries correspond to "ridges". The normalized cut can facilitate the detection of clusters on the SOM by segmenting "valleys" from "ridges".

We apply the normalized cut to over-segment the SOM into a small number of segments, which ensures the fine granularity of volume exploration given that the actual number of structures contained in the volume is unknown. Figure 4.2(a) shows the over-segmentation results on the Tooth dataset. It can be seen that the normalized cut is effective in separating the "valleys" and "ridges". Due to the over-segmentation, the structure represented by one cluster may be incomplete. The topology preserving nature of SOM implies that neighboring clusters usually represent similar structures. This spatial relationship between clusters provides a useful guideline for users exploring the map, which is that neighboring clusters can be combined to form a complete structure. An example is shown in Fig. 4.2(b).

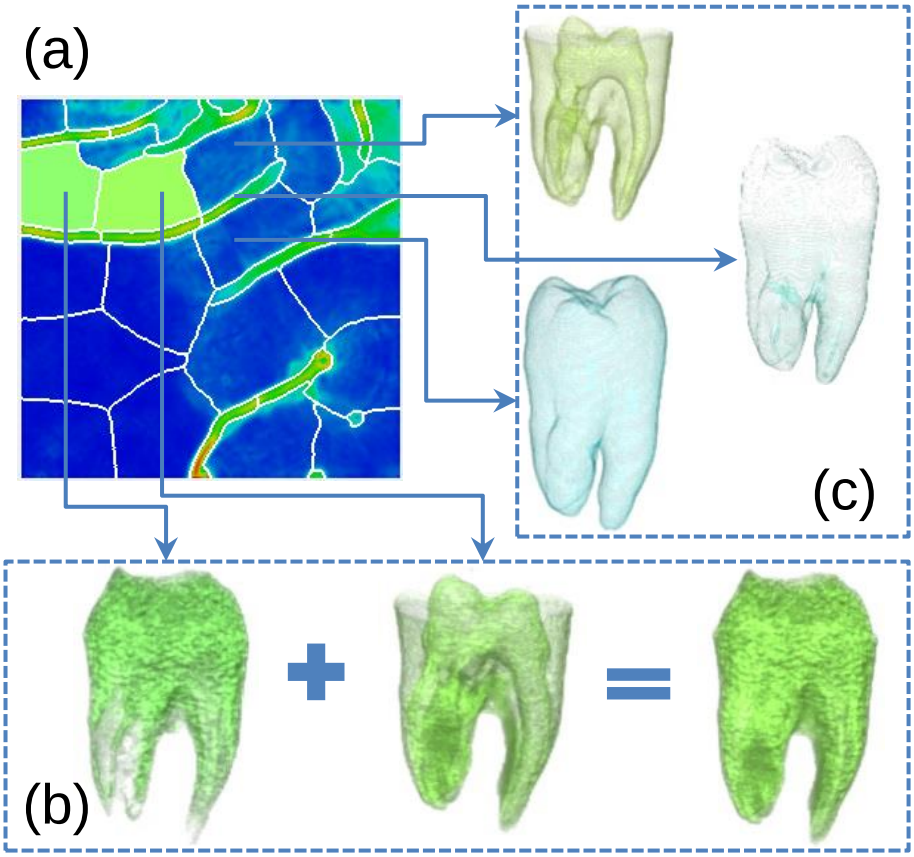


Figure 4.2: Second-level clustering on the Tooth dataset by the normalized cut. The U-matrix is mapped to a hot-to-cold color scale, where hot colors represent large values. The segment boundaries are denoted by white lines. (a) The over-segmentation results on the U-matrix of the SOM; (b) Neighboring clusters represent similar structures due to the topology preserving nature of the SOM and can be combined to form a complete structure; (c) "Valleys" represent clusters while "ridges" represent cluster boundaries.

Voxel Membership Computation

After grouping the SOM neurons into a small number of clusters, an index is needed to record each voxel's membership to the resulting clusters. We may first obtain the index of voxels belonging to each SOM neuron (first level index) and then obtain the index of SOM neurons belonging to each cluster (second level index). However, as the user only needs to interact with the clusters generated by the second level clustering, the first level index is actually redundant. From the perspective of computational cost, assigning voxels to a small number of clusters can be done much more efficiently than to a large number of neurons. This motivates us to directly compute the voxels' membership to the second level clusters.

We use a weighted Euclidean distance approach to assign voxels to the clusters. First, the

center of each cluster is computed by averaging the neuron vectors within each cluster:

$$\mathbf{c}_i = \sum_{j \in cluster_i} \mathbf{v}_j \frac{h_j}{s_i}, \tag{4.9}$$

where $\mathbf{c}_i$ is the center of cluster i, $\mathbf{v}_j$ is the weight vector of neuron j, h_j is the number of training samples whose BMU is neuron j, and $s_i = \sum_{j \in cluster_i} h_j$ is the total number of training samples in cluster i. Voxel p is assigned to the cluster for which the weighted Euclidean distance is minimized:

$$idx_p = \arg \min_{i \in [1,\ldots,n]} \frac{||\mathbf{f}_p - \mathbf{c}_i||}{weight_i}, \tag{4.10}$$

where idx_p is the cluster index of voxel p, $\mathbf{f}_p$ is the feature vector of voxel p, n is the total number of clusters, and $weight_i$ is a positive weighted value for cluster i, which can be interactively set by the user.

It has been shown that the normalized cut can effectively separate "valleys" and "ridges". "Valleys" represent clusters and "ridges" represent cluster boundaries. Strictly, we should exclude "ridges" from the assignment of voxels, but this will incur the extra step of quantitatively distinguishing between "valleys" and "ridges". We thus include all the segments in voxel membership computation and assign a voxel to its nearest neighbor among these segments. Figure 4.2(c) shows three clustering results on the Tooth dataset. It can be seen that the two "valley" clusters represent meaningful structures (the light green tooth root and the aqua tooth shell), while the "ridge" cluster separating the two "valley" clusters only contains a small fraction of voxels that do not form a meaningful structure. This provides another useful guideline for the user exploring the map, which is that priority should be accorded to "valley" clusters when selecting clusters to visualize.

Cluster Modification by Weight Adjustment

While the clusters generated by second-level clustering usually represent meaningful and well-separated structures, it is still desirable to offer a way for the user to modify the clusters when he or she is not satisfied with the automatically generated results. As voxels are assigned to the clusters according to the weighted Euclidean distance, weight adjustment serves as an effective manner to change voxel membership. We propose a cluster modification algorithm, summarized in Algorithm 1, to realize this function.

In Algorithm 1, the modification procedure takes place between the cluster whose weight is changed and the clusters that are selected by the user to take part in the modification procedure. These clusters are usually neighboring clusters as their centers are in close proximity and a change in weights will effectively change the voxel membership. To improve the efficiency of the algorithm, weight increasing and decreasing cases are processed separately.

Algorithm 1 Cluster Modification by Weight Adjustment

Input:
$cluster_{chg}$: the index of the cluster whose weight is changed
$cluster_{sel}$: the indexes of the clusters selected by the user to take part in the modification
n_{sel}: the number of clusters in $cluster_{sel}$
weight: the array that stores cluster weight values
index: the array that stores the cluster index for each voxel
Output:
updated *index*
Functions:
distance(v, s): compute the weighted Euclidean distance between voxel v and cluster s
procedure
    case $weight(cluster_{chg}) \uparrow$
    for each voxel p belonging to $cluster_{sel}$ **do**
        $dist_{new}$ = distance(p, $cluster_{chg}$)
        $dist_{old}$ = distance(p, $index(p)$)
        if $dist_{new} < dist_{old}$ **then**
            $index(p) = cluster_{chg}$
        end if
    end for
    case $weight(cluster_{chg}) \downarrow$
    for each voxel p belonging to $cluster_{chg}$ **do**
        $dist_{min}$ = distance(p, $cluster_{chg}$)
        $idx_{min} = cluster_{chg}$
        for $i = 1$ to n_{sel} **do**
            $dist$ = distance(p, $cluster_{sel}(i)$)
            if $dist < dist_{min}$ **then**
                $dist_{min} = dist$
                $idx_{min} = cluster_{sel}(i)$
            end if
        end for
        $index(p) = idx_{min}$
    end for
end procedure

The weight adjustment procedure is illustrated by an example in Fig. 4.3. Under the default weight values (1.0 for all clusters), the purple structure contains some voxels from the pink structure. As boundaries are important for visualization, the user may wish to remove the noisy voxels to obtain a clean boundary. To achieve this, the user can decrease the weight value of the purple structure so that the distances between the noisy voxels and the purple cluster are increased to the extent that the purple cluster is no longer their nearest neighbor among the three affected clusters. As a result, the noisy voxels in the purple structure are gradually reallocated to the pink structure. Once the purple structure becomes clean or the user is satisfied with the modified results, he or she can stop decreasing the weight value.

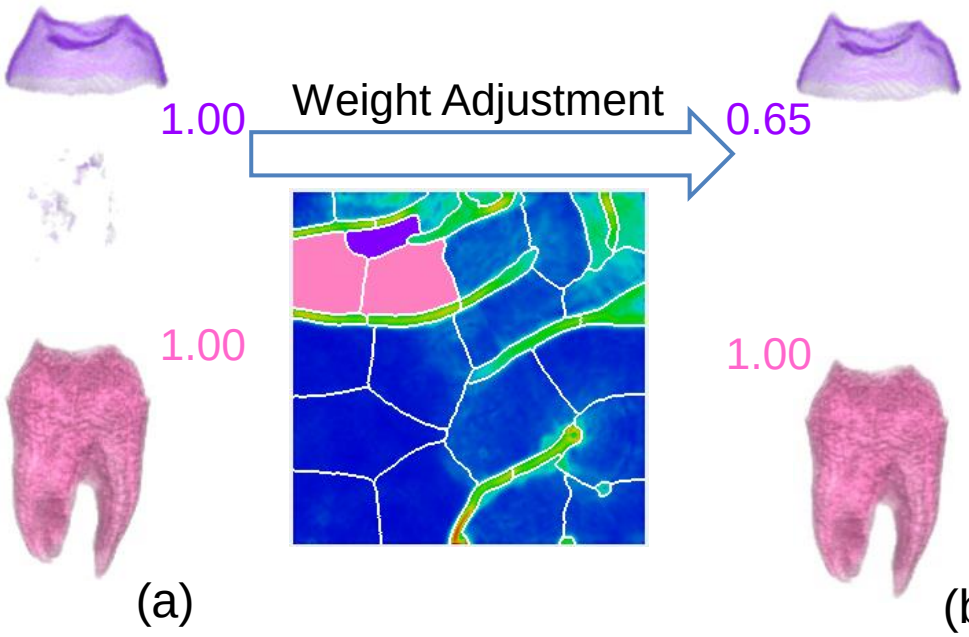


Figure 4.3: Cluster modification by weight adjustment. (a) When the weight values are the same for the purple cluster and the pink clusters, the purple structure contains some noisy voxels from the pink structure. (b) By decreasing the weight value of the purple cluster from 1 to 0.65, a clean purple structure can be obtained.

4.2.3 Overview Generation

To give the user a better idea of *what is to be explored* and *what remains to be explored* in the volume, we generate an overview of the volume by rendering all the clusters with an automatic color and opacity assignment scheme.

Our coloring scheme is illustrated in Fig. 4.4. The centers of the clusters as defined in Eq. (4.9) are utilized to determine the color assignment. This is motivated by the fact that these cluster centers essentially reflect the similarity between the clusters, and a color mapping based on them will be able to differentiate dissimilar structures. Multi-dimensional

scaling (MDS) is performed on the cluster centers to convert them to 1-D points. Each of these 1-D points determines a position on a color scale. We employ the rainbow color scale, which takes into consideration hue and lightness to maximize its discriminating ability. The cluster is then assigned the corresponding color at the projected position. Under such a coloring scheme, clusters that are close in the feature space are assigned similar colors and those that are far-away are assigned distinct colors.

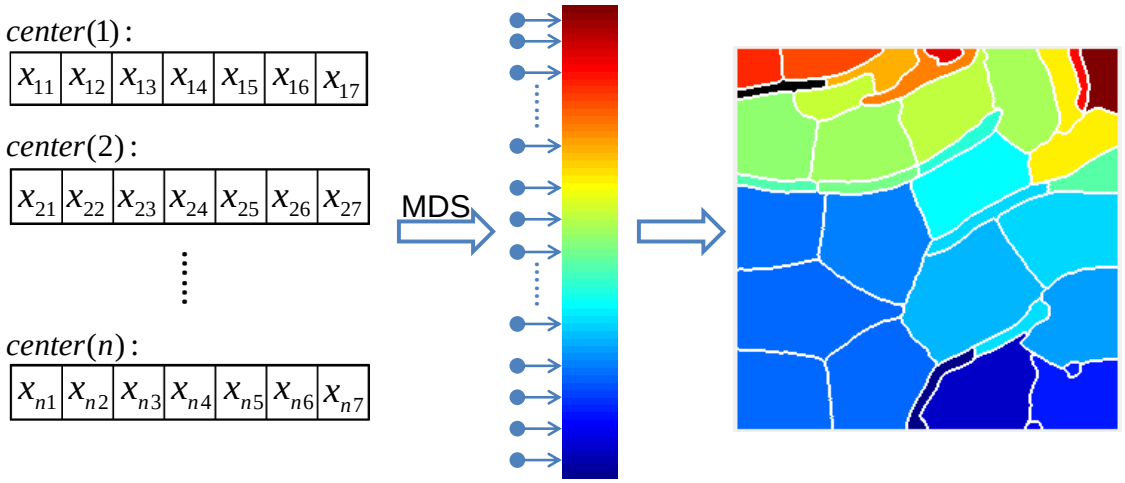


Figure 4.4: The proposed color assignment scheme for overview generation.

Our opacity assignment scheme utilizes the two fundamental properties of the volume data, intensity and gradient, to emphasize high-intensity and high-gradient regions, which usually correspond to important structures. Letting $\text{int}(i)$ and $\text{gmag}(i)$ be the average intensity and gradient magnitude of voxels in cluster i, the opacity for cluster i, $o(i)$, is obtained by:

$$o(i) = \alpha \cdot \text{int}(i) \cdot \text{gmag}(i) + \beta, \tag{4.11}$$

where $\text{int}(i) \cdot \text{gmag}(i)$ is normalized to the range $[0, 1]$, and α and β are parameters controlling the contrast and opaqueness of the overview. The influence of α and β on the overview is shown in Fig. 4.5. It can be seen that different α and β values can create different visual effects that emphasize or suppress different structures in the volume. To allow users to obtain as much information as possible from the overview, the adjustment of α and β is implemented as sliders in our interface, with the range of α set to $[0, 0.5]$ and that of β set to $[0, 0.1]$.

4.2.4 Volume Exploration

Based on the two-level clustering results and the volume overview, we propose a volume exploration scheme with top-down navigation. In our interface, the clusters are shown in their default color on the segmented SOM. The volumetric objects in the overview are rendered in

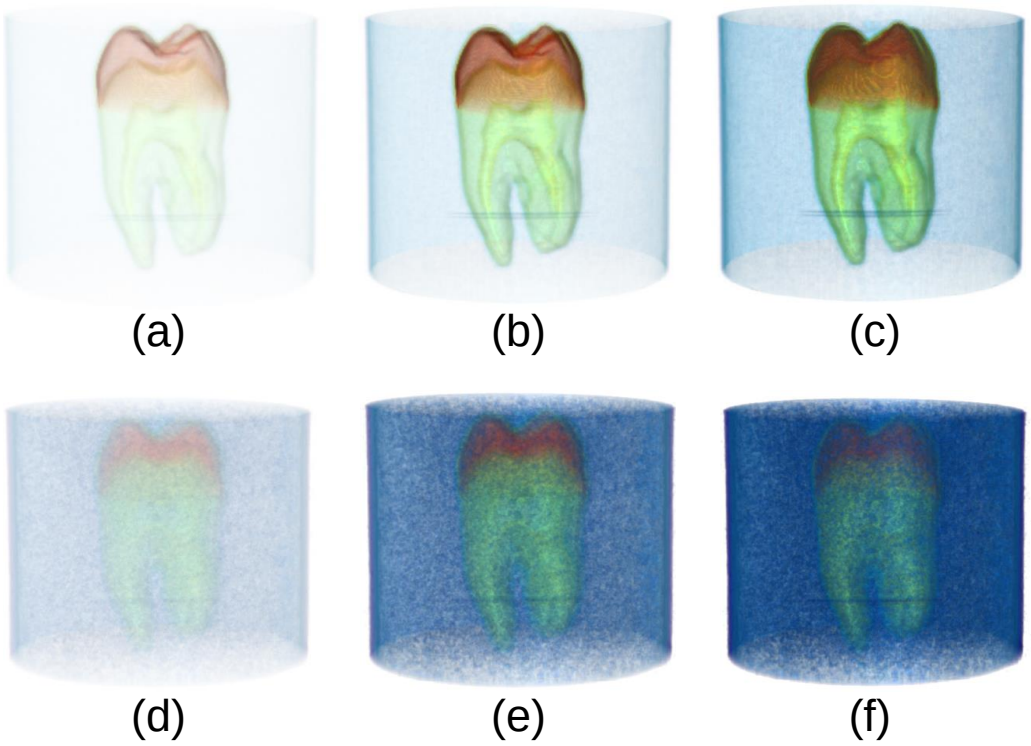


Figure 4.5: Overviews rendered with different α and β values. Top row: fix $\beta = 0$, vary α. (a) $\alpha = 0.05$; (b) $\alpha = 0.2$; (c) $\alpha = 0.5$. Bottom row: fix $\alpha = 0$, vary β. (d) $\beta = 0.01$; (e) $\beta = 0.05$; (f) $\beta = 0.1$.

their default color, and users can locate the corresponding clusters for a particular structure from the color, thus avoiding an exhaustive search of the entire map. At the beginning of volume exploration, the user adjusts the α and β sliders to obtain a quick and informative view of the volume. When the user wants to inspect a particular structure shown in the overview, he or she can just select the clusters in the corresponding colored region to look for it. The two exploration guidelines discussed in Section 4.2.2, i.e., (a) neighboring clusters can be combined to form a complete structure, and (b) priority should be accorded to "valley" clusters when selecting clusters to visualize, will be helpful for the user in exploring the volume. Once a meaningful structure is found, the user can save it and move on to explore other structures. Finally, the user creates a visualization of the volume by combining the structures deemed important.

The proposed exploration scheme is illustrated by an example in Fig. 4.6. In Fig. 4.6(a), the overview reveals that the volume contains three major structures: the orange red crown, the lawn green root and the light blue container. The container is usually of little interest to the user, so it is removed from the overview by deselecting the blue clusters on the segmented SOM (Fig. 4.6(b)). To study the details of the crown, the user selects the clusters at the top

of the segmented SOM for closer inspection, which will reveal the substructures composing the crown. Similarly, to study the details of the root, the user selects the light green clusters, which will allow the user to gain a deeper understanding of the internal structures of the root (Fig. 4.6(c)). After exploring the volume, the user can create a final visualization by fusing interesting structures and modifying the optical properties accordingly (Fig. 4.6(d)).

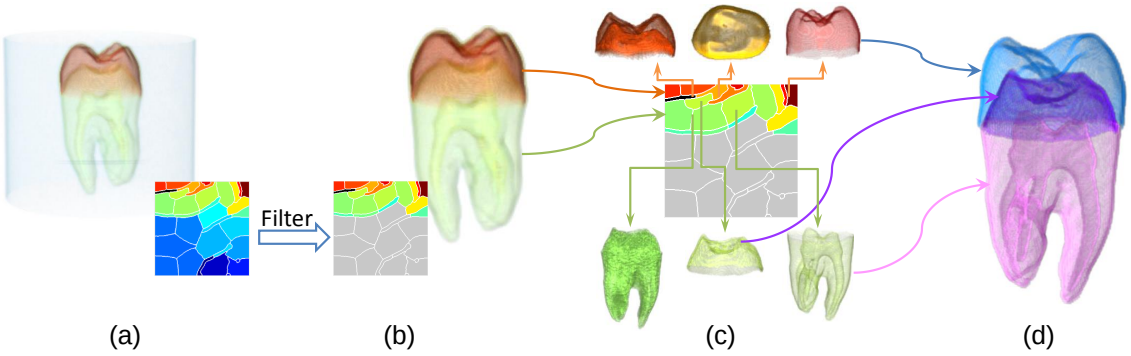


Figure 4.6: The proposed overview-guided exploration scheme. (a) Automatically generated volume overview. (b) Deselect unimportant clusters (colored in gray) from the overview to avoid visual clutter. (c) Cluster inspection guided by the overview. (d) Volume visualization by fusing interesting structures.

4.3 Experimental Results

We applied our method to a variety of volumetric datasets. For each dataset, we show the automatically generated volume overview along with the default colored clusters arranged on the segmented SOM to demonstrate how the overview can guide the exploration process. We also present the volume structures that can be discovered when the user explores the volume, with the corresponding clusters displayed in the same color of the structure that they represent. A final visualization of the volume, which is generated by combining interesting structures, is also provided. The colors of the chosen structures (indicated by red boxes) may have been modified in order to achieve more visually pleasing results.

Tooth (16-bit CT): The Tooth dataset is visualized in Fig. 4.7. From the overview, we observe that the volume contains three major structures: the orange red crown, the lawn green root and the light blue container, corresponding to the top, top-middle and middle-bottom of the SOM. By inspecting the corresponding clusters, we see the detailed structures such as enamel, the inner and outer boundaries of the enamel, the dentine, the upper and lower halves of the dentine boundary, the pulp, etc. We compose a visualization by combining the enamel, the inner and outer boundaries of the enamel, the pulp surface and a thin shell encompassing the tooth, with the color and opacity properly modulated to achieve better visual effects. Note that a different visualization of the tooth is composed in Fig. 4.7

compared to Fig. 4.6, which serves to demonstrate that our method allows users to easily create various visualizations of the same dataset based on their particular interests. The SOM was trained with seven features, namely intensity, gradient magnitude, second order derivative along the gradient direction and four first-order textural features (mean, standard deviation, skewness and entropy).

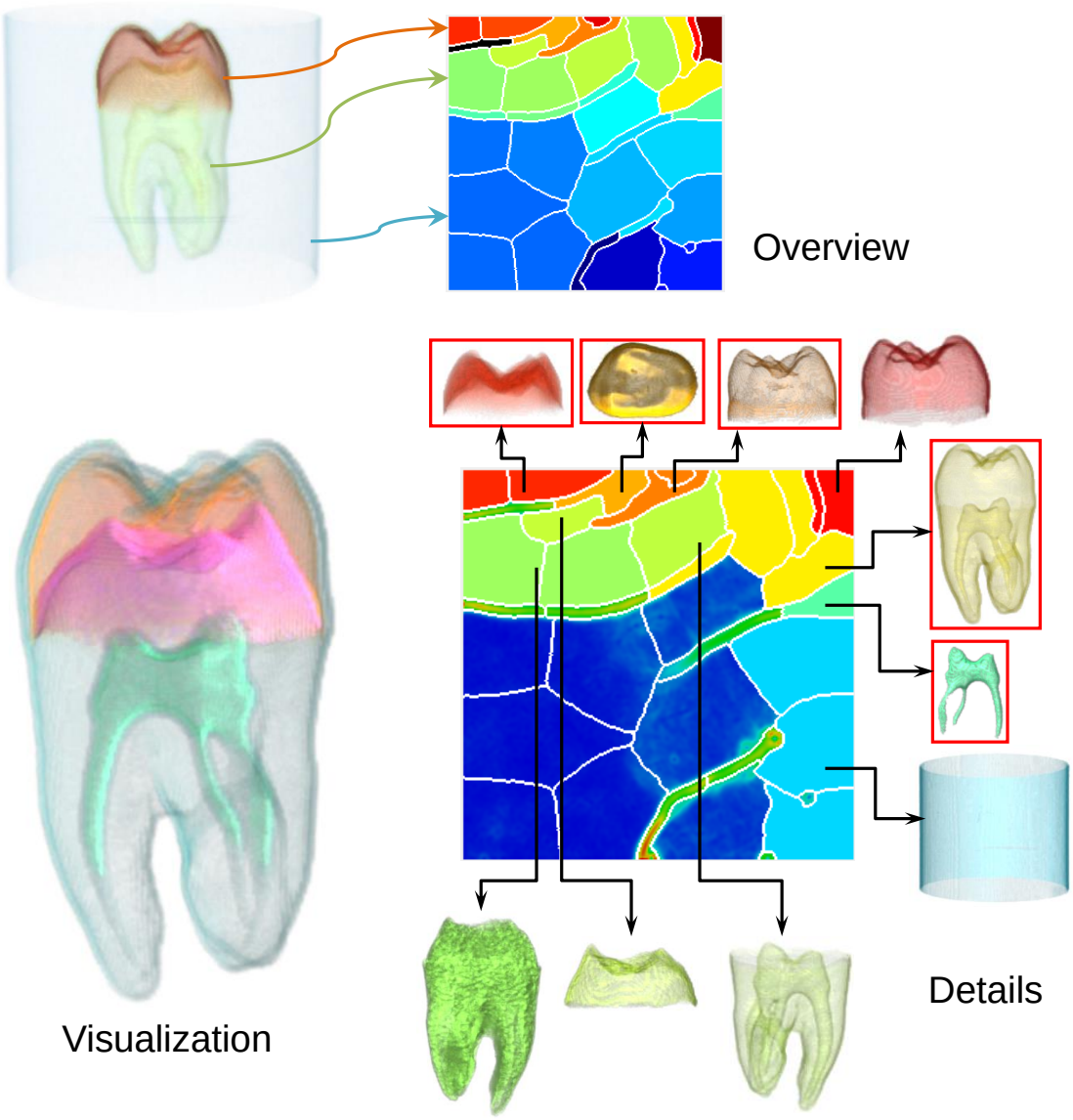


Figure 4.7: Visualization of the Tooth dataset.

Engine (8-bit CT): Figure 4.8 shows the visualization of the Engine dataset. The overview reveals that the dataset contains an engine block, with the internal structures highlighted in blue. By checking the corresponding colored clusters, we can view the internal structures at the blue region of the SOM, the engine block body at the aqua region, the engine block surface at the orange region, as well as a thin surface of the internal structure at the yellow region. The final visualization is composed by combining the internal structure, the internal

structure surface and the engine block surface. The features used to train the SOM were the same as those for the Tooth dataset.

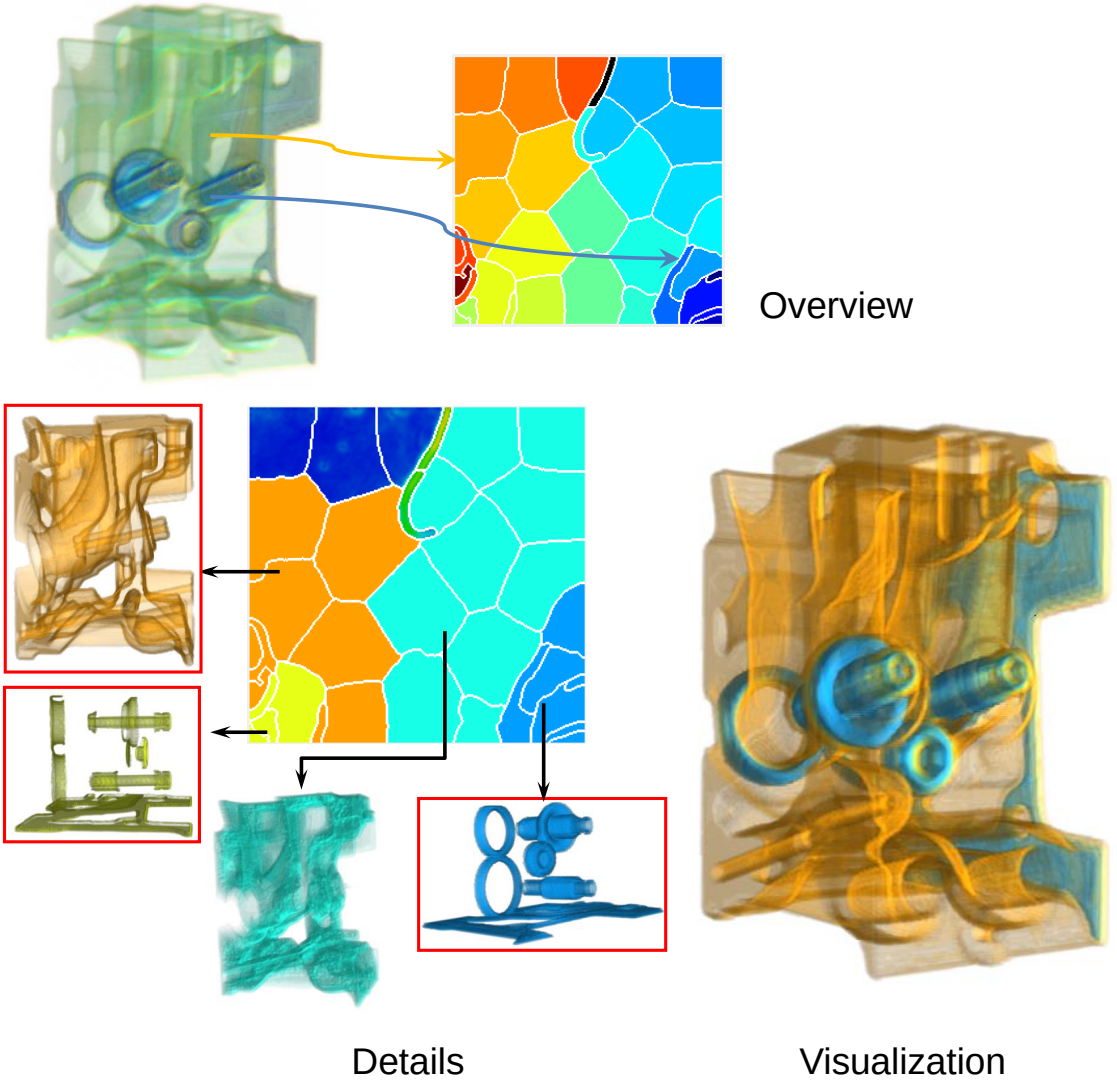


Figure 4.8: Visualization of the Engine dataset.

Foot (8-bit CT): The Foot dataset is visualized in Fig. 4.9. The overview shows three major structures in the dataset: the bones in red, the joints in green and the flesh in blue. Guided by the overview, we obtain the bones, the joints and the boundaries of the hard tissues at the top-left corner of the SOM. The flesh occupies a large portion of the map due to its high occurrence in the volume. The skin portion is located at the bottom of the map. The bones, joints, flesh and skin are combined to create the final visualization. Three features, including intensity, standard deviation and entropy, were used to train the map.

Tomato (8-bit MRI): We demonstrate our method on an MRI dataset in Fig. 4.10. Two overviews created with different α and β values ($\alpha = 0.075, \beta = 0$ for the left view and

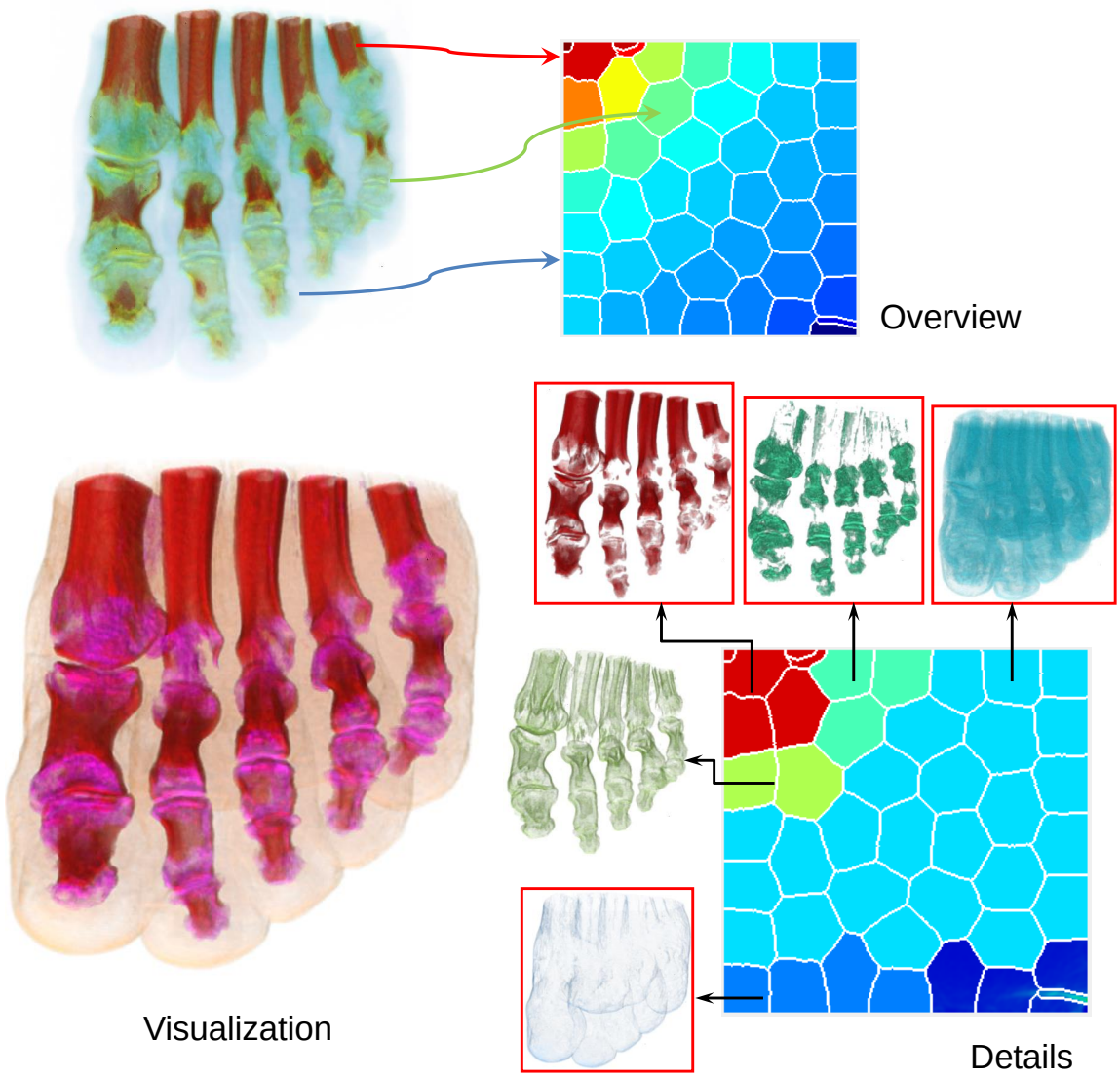


Figure 4.9: Visualization of the Foot dataset.

$\alpha = 0, \beta = 0.02$ for the right) are shown in Fig. 4.10 to illustrate the different information that can be obtained by varying α and β. The left overview informs us of the presence of the seed, valve and peel, while the right one emphasizes the orange pulp and the blue columella. With guidance from the two overviews, we can go on to obtain the detailed structures at the corresponding colored regions of the map. The final visualization is created by combining the peel, the seeds, the valve and the columella. The features used for this dataset were gradient, mean, standard deviation, skew and fourth order moment.

Carp (16-bit CT): The visualization result of the Carp dataset is displayed in Fig. 4.11. The overview presents an informative look at the dataset, which consists of the skeleton (in orange), skin and air bladder (in blue) and a thin surface of the skeleton and air bladder (in

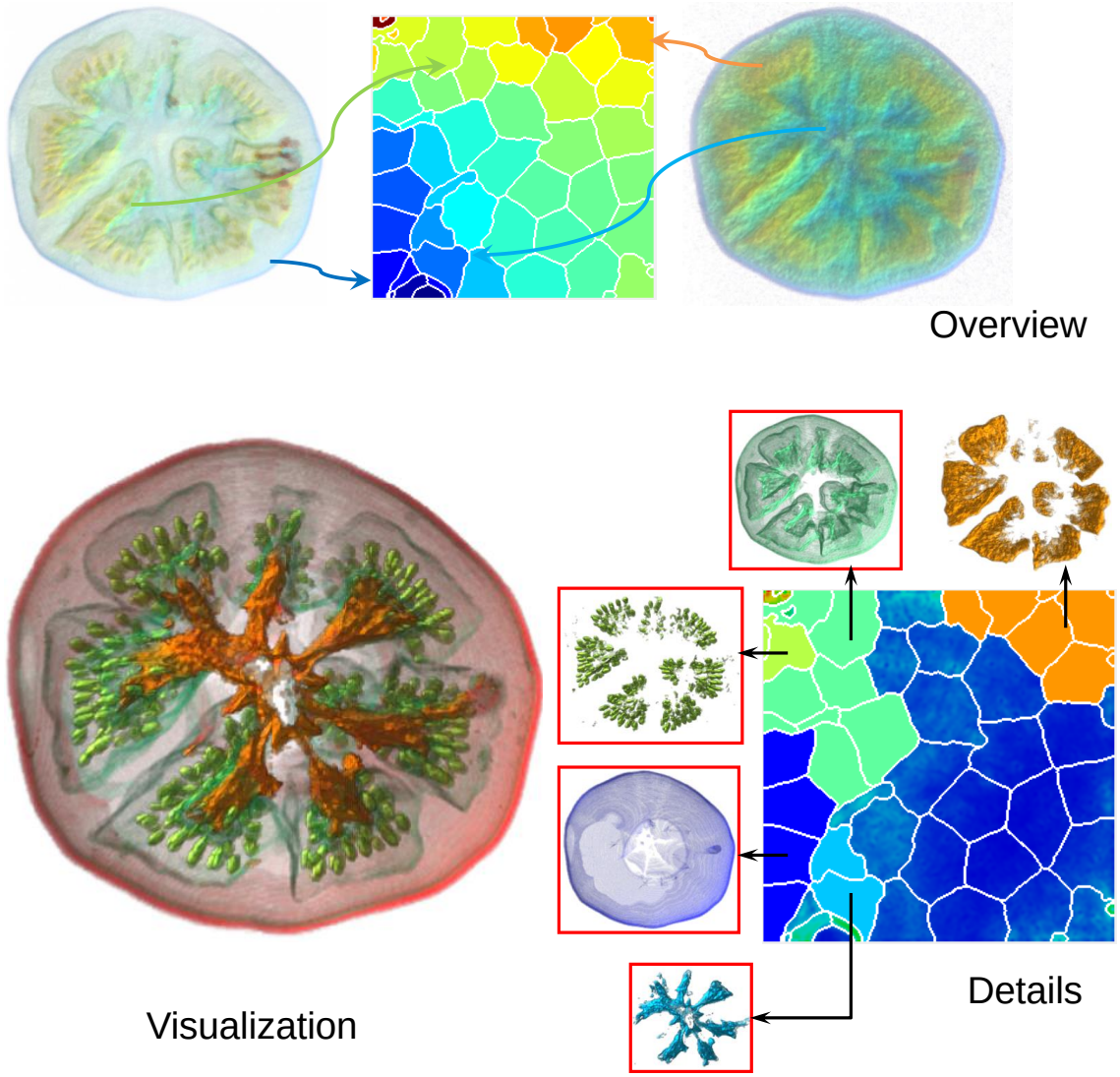


Figure 4.10: Visualization of the Tomato dataset.

yellow). We construct the final visualization by combining the skeleton, skin and air bladder. The map was trained with three features: intensity, gradient magnitude and mean.

Visible Female Feet (16-bit CT): Figure 4.12 presents the test result on the Visible Female Feet dataset. As shown in the overview, the dataset consists of the bone (in orange), joints (in yellow green), skin (in blue) and a plate (in deep blue). Four meaningful structures are provided for individual inspection and the final visualization comprises the bone, joints and skin. We used three features, intensity, mean and standard deviation, to train the SOM for this dataset.

Pig (16-bit CT): This dataset was obtained from an experiment on a live pig to test a device for secondary tracheoesophageal puncture [LCC14]. The visualization result is presented

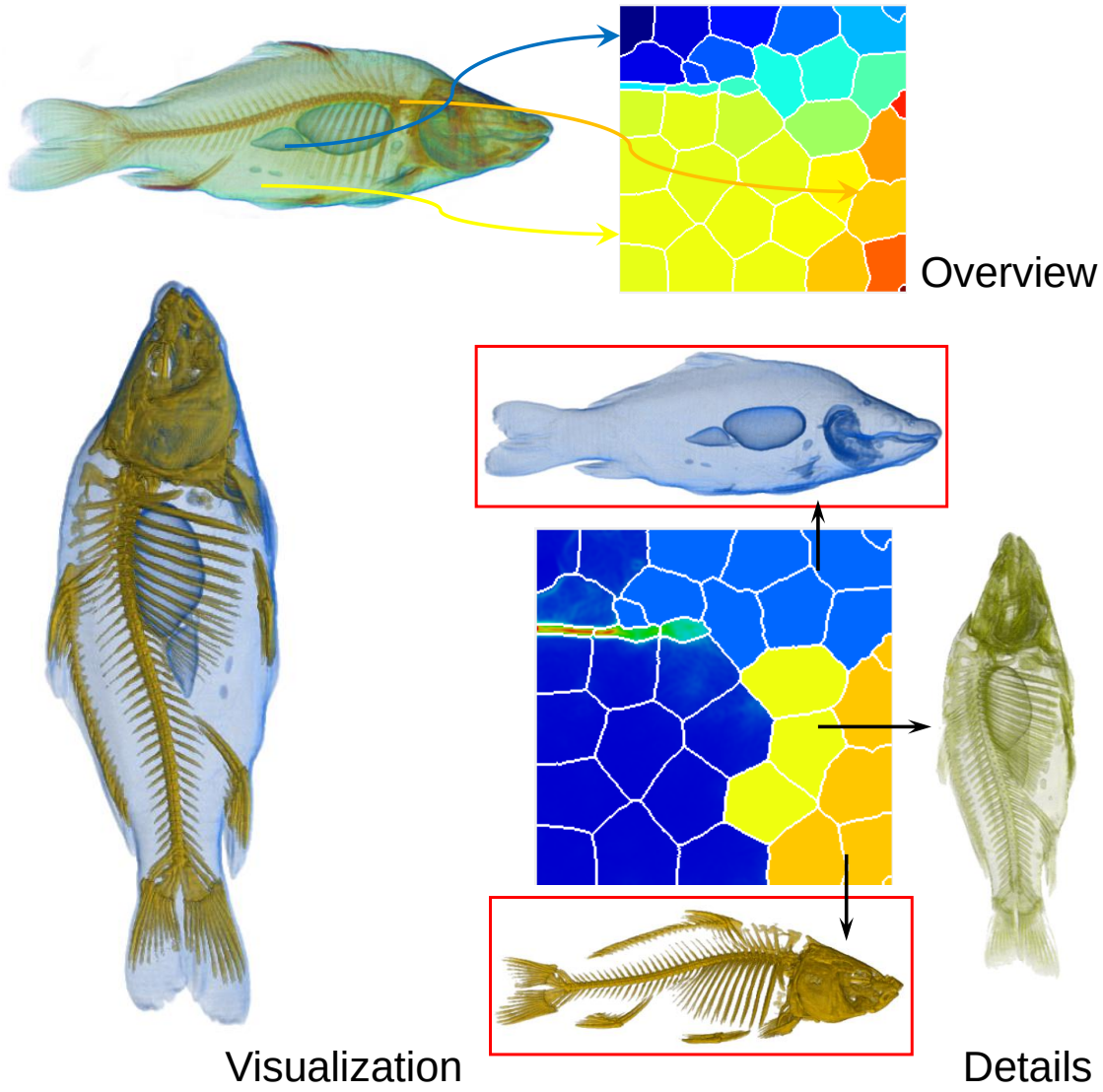


Figure 4.11: Visualization of the Carp dataset.

in Fig. 4.13. The overview reveals that the dataset consists of the yellow green bone, the blue lung, the aqua flesh and the red device in the mouth. Due to the relatively small number of voxels comprising the mouth device, this structure was represented by a very small region on the map and would be difficult to discover by trial-and-error methods. With our approach, users can easily locate the clusters corresponding to the device. The final visualization comprises the bone, lung and mouth device. Four features (intensity, gradient, mean and fourth order moment) were employed to train the SOM for this dataset.

The parameters of the evaluation datasets and the time needed for each major operation in our system are listed in Table 4.1 and Table 4.2, respectively. Our method was implemented on a PC with an Intel i7-2600 CPU and an Nvidia GTX680 graphics card. The code provided

by Shi and Malik [SM04] was used for the normalized cut segmentation. All the operations listed in Table 4.2 need to be done only once for a given dataset and the TF design is conducted at interactive frame rates.

As can be seen from Table 4.2, SOM training is the most time-consuming step. Its time complexity is $O(m^2 d_f n_s T)$, where m is the map width (and height), d_f the feature dimension, n_s the number of training samples and T the number of training epochs. The size of SOM in our experiments was set to be 256×256, the initial learning rate η_0 was 0.1, initial neighborhood size σ_0 was 362 (i.e., $\sqrt{m^2 + m^2}$). To reduce the time cost for SOM training, two measures were taken:

(a) The SOM algorithm was implemented in CUDA C to utilize the parallel computing power of Nvidia GPU. The sequential learning algorithm proposed by Gajdoš and Platoš [GP11] was implemented for SOM training, where each dimension of the weight vector was processed by one thread.

(b) Since the number of voxels is huge, it is unnecessary to use all of them for training. A subset of the voxels is sufficient to represent the volume. In our work, 1% of the total voxels were randomly sampled to train the SOM.

The training of SOM consists of a self-organizing or ordering phase to construct the map, and a convergence phase to fine tune the map. In our application, the convergence of the neuron weights is not critical, hence the second phase is not needed. For the first phase, we found that 10 training epochs was sufficient for the input samples to self-organize themselves into a reasonably good map.

Dataset	Resolution	No. of Non-Background Voxels	No. of Features	No. of Clusters
Tooth	$256 \times 256 \times 161$	1,971,836	7	35
Engine	$256 \times 256 \times 256$	1,731,738	7	35
Foot	$256 \times 256 \times 256$	5,066,965	3	45
Tomato	$256 \times 256 \times 64$	1,733,102	5	55
Carp	$256 \times 256 \times 512$	6,322,815	3	35
Visible Female Feet	$512 \times 512 \times 256$	5,275,796	3	45
Pig	$256 \times 256 \times 584$	28,217,983	4	55

Table 4.1: Parameters of the evaluation datasets.

Dataset	Region Growing	Feature Extraction	SOM Training	Normalized Cut	Voxel Assignment
Tooth	8.9	3.6	49.6	12.5	2.3
Engine	13.9	3.2	43.8	12.9	2.0
Foot	10.4	8.2	74.8	12.2	5.7
Tomato	2.4	2.9	34.1	16.5	2.0
Carp	24.3	10.8	93.1	12.0	7.3
Visible Female Feet	58.8	8.6	77.8	12.3	6.0
Pig	10.7	43.8	482	14.2	32.8

Table 4.2: Time costs of the evaluation datasets (in seconds).

4.4 Discussion

4.4.1 Parameter Choices

To ensure the effectiveness of volume exploration and visualization, the clusters proposed by the two-level clustering should represent well-separated and meaningful volumetric objects. The parameters that most greatly affect the quality of the resulting clusters are the features used for SOM training and the number of clusters in the second-level clustering.

Feature Selection

In our experiment, eight candidate features, which are intensity, gradient, second order directive along gradient direction and five first order textural features (i.e., mean, standard deviation, entropy, skewness and fourth order moment), are considered in the feature selection process. Different feature combinations can produce different results. We find that for some datasets (e.g., the Tooth dataset), using more features in the training of the SOM can result in clusters with better separability, while for other datasets (e.g., the Carp dataset), selecting more features for SOM training does not bring any improvement, probably due to the fact that the additional selected features are redundant or noisy. As the number of candidate features is small, we select the features empirically by visualizing the feature values on 2-D slices, and from there deciding whether to select a feature by observing how well it can discriminate different structures. However, manual feature selection is undoubtedly subjective and inefficient, especially when more candidate features are taken into consideration. A quantitative analysis of the feature's separability will assist the user in the feature selection process and facilitate the design of an automatic feature selection algorithm.

Though an efficient algorithm to determine the optimal feature combination is lacking in the current work, the robustness of the two-level clustering approach serves to remedy this

problem to some extent, as we find that the results are not very sensitive to the selected features. This is illustrated by the examples in Figs. 4.14(a) and (b). The results shown in Fig. 4.14(a) are trained with 7 features, which are intensity, gradient magnitude, second order derivative along the gradient direction and four first-order textural features (mean, standard deviation, skewness and entropy), while those shown in Fig. 4.14(b) are trained with only intensity and its derivatives (gradient magnitude and second order derivative along the gradient direction), excluding the textural features. Though the quality of some structures shown in Fig. 4.14(b) is degraded (e.g., objects 2, 7, 15 in Fig. 4.14(b)), other structures including important boundaries (e.g., objects 1, 6, 11 in Fig. 4.14(b)) remain well defined. Therefore, depending on the user's requirements and interests, a non-optimal feature combination may be sufficient for volume exploration and visualization, thereby easing the reliance on fine-tuned feature selection.

Number of Clusters

The number of clusters in the second-level clustering can also influence the resulting clusters. This number is set to 35-55 in our experiment. Typically, using a larger number can result in a smaller and finer structure represented by each cluster at the cost of more exploration burden for the user. A difficult dataset will generally benefit from using a larger number of clusters. Figure 4.14(a) and (c) show the structures represented by each cluster on two typical datasets. For the Tooth dataset, first-level clustering generates a SOM where the "valleys" are clearly separated from the "ridges". This indicates that clusters are well separated in the SOM, and a cluster number of 35 will be sufficient for effective visualization. On the other hand, the Tomato dataset presents a difficult case by not having clear "valleys" and "ridges" in its SOM. This implies that clusters are not well defined by the first-level clustering and thus a larger number of clusters (i.e., 55) is used in the second-level clustering to generate clusters with finer granularity. In other words, users can decide the number of clusters empirically based on the result of the first-level clustering.

4.4.2 Comparison with Other Transfer Function Design Methods

We are aware of two other high-dimensional TF design methods that also employ the SOM to perform dimensional reduction [PF07, KKG13]. These two methods only conduct one-level clustering and the TF is directly designed on the SOM visualization map in a bottom-up manner. In Pinto and Freitas's work [PF07], users are required to draw Gaussian circles on the map with the TF obtained by composing the user-specified Gaussian circles; in Khan et al.'s work [KKG13], the task of dividing the map into meaningful regions is left to the user and a TF is obtained by direct selection and grouping of SOM neurons. Our method uses the trained SOM as the input to a second-level clustering and the TF design is conducted on a manageable number of clusters, thus avoiding direct interaction with a large number

of SOM neurons and facilitating the detection and separation of interesting regions on the SOM. Moreover, our method enables the generation of an informative volume overview to provide useful guidance for volume exploration, which is not available in the other methods. For visual comparison, readers may refer to Fig.3 in [PF07], Fig.2 in [KKG13], and Figs. 4.8, 4.9 and 4.11 in this paper to compare the visualization results on the Engine, Foot and Carp datasets. It can be seen that our method can separate and highlight important structures more clearly (e.g., the internal structure of the engine block, the bone and joints of the foot, and the bone and air bladder of the carp).

The normalized cut has been used in Cheuk et al.'s work [IVJ12] to segment the IGM histogram for 2-D TF design, while in our work, the normalized cut is used to segment the SOM for higher-dimensional TF design. The accuracy of volume structure classification in their work is limited by the number of features used to construct the TF space. For instance, our method is able to separate and reveal more structures on the Foot and Tomato dataset than their method (comparing Figs. 4.9 and 4.10 in this paper, and Figs.10 and 11 in [IVJ12]).

Several works have been done to segment or cluster the TF space into meaningful regions and they all automatically create an initial rendering of the volume based on the segmentation results [MCWE09, WCZ+11, WZJ+12]. A major difference of our work from the previous works lies in the exploration schemes. In earlier works, the initial rendering is used as a starting point for transfer function design and the user performs various operations, such as changing the number of clusters in [MCWE09, WZJ+12] and modifying the shape and position of clusters in [WCZ+11], to tune the initial rendering into a desired one. As the initial rendering is not informative enough to guide the user in exploring volume structures, the user still needs to check each cluster individually in the transfer function design process. In our work, the initial rendering serves as an informative overview of the volume, which allows users to intuitively select structures of interest for close inspection with the final visualization generated by fusing interesting structures. Our approach places a high demand on the structure separability of the segments. This is guaranteed by the high-dimensional features vectors employed to construct the transfer function space, which are able to perform more sophisticated classification compared to the 2-D intensity and gradient magnitude histogram used in [MCWE09, WCZ+11, WZJ+12]. Readers may get a visual comparison on the Visible Female Feet dataset between our method and [MCWE09, WCZ+11, WZJ+12] by checking the corresponding figures (Fig. 4.12 in this paper and Fig.1 in [MCWE09, WCZ+11, WZJ+12]). It can be seen that our overview can effectively guide the user to locate the corresponding structures and our results can better distinguish between different structures (e.g., the joints and bone).

4.5 Summary

In this work, a two-level clustering approach was proposed for semiautomatic multidimensional TF design. We applied the SOM to conduct first-level clustering, which is able to "self-organize" the high-dimensional feature data on a 2-D rectangle map. To facilitate the detection and separation of meaningful volumetric objects, we applied the normalized cut on the SOM neurons to perform a second-level clustering, which resulted in clusters that represented well-separated volumetric objects. We proposed an automatic color and opacity assignment scheme to generate an informative volume overview, which can effectively guide the user in the volume exploration process. By selecting and modifying clusters to visualize, the user gradually understands the volume's content, i.e., what structures the volume contains and the relationship between them. The user can then compose the final visualization based on their understanding of the volume. The effectiveness of our method has been demonstrated on a large variety of volumetric datasets.

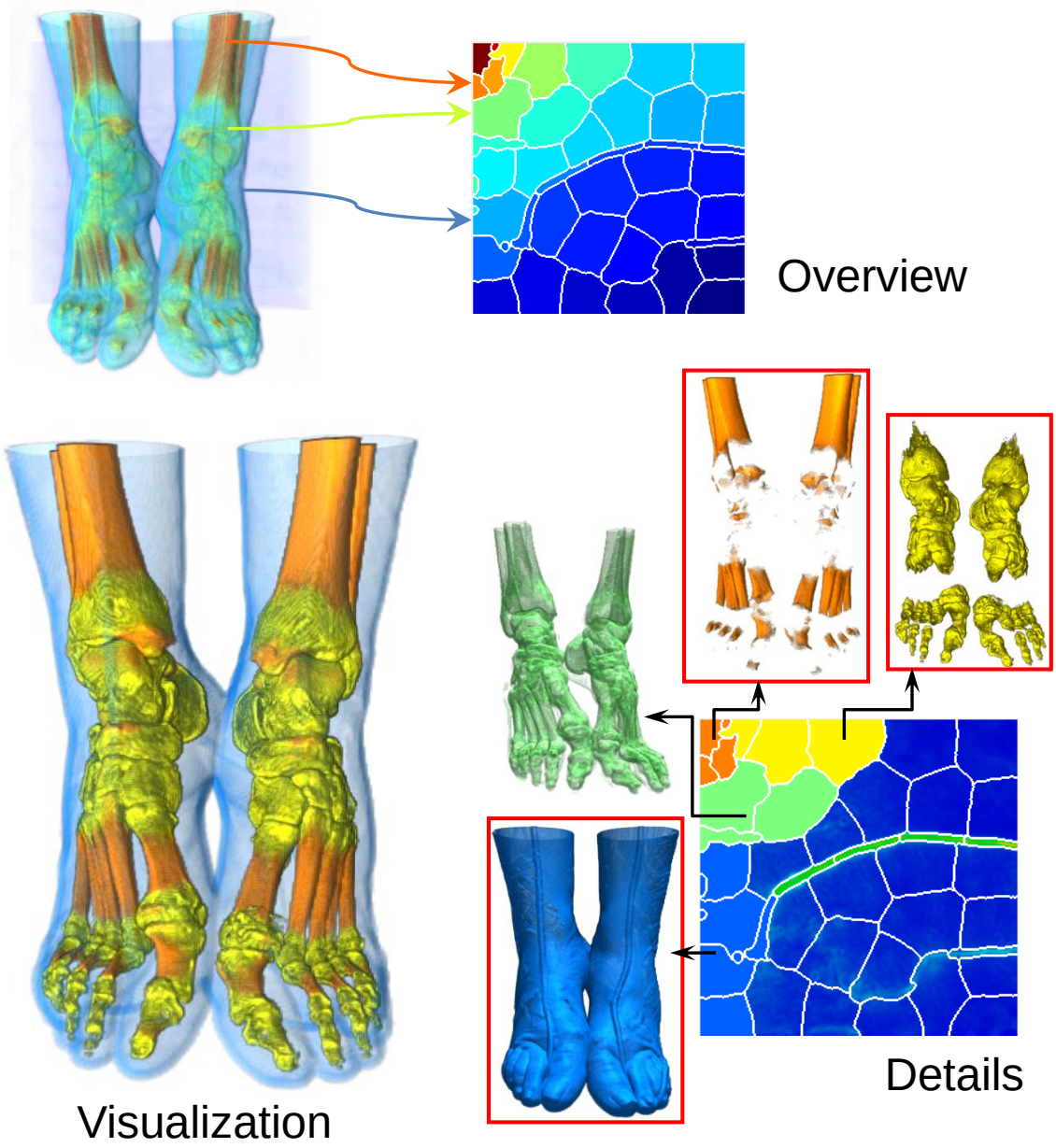


Figure 4.12: Visualization of the Visible Female Feet dataset.

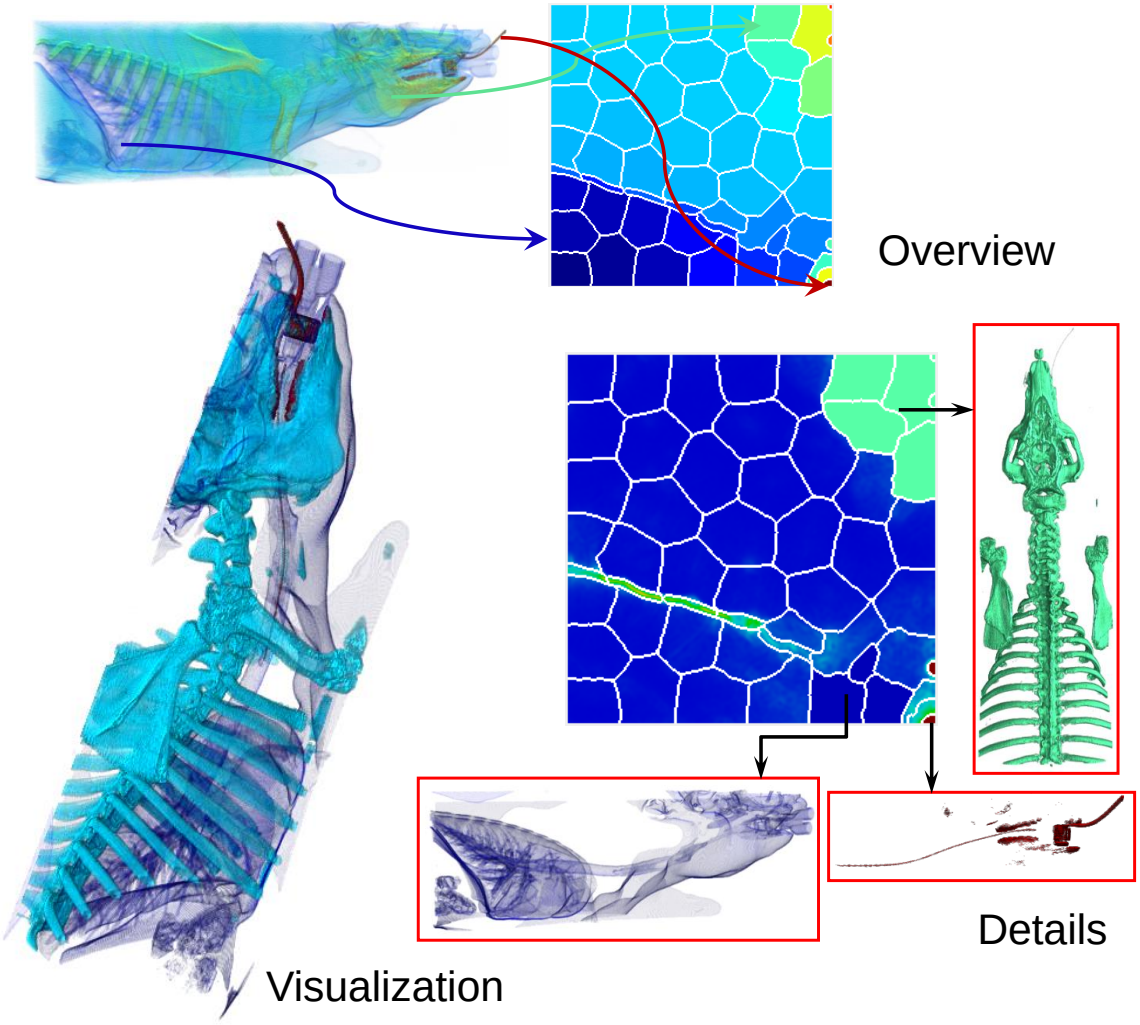


Figure 4.13: Visualization of the Pig dataset.

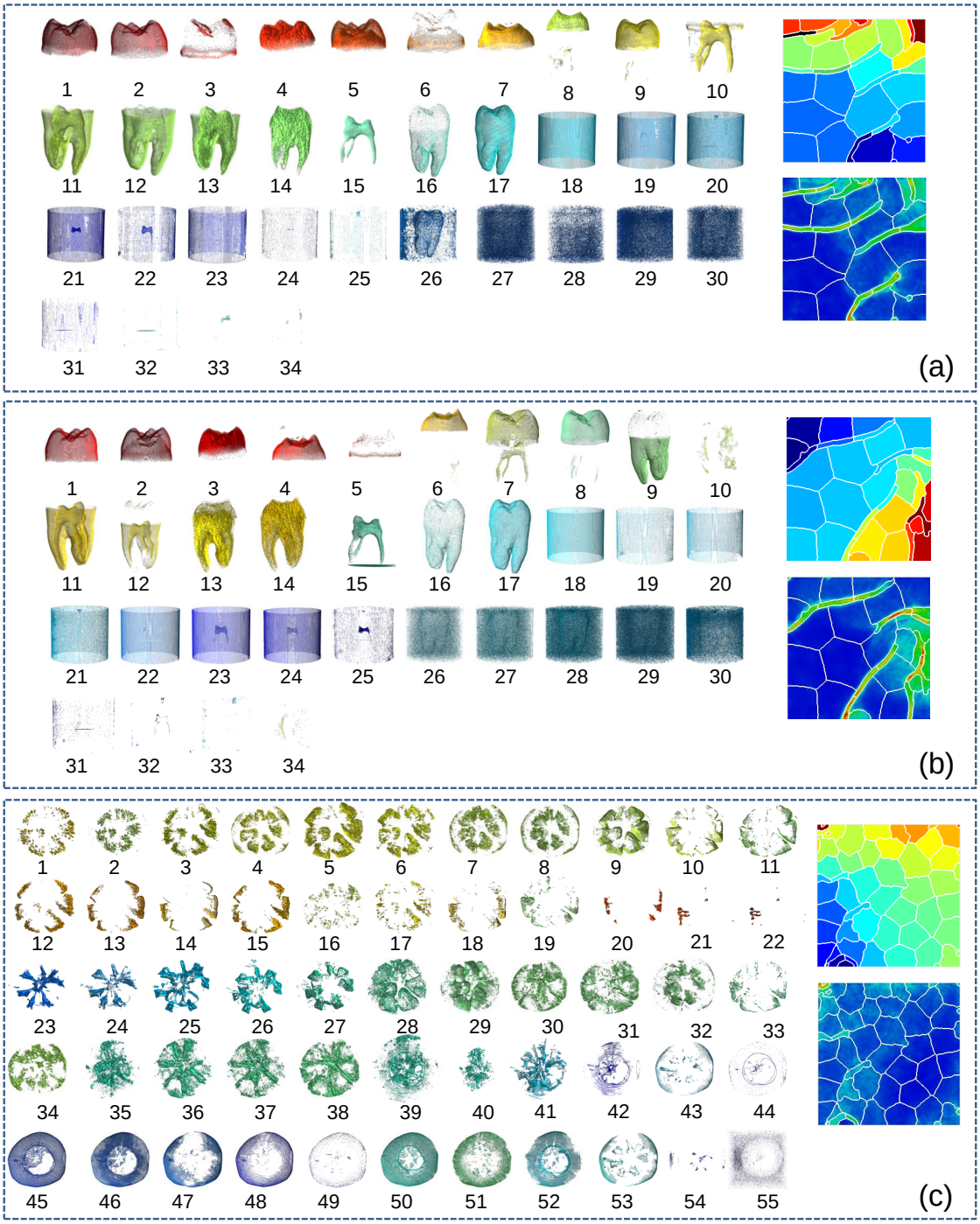


Figure 4.14: The corresponding volumetric objects of the clusters produced by the two-level clustering approach. Each subfigure consists of the collection of volumetric objects (each labeled with a number and rendered in the default color), the default colored clusters arranged on the segmented SOM and the U-matrix visualization of the SOM. (a) The Tooth dataset trained with 7 features (the same set of features as mentioned in the Results section, the 35th cluster is empty). (b) The Tooth dataset trained with 3 features (intensity, gradient magnitude and second order derivative along the gradient direction, the 35th cluster is empty). (c) The Tomato dataset.

Chapter 5

Rule-Enhanced Transfer Function Generation for Medical Volume Visualization

Medical volumetric datasets acquired from CT or MRI scans typically contain complex combinations of multiple materials. Tissues and organs frequently exhibit similar properties and overlapping data values. TF design methods based purely on the analysis of the TF feature space may fail to work well on these datasets, since various structures can be projected to the same or nearby regions of the feature space, making it difficult to separate them in this space. While high dimensional features can help to improve the accuracy of volume structure classification, it is not trivial to select features that can make an effective feature set. Moreover, in dimensional reduction based higher-dimensional TF methods, the separation ability of high dimensional features may decline after the projection. Directly manipulating and interacting with the high dimensional features space without dimensional reduction is very unintuitive and pose other challenges.

In this work, we propose a rule-enhanced TF design method to tackle this problem. A rule can be viewed as a form to encode knowledge and experience, which can then be utilized to make predictions. Our approach is motivated by the idea that segmentation data labelled in the spatial domain contains information to characterize a user-specified target tissue and to distinguish it from other tissues, and this information can be captured by a set of rules. The rules used in our method are defined based on the local frequency distribution of a voxel's neighborhood. Local frequency distribution has been shown to offer a novel and effective manner for feature identification [JH09]. In order to obtain the set of rules that are most effective in describing the patterns of the target tissue, a rule-selection method based on a genetic algorithm (GA) is proposed. A small amount of segmentation data with the target tissue labelled by the user is employed to guide the rule-selection process. The selected rules are then utilized in the TF design stage, with voxels satisfying these rules being rendered with higher opacities. The rule-enhanced TFs are able to highlight important volumetric objects in challenging datasets, which would otherwise be difficult to achieve by traditional TF design methods.

5.1 System Overview

The framework of the proposed method is presented in Fig. 5.1. As input, the user segments the target tissue on a small number (typically 2 or 3) of slices. The segmentation serves to tell the system which structure of the volume is of interest and also provides learning samples for the rule-selection procedure. The rules used in our method are defined based on the local frequency distribution of volume attributes (e.g., intensity, gradient). Each rule is represented in a similar form as a quantitative association rule for compactness purposes, while it is also straightforward to convert it into the classic *if-then* form.

The number of rules that can be generated is large, yet not all of them are useful for highlighting the target tissue. Some may in fact be distracting or ineffective. To obtain the most effective set of rules in distinguishing the target tissue, a rule-selection procedure is conducted based on a GA, which is a global optimization method that finds the solution to a problem by biologically inspired operations. The segmentation data labelled by the user is employed in the fitness evaluation process of the GA to guide the evolution towards a solution that can best distinguish the target tissue. The selected rules are then evaluated on the entire volume and connected component analysis is performed to refine the rule-evaluation results. The refined rule-evaluation results, integrated with the default transfer function, are applied to create the visualization where the target tissue is emphasized in higher opacities.

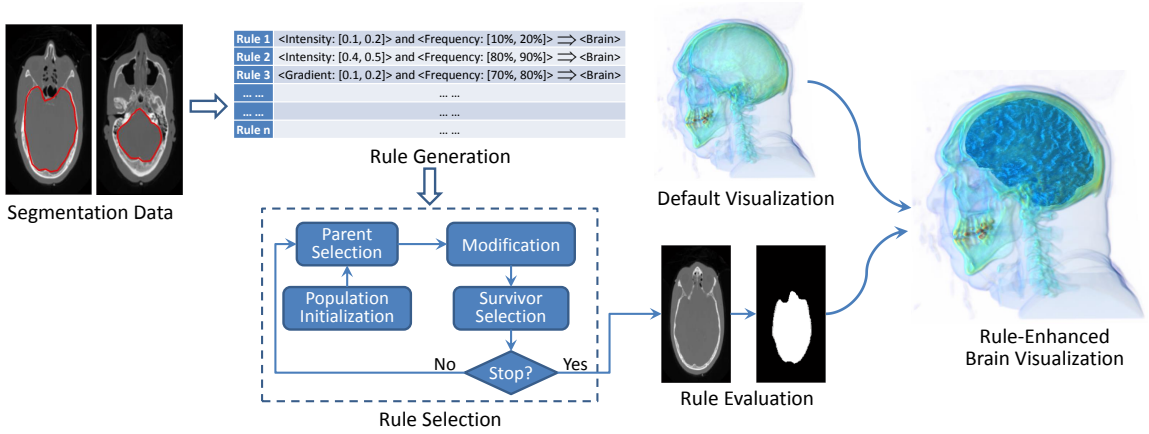


Figure 5.1: Framework of the proposed method.

5.2 Method

5.2.1 Rule Generation

The local frequency distribution captures the characteristics of a sample's neighborhood. Features of interest usually have correlated patterns in the local frequency distribution. This

motivates us to capture the correlation between the local frequency distribution and the features of interest with rules.

A rule can be represented in various forms. In the expert system, a rule is a conditional statement taking the form "if X then Y", where X is the condition and Y is the consequent. In association rule mining, a rule is an implication of the form X⇒Y, where X and Y are two sets of items. The meaning of such a rule is that whenever a transaction of the database contains X then it probably also contains Y. We can define a rule based on local frequency distribution as an *if-then* rule:

> If a voxel's local neighborhood has 10% - 20% of intensity values falling into the range [0, 0.1], then this voxel is likely to belong to a brain tumor.

The rule may also be expressed in a form similar to a quantitative association rule [SA96]:

> <Intensity: [0, 0.1]> and <Frequency:[10%, 20%]> ⇒ <Brain tumor>.

In this work, we take the latter for compactness. It can be seen that each rule defined based on the local frequency distribution consists of four fields: attribute, value range, frequency range and the target tissue. The field "attribute" refers to the property that is used to compute the local distribution. The value range is an interval where the value of the property may fall into, and the frequency range describes the percentage of voxels in a neighborhood whose values of the given attribute fall into the given value range. The last field "tissue" indicates which feature of interest this rule is correlated with.

Numerous rules can be defined in the form described above, since the value range and frequency range can take any reasonable values. To simplify the problem, we limit the number of rules by partitioning the value range and frequency range into equal intervals. Letting n_a denote the number of attributes, n_v the number of value ranges and n_f the number of frequency ranges, the number of rules that can be defined for a target tissue will be $n_a \cdot n_v \cdot n_f$.

5.2.2 Rule Selection

Not all the rules generated can be used to highlight the target tissue. We need to select the set of rules that are most effective in distinguishing the target tissue. The solution cannot be obtained by simply selecting the rules that are able to make good predictions by themselves, as the prediction results of these rules may overlap and not form a complete target tissue. On the other hand, a rule may only be able to predict a small portion of the target tissue by itself, but combining it with other rules may produce a better prediction result than any other combination. To take into consideration the performance of various rule combinations, a rule-selection procedure based on a GA is proposed.

GAs belong to the category of evolutionary algorithms, which use natural-selection-inspired

techniques to solve optimization problems. In a GA, a solution to the optimization problem is represented as a genotype. The evolution process is guided by a fitness function, which is usually the value of the objective function in the optimization problem. The algorithm consists of the following steps:

1. Population initialization: a pool of candidate solutions is initialized. The initialization can be done randomly, based on domain knowledge or include solutions created by other techniques.
2. Parent selection: the fitness of the current population is evaluated and parents are selected. Highly fit individuals have a higher chance to be selected as parents.
3. Modification: variation operators, i.e., crossover and mutation, are applied to the parents to produce offspring solutions.
4. Survivor selection: the fitness of the offspring population is evaluated and individuals that form the next generation are selected from the parent and offspring population.
5. If the stopping criterion is met, stop; otherwise, go to Step 2.

To select the most effective rules in distinguishing the target tissue, we first obtain the pool of candidate rules that can potentially separate the target tissue from other tissues. Among all the rules defined for the target tissue, a rule is considered to be a candidate rule if it is satisfied by at least one voxel of the target tissue in the segmentation data. This will remove the rules that are not met by any voxel of the target tissue. Then, the candidate rules are indexed and encoded by a binary string. Each bit of the string corresponds to one rule. A bit value of 1 indicates that the corresponding rule is effective in distinguishing the target tissue and should be selected, while a value of 0 indicates the opposite. The rule-selection problem is now equivalent to the problem of evolving the fittest string from the initial population. The fitness function, which guides the evolution process, is the key to generating desired results when using an evolution algorithm.

Intuitively, the fitness of a string should depend on how well the rules represented by the string can classify the segmentation data. We let s denote a string, $n_s(v)$ the number of rules that voxel v satisfies among the set of selected rules represented by string s (i.e., the rules whose corresponding bits are set to 1 in s), and $label(v)$ the true label of voxel v. The following sets can be defined based on $n_s(v)$ and $label(v)$ for the segmentation data:

$$\begin{aligned} \mathrm{TP}_s &= \{v : n_s(v) > 0 \text{ and } label(v) = t\}, \\ \mathrm{FP}_s &= \{v : n_s(v) > 0 \text{ and } label(v) \neq t\}, \\ \mathrm{TN}_s &= \{v : n_s(v) = 0 \text{ and } label(v) \neq t\}, \\ \mathrm{FN}_s &= \{v : n_s(v) = 0 \text{ and } label(v) = t\}, \\ \mathrm{P} = \{v : label(v) = t\}, &\mathrm{N} = \{v : label(v) \neq t\}, \end{aligned} \tag{5.1}$$

where t is the label of the target tissue. The fitness of string s is computed by the following function:

$$fitness(s) = \frac{\sum_{v \in \mathrm{TP}_s} n_s(v)}{\sum_{s \in \mathrm{TP}_s} n_s(v) + \sum_{v \in \mathrm{FP}_s} n_s(v)} + \frac{|\mathrm{TN}_s|}{|\mathrm{TN}_s| + |\mathrm{FN}_s|} - \frac{|\mathrm{FN}_s|}{|\mathrm{P}|} - \frac{|\mathrm{FP}_s|}{|\mathrm{N}|}, \tag{5.2}$$

where $|V|$ is the cardinality of a set V. The first two terms in Eq. (5.2) are essentially the positive and negative predictive values, which favor the individuals that can make correct predictions. Note that for the positive predictive value, we use $\sum_{v \in \mathrm{TP}_s} n_s(v)$ instead of $|\mathrm{TP}|$, as we want to take into consideration not only how many voxels satisfy the rules but also how many rules these voxels satisfy. The third term is the false negative rate, which is added to handle the situation where the target tissue is extremely small compared to the non-target tissues (otherwise the algorithm will tend to select the rules that classify all voxels as non-target tissues to obtain a high negative predictive value). The fourth term is the false positive rate, which, similar to the third term, handles the situation where the non-target tissues are extremely small compared to the target tissue.

5.2.3 Rule Evaluation

The fittest string evolved by the GA represents a set of rules that are effective in distinguishing the target tissue. These rules are evaluated on each voxel of the volume. Letting fs denote the fittest string and $n_{fs}(v)$ the number of rules that voxel v satisfies among the set of selected rules represented by string fs, then $n_{fs}(v)$ can be considered an indication of the voxel's likelihood of being the target tissue. However, before $n_{fs}(v)$ can be utilized for TF design, a refining step is needed, as it can happen that voxels from non-target tissues share similar local distributions with the target tissue and obtain an $n_{fs}(v) > 0$. These voxels should be removed to obtain cleaner results.

Our method to refine the rule-evaluation result is summarized in Algorithm 2. The method is based on connected component analysis, which is motivated by the fact that voxels belonging to the target tissue should form a relatively large connected component in the spatial domain. The method consists of three main steps. First, the original rule-evaluation results are thresholded to remove voxels with a low probability of being the target tissue. The threshold is determined by the average rule-evaluation value of the labelled target tissue. Second, a 2-D connected component analysis is conducted on each slice to remove small patches of misclassified voxels. Finally, a 3-D connected component analysis is conducted on the entire volume, and the overlap between these components and the labelled target tissue is computed. The component with the largest overlap is selected as the target tissue. The algorithm is illustrated by an example in Fig. 5.2.

Algorithm 2 Rule-Evaluation Result Refining Based on Connected Component Analysis

Input:
$ruleVolOri$: the original rule evaluation result of the given volume
Output:
$ruleVolRefine$: the refined rule evaluation result of the given volume
procedure
    $mean_{target} = \text{round}(\sum_{v \in P} n_{fs}(v)/|P|)$
    for each voxel of the volume **do**
        if $n_{fs}(v) < mean_{target}$ **then**
            $n_{fs}(v) \leftarrow 0$
        end if
    end for
    for each slice of the volume **do**
        Perform connected component analysis, and obtain n_{cc2D} connected components CC_{2D}
        for $i = 1$ to n_{cc2D} **do**
            $weight_{\text{2D}}(i) \leftarrow \sum_{v \in CC_{\text{2D}}(i)} n_{fs}(v)$
        end for
        Sort CC_{2D} in descending order according to $weight_{\text{2D}}$ and select the first n_{sel2D} components
    end for
    Perform connected component analysis on the 3-D volume, and obtain n_{cc3D} connected components CC_{3D}
    for $i = 1$ to n_{cc3D} **do**
        $T_{CC} \leftarrow \{v : v \in CC_{\text{3D}}(i) \text{ and } label(v) = t\}$
        $ovlRate(i) \leftarrow |T_{CC}|/|P|$
    end for
    Sort CC_{3D} in descending order according to $ovlRate$ and select the first component
end procedure

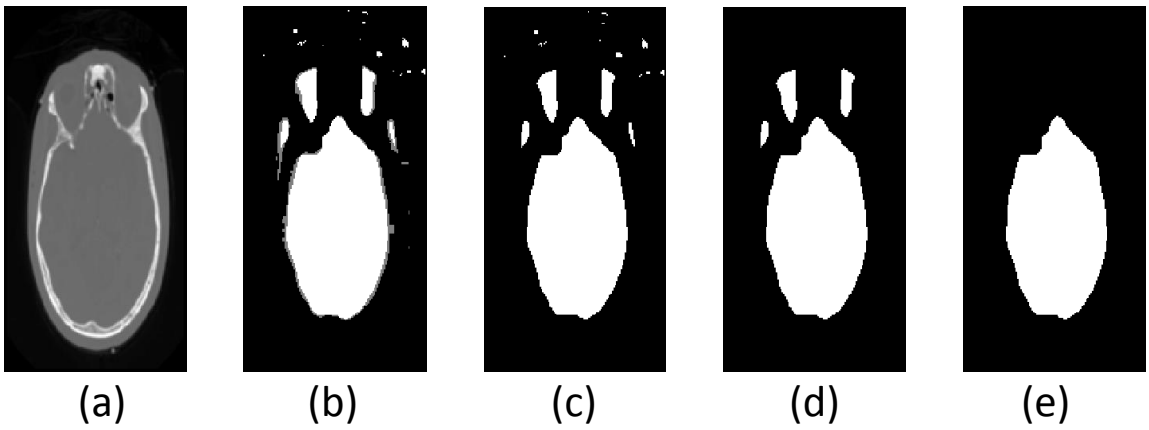


Figure 5.2: Refine rule-evaluation result by connected component analysis. (a) A slice taken from the Visible Male Head dataset; (b) Original rule-evaluation result on the slice; (c) Result after thresholding; (d) Result after 2-D connected component analysis; (e) Result after 3-D connected component analysis.

In Algorithm 2, we perform a 2-D connected component analysis before the 3-D connected component analysis. This step is useful when the target tissue is connected with other structures via small linking patches. If these patches are not removed, the following 3-D connected component analysis will fail to obtain an isolated target structure. The parameter n_{sel2D} in the 2-D connected component analysis can be automatically determined in a greedy search manner: Algorithm 2 is applied with increasing n_{sel2D}, which stops when the size of the final selected component remains unchanged for three consecutive iterations (implying that all the voxels belonging to the target tissue have been selected) or the size increases dramatically in the subsequent iteration (implying that some small linking patches have been selected and the target tissue is no longer isolated).

5.2.4 Rule-Enhanced Transfer Function Generation

The refined rule-evaluation result can now be used to generate a TF that enhances the visualization of the target tissue. In TF design, higher opacity is usually assigned to important structures so that they can stand out and are not occluded by unimportant structures. This motivates us to modulate the default opacity of each voxel v by its rule-evaluation result $n_{fs}(v)$. Specifically, the modulation is described by:

$$o_e(v) = o_d(v) \cdot (1 + \log(n_{fs}(v) + 1) \cdot k), \tag{5.3}$$

where $o_e(v)$ is the enhanced opacity of voxel v, $o_d(v)$ is the default opacity of voxel v and k is a user-controlled value to decide the extent to which the enhanced effect should be applied.

Eq. (5.3) has the following properties: 1) no enhancement will be applied to the voxels whose local distribution does not satisfy any of the rules, i.e., if $n_{fs}(v) = 0$, $o_e(v) = o_d(v)$; 2) no enhancement will be applied to the volume if the user does not want to enhance the visualization by the rule-evaluation result, i.e., if $k = 0$, $o_e(v) = o_d(v)$.

In our work, the default opacity $o_d(v)$ is set to be proportional to the voxel's intensity and modulated by gradient in order to highlight boundaries, i.e., $o_d(v) \propto \text{int}(v) \cdot \text{gm}(v)$, where $\text{int}(v), \text{gm}(v) \in [0, 1]$ are the normalized intensity and gradient magnitude of voxel v. TFs generated by other more sophisticated methods can also be easily integrated with our method, since the rule-evaluation result essentially serves as a weighting factor on the original opacity values.

5.3 Experimental Results

In this section, we demonstrate our method on a variety of volumetric datasets. For each dataset, we present the default visualization (generated by the default transfer function as described in Section 5.2.4), the rule-enhanced visualization, as well as the labelled slices on which the rules are learned, with the target tissue delineated in red.

Visible Male Head dataset ($128 \times 256 \times 256$*, 16 bit CT)*: This dataset consists of the CT scan of a male cadaver. Traditional TFs usually fail to reveal the brain due to the occlusion of the skull and the similar intensity range of the brain and other soft tissues (e.g., flesh). Our method addressed this problem by learning a set of rules that are effective in enhancing the visualization of the brain. The rules are learned on the local gradient distribution of the two labelled slices. Figure 5.3 shows the results. With the rules applied to the default visualization, the brain structure, which was previously hardly visible, is now clearly seen, resulting in a more informative visualization of the dataset compared to the default visualization.

Aneurysm dataset ($256 \times 256 \times 256$*, 8 bit CT)*: This dataset is the rotational C-arm X-ray scan of the arteries of the right half of a human head. A contrast agent was injected into the blood to highlight the aneurysm. The contrast agent results in similar intensities of the blood vessels and the aneurysm. Traditional TFs based on intensity or gradient would fail to separate the aneurysm from the vessels. This problem is solved by learning rules for the aneurysm using the local size distribution of two labelled slices. The feature size is computed based on the method proposed by Wesarg and Kirschner [WK09]. As the size of the aneurysm is larger than the vessels, the rules learned on the local distribution of feature size values are able to separate the aneurysm from the surrounding similar-intensity structures (note how the aneurysm stands out from the blood vessels in the enhanced visualization in Fig. 5.4).

MRI Woman Head dataset ($256 \times 256 \times 109$*, 16-bit MRI)*: This dataset is the MRI scan of a woman's head. While an MRI scan can reveal more detailed structures of soft tissues, these

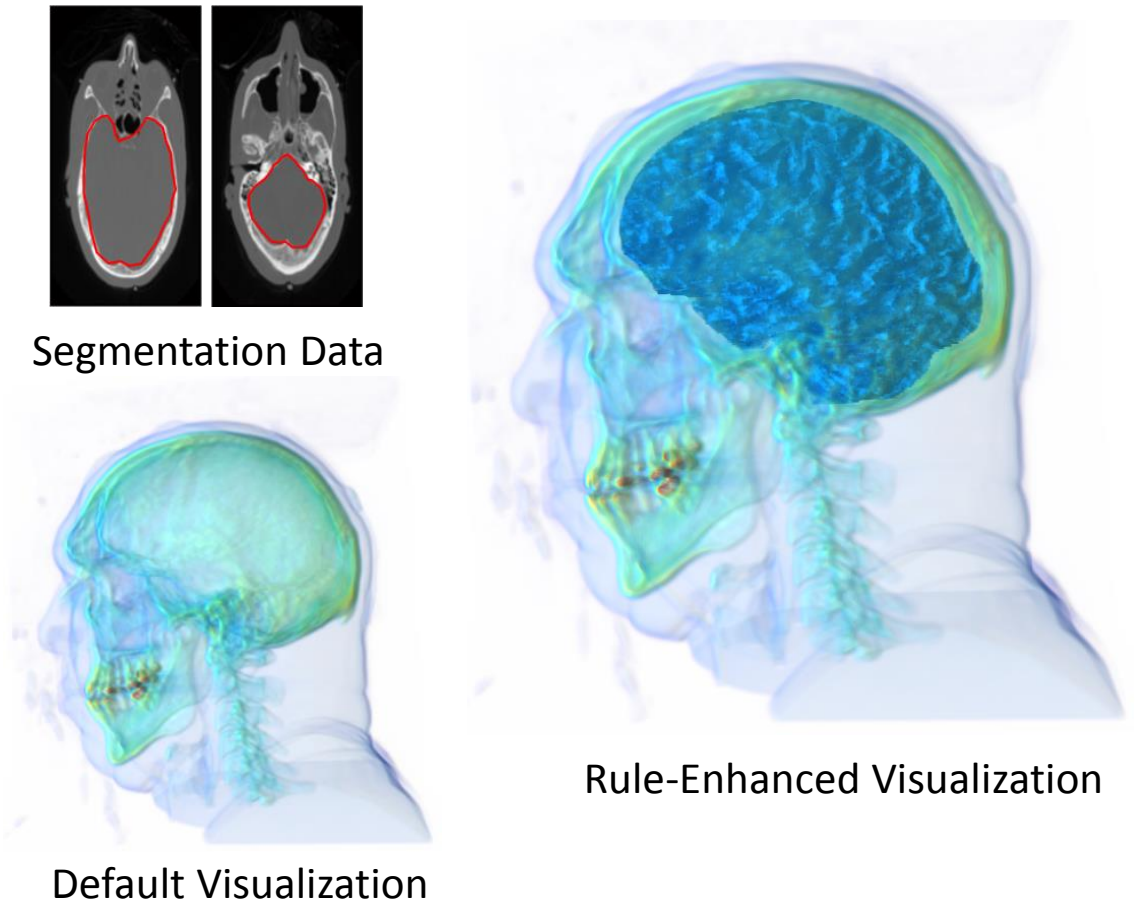


Figure 5.3: Visualization of the Visible Male Head dataset.

tissues usually obtain similar intensity values. In order to highlight the brain in the dataset, rules are learned on the local distribution of standard deviation, which is computed from a $5 \times 5 \times 3$ local window centered at each voxel. Employing mean and standard deviation as the TF space was shown to perform better than the IGM space for noisy datasets [HPB+10]. Results are presented in Fig. 5.5. The enhanced visualization can not only highlight the brain, but also clearly shows its fold structures.

Brain Tumor dataset ($512 \times 512 \times 110$*, 16-bit MRI)*: This dataset is obtained by the MRI head scan of a patient with a brain tumor. The brain tumor typically has inhomogeneous appearance and shape. Its intensity values usually overlap with other tissues, making it difficult to be effectively highlighted by traditional TFs. In order to enhance the visualization of the tumor, rules are learned from the local distribution of intensity values. The results are shown in Fig. 5.6. In the default visualization, the brain tumor is only slightly visible and its appearance cannot be viewed clearly, while in the enhanced visualization, it is clearly visualized and separated from the surrounding tissues.

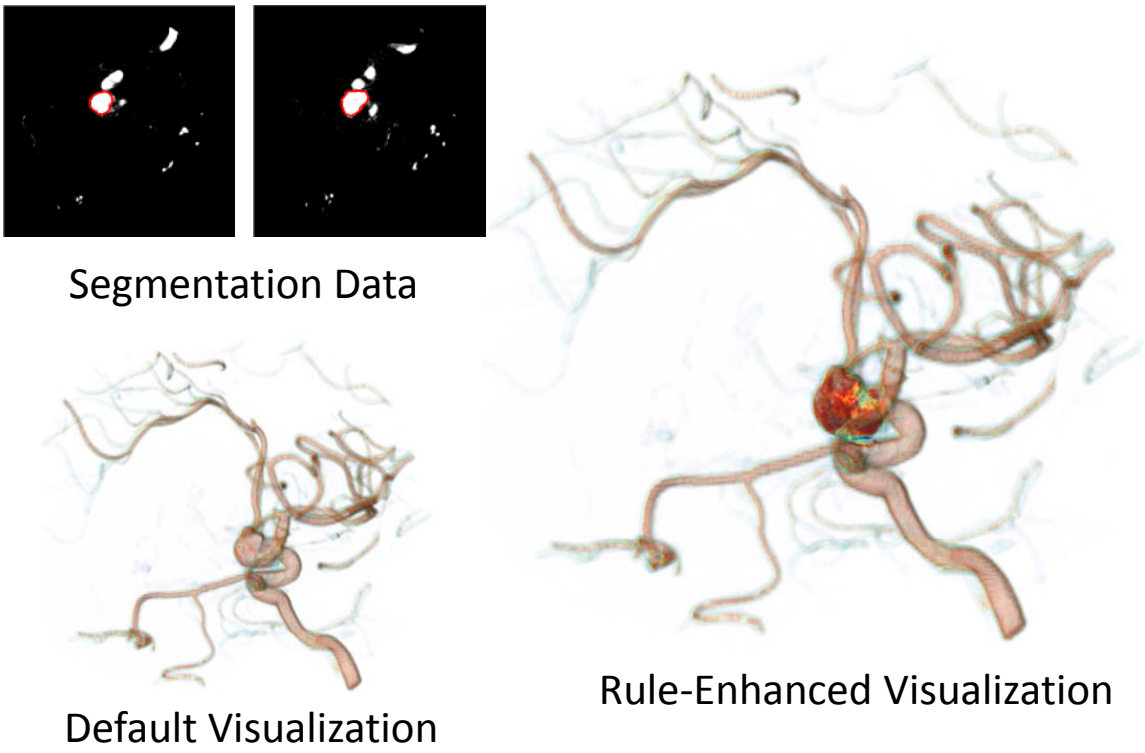


Figure 5.4: Visualization of the Aneurysm dataset.

The above datasets employ rules that are learned from the same dataset to highlight the target tissue. To generate the samples for rule learning, the user needs to manually label the target tissue on two or three slices of the volume. It will save the user labelling work if the rules learned from one dataset can be reused for another dataset to enhance the visualization of the same target tissue. An example of this is shown in Fig. 5.7 using the CT Head dataset ($256 \times 256 \times 113$, 16 bit CT). Here, the rules used to enhance the visualization of the brain are those that are learned from the Visible Male Head dataset. As the label information is unavailable, we simply select the connected component with the largest size when refining the rule-evaluation result. It can be seen that the rules learned from the Visible Male Head dataset are still effective for the brain enhancement in the CT Head dataset. This demonstrates the reusability of the rules generated by the proposed method.

Our experimental platform was a PC with an Intel i7-2600 CPU and an Nvidia GT440 graphics card. As a preprocessing step, a simple region-growing algorithm is applied to remove background voxels. For each non-background voxel, its local frequency distribution is computed. The computation of local frequency distribution is equivalent to the computation of a histogram within each non-background voxel's local neighborhood. The computation was implemented in CUDA C, with each thread attending to one voxel's local distribution computation. The GA was also implemented in CUDA C, where the process of evaluating

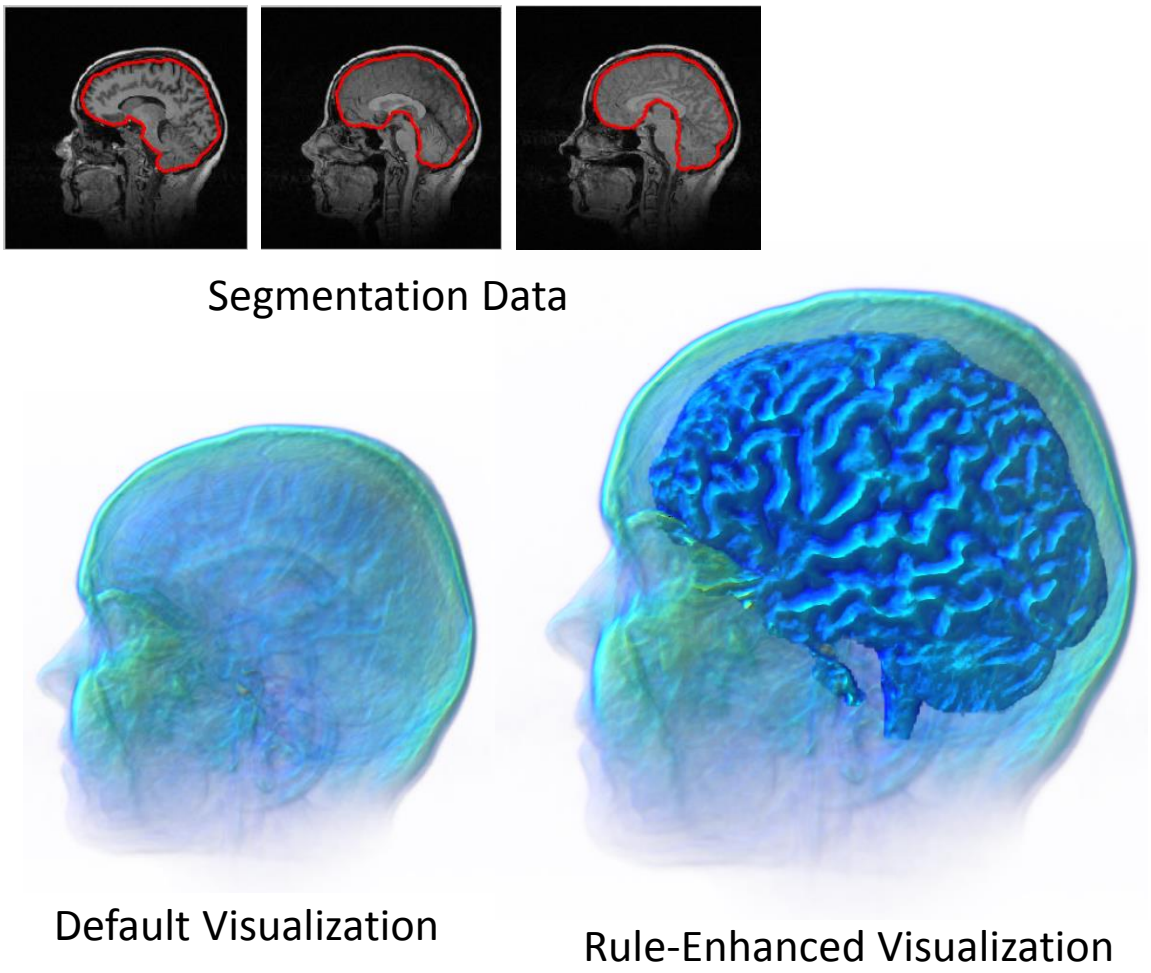


Figure 5.5: Visualization of the MRI Woman Head dataset.

a rule on the volume is parallelized by each thread evaluating the rule on one voxel. The time costs on the evaluation datasets are listed in Table 5.1. The time complexity of the rule-selection procedure by the GA is $O(n_{ls} \cdot d_g \cdot s_p \cdot n_g)$, where n_{ls} is the number of learning samples, d_g the dimension of the GA's genotype, s_p the population size of the GA and n_g the number of generations of the GA. The dimension of the GA's genotype, d_g, is equal to the number of the candidate rules, which is directly related to the total number of rules $n_a \cdot n_v \cdot n_f$ and the given volume (empirically we found that the ratio of the candidate rules among the rule base is 20% - 40%).

The parameters used in our experiment are set as follows. The local distribution is computed within a 11×11 neighborhood centered at each non-background voxel. A neighborhood of size 11×11 is large enough to capture the local distribution information while small enough to allow the extracted features to discriminate different voxels. In rule generation,

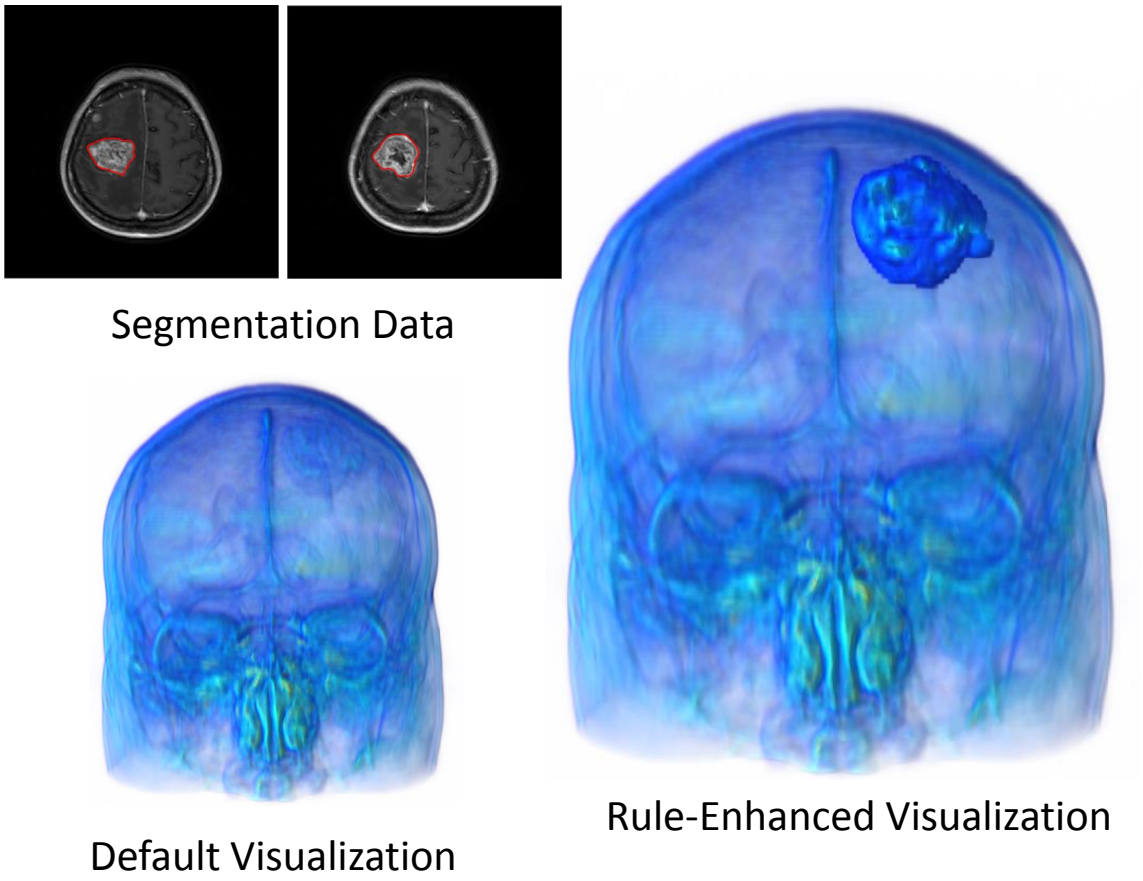


Figure 5.6: Visualization of the Brain Tumor dataset.

the number of value ranges n_v and the number of frequency ranges n_f are both set to 10. An interval number of 10 is generally sufficient for the rules to characterize different distribution patterns. For the GA, the mutation rate is set to 0.1, the crossover rate to 0.3, and the population size to 100. A small mutation rate and relatively large other parameters (i.e., the crossover rate and the initial population size) allow the evolution to explore the solution space sufficiently while converging at reasonable speed. The evolution is stopped if the maximum fitness value of each generation remains stable for 30 consecutive generations.

5.4 Discussion

In the proposed work, we enhance the visualization of important structures in the volume using the rule-evaluation results. The rules are defined based on the local frequency distribution of volume properties. The effectiveness of the local distribution in describing and visualizing volumetric features has been discussed in Johnson and Huang's work [JH09], which showed

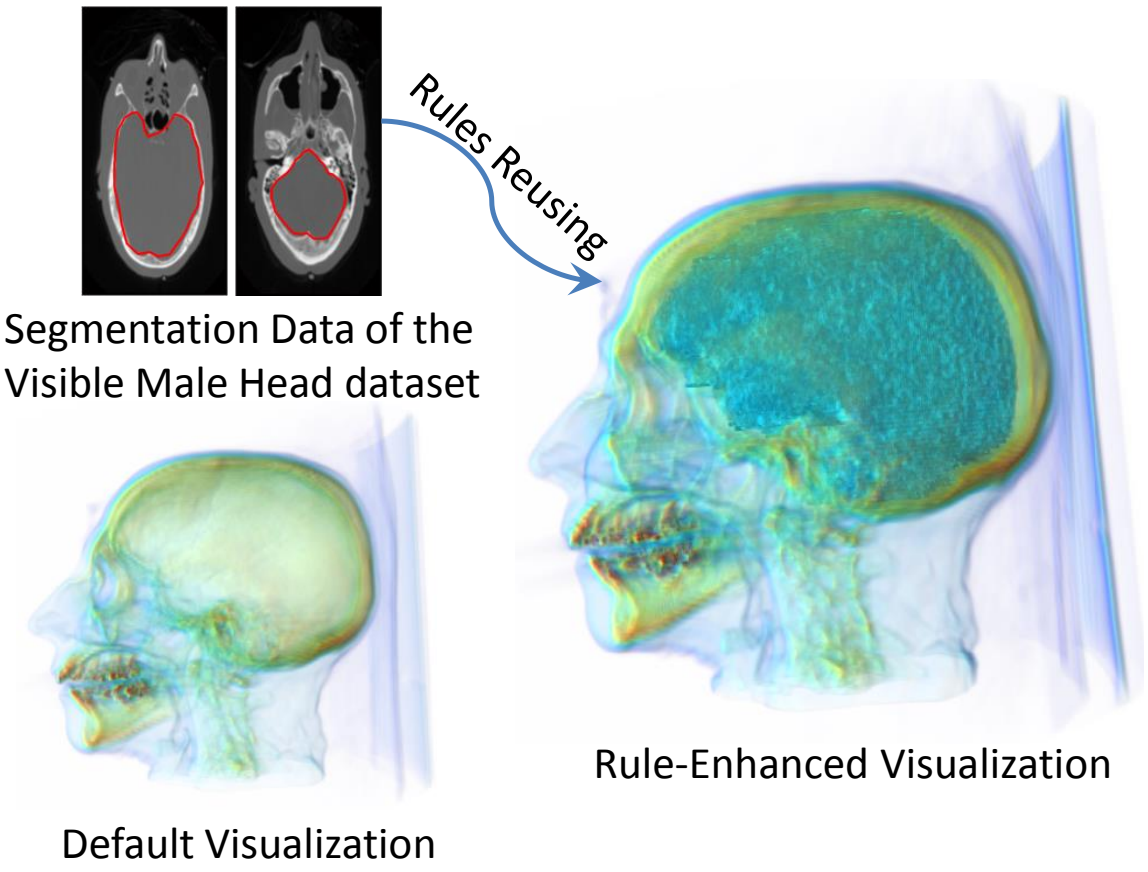


Figure 5.7: Visualization of the CT Head dataset. The rules used to enhance the visualization of the brain are those that are learned from the Visible Male Head dataset.

that using the local distribution as the basis for feature identification not only produces results that agree with and enhance traditional feature detection methods, but also reveals new features that were previously inaccessible to be visualized or not easily specified by the user. Our work drew inspiration from that work. However, in Johnson and Huang's work, users need to specify various *queries* (the counterpart of *rules* in their work) to retrieve the volume and voxels that match the queries are returned and rendered. As the local frequency distribution contains essentially low-level statistical properties, it is not straightforward to specify the appropriate queries that can return desired results. In our work, rules are automatically learned from small segmentation data using a GA, which provides a more intuitive manner to present and highlight features of interest.

Many approaches for TF generation have focused on analyzing the TF space. While exploring and visualizing a volume by manipulating the TF space allow users to view the content of the volume conveniently, its effectiveness to isolate and visualize an object of interest is compromised by the loss of information when representing a 3-D volume with a 1-D or 2-D

Dataset	Region Growing	No. of NBG Voxels	Local Distribution Computation	No. of Learning Samples	GA
Visible Male Head	4.06 s	4,116,146	0.28 s	36,672	15.4 s
Aneurysm	14.79 s	387,584	1.03 s	7,757	10.16 s
MRI Woman Head	4.91 s	2,193,048	0.20 s	91,207	174.9 s
Brain Tumor	19.03 s	9,078,816	0.59 s	149,248	108.6 s
CT Head	4.23 s	3,076,359	0.22 s	\	\

Table 5.1: Time costs, number of non-background (NBG) voxels and number of learning samples of the evaluation datasets.

TF space. The rule-based method allows for a much more effective separation of the user's interested object and a visualization that better suits the user's needs. Moreover, various properties of the target tissue can be directly inferred from the selected rules, which may also assist in the volume understanding process. For example, the rule <Gradient magnitude: [0, 0.1]> and <Frequency:[90%, 100%]> ⇒ <Brain> selected for the Visible Male Head dataset can inform the user that the brain in the dataset consists of homogeneous (small-gradient) regions.

The major limitation of our method is that an additional refining step is needed before the rule-evaluation result can be applied to enhance the visualization of the target tissue. This problem is caused by the fact that some regions of the non-target tissues have similar or the same local distributions with the target tissue, and the selected rules would lack the discriminative power to separate them. For a quantitative evaluation of the selected rules, we draw the fitness curve through the evolution process for the four evaluation datasets in Fig. 5.8. The fitness function defined by Eq. (5.2) has a maximum value of 2, obtained when all the target samples fulfill $n_s(v) > 0$ (satisfying at least one of the selected rules) and all the non-target samples fulfill $n_s(v) = 0$ (satisfying none of the selected rules). In the experiment, however, the GA usually cannot obtain this value when it converges. As shown in Fig. 5.8, the fitness values after converging vary from 1.15 to 1.71, implying the occurrence of misclassified voxels. If the misclassified non-target patches are large and are connected with the target tissue, the refining method based on connected component analysis will fail to isolate the target tissue. This can occur, for example, in some MRI brain tumor datasets, where part of the boundary between the tumor and the brain is indefinite and the tumor shares overlapping intensity ranges with the brain.

In Fig. 5.7, we show an example of reusing the rules learned from one dataset for another dataset to enhance the same target tissue in the visualization. Reusability of rules is useful as it allows rules to be stored as templates, which can then be applied to new datasets. However, the extent to which reusing rules is effective for the new datasets needs to be

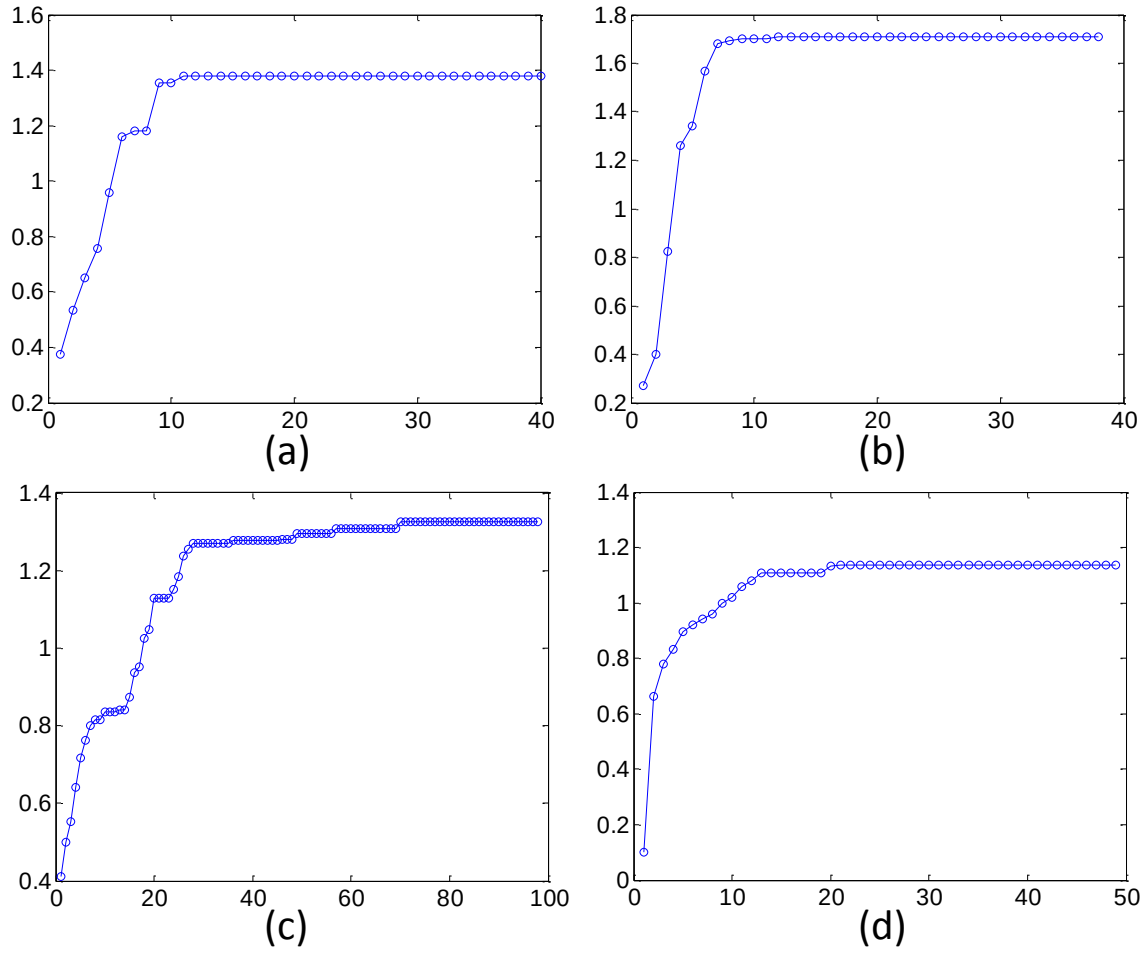


Figure 5.8: The maximum fitness value of each generation vs number of generations plot for: (a) Visible Male Head dataset; (b) Aneurysm dataset; (c) MRI Woman Head dataset; (d) Brain Tumor dataset.

considered carefully. The modalities of the Visible Male Head dataset and the CT Head dataset are both CT, hence the target tissues (i.e., the brain) have similar intensity and gradient distributions, which ensures the effectiveness of rules reusing. On the other hand, CT and MRI datasets usually have distinct features and the target tissues typically exhibit different patterns. Therefore, it would be difficult to find an attribute for the rules to be reused between CT and MRI datasets. One promising solution to this problem is the use of high-level features (e.g., size, shape), since they are less sensitive to modalities.

5.5 Summary

In this chapter, we proposed a rule-enhanced TF generation method for medical volume visualization. We defined a set of rules based on the local frequency distribution of volume

attributes. A GA was employed to select the set of rules that were most effective in distinguishing the target tissue from other tissues, and connected component analysis was used to refine the rule-evaluation results. We applied the proposed method to enhance the visualization of volumetric objects that were difficult to be separated and highlighted by traditional TF design methods. The experimental results on various volumetric datasets demonstrated the effectiveness of the proposed method.

Chapter 6

An Eye-Tracking Based Study on Visual Attention Deployment in Volume Visualization

Human viewers are on the receiving end of the visualization process. They deploy visual attention to different places of the visualization to facilitate the perception and understanding of the volume. Conversely, visualization experts take into account how viewers deploy their attention in order to create a visualization that draws attention to desired regions. An understanding of human visual attention deployment can significantly improve the efficiency and effectiveness of the visualization in conveying information to the viewer.

The eye-tracking methodology has been widely used to study saliency-based form of bottom-up attention in free-viewing natural images. Many eye-tracking datasets have been published. For example, MIT300 [JDT12] is a saliency benchmark dataset that contains 300 natural indoor and outdoor images with held-out eye-tracking data. CAT2000 [BI15] is another saliency dataset that contains 4000 images from 20 different categories (e.g., action, carton, indoor) and part of the eye-tracking data (i.e., fixations) is shared. To the best of our knowledge, such a dataset has not been developed for volume visualization images. In this work, we conduct an eye-tracking experiment to study bottom-up attention in volume visualization images to fill this gap. Such an experiment would allow us to investigate various interesting questions relevant to visualization design, e.g., which parts of the visualization images viewers will direct their attention to, and how viewers deploy their attention is influenced by the way volumetric datasets are visualized. Specifically, we propose two hypotheses that can be tested by the eye-tracking study: 1) the enhancement of opacity can help to attract more attention to the enhanced region; 2) attention distribution can indicate the visual quality of the visualization image. An investigation of these questions can provide insights into designing a visualization that guides attention to important regions more effectively.

6.1 Method

Images

70 images were collected from our own works (number of images: 46) and other published works (number of images: 24). All the images were the visualization results of volumetric datasets. Both medical and non-medical volumes were taken into consideration. The datasets used and the number of images for each dataset (shown in the parentheses) were: Visible Male Head (8), CT Carp (7), Tooth (9), Engine (6), Tomato (10), Feet (5), Foot (4), CT Head (7), Aneurism (2), MRI Woman (2), Brain Tumor (2), Pig (2), DTI (6). Multiple visualization results of the same volume may be generated with only one parameter varied (i.e., the opacity TF), or several parameters (e.g., color, opacity, lighting) varied. The images have been re-scaled so that the maximum height is 600 pixels and the maximum width is 800 pixels.

When ordering the images for displaying, we considered differential carryover effects and the counter-balancing problem for pairwise analysis of the results. As each dataset had multiple renderings and some of them were very similar, these images were placed far apart to alleviate differential carryover. At the same time, we randomized the order of the visualization images of the same dataset to counterbalance the influence of display sequence.

Participants

Ten volunteers (4 females and 6 males, age range 21 - 40) from the National University of Singapore participated in the experiment without remuneration. Each reported normal or corrected-to-normal vision.

Apparatus

The experiment was carried out in a dark, windowless room. An EyeLink 1000 eye tracker from SR Research Ltd was used to record the movement of the dominant eye at a sampling rate of 2000 Hz. The eye tracker was set up in the Desktop Mount mode, where the participant's head is supported by a chin and forehead rest to minimize the movements. Before running the experiment, a 9-point calibration was conducted to calibrate the participant's gaze position, followed by a validation step to compute the calibration accuracy. Only when the validation accuracy was rated as "good" by the system would the experiment proceed.

Images were presented on a 22-inch LCD monitor with a display resolution 800×600 and a refresh rate 120Hz . The distance between the monitor and the chin rest was 57cm. Our experimental setup is shown in Fig. 6.1.

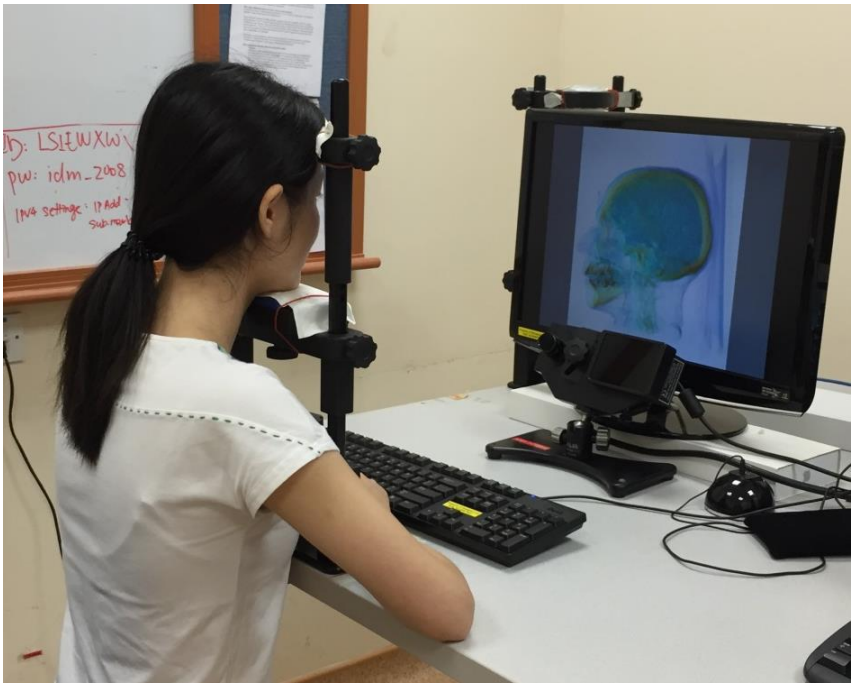


Figure 6.1: Our experimental setup for the eye-tracking study.

Procedure

The experiment was carried out individually for each participant. At the beginning of the experiment, an instruction page appeared, which asked the participants not to move their heads anymore and to press the space bar to start. Each trial started with the presentation of a fixation image lasting 2 seconds, followed by a visualization image displaying for 5 seconds. The fixation image had a gray background and a cross at the center. The participant was asked to look at the cross when it appeared. This ensured that each trial started with the participant's eyes fixated at the center of the image. Participants were told to freely view the images with no assigned goals or tasks.

Data Analysis

The EyeLink 1000 tracking system incorporates an on-line parser that analyzes and extracts eye position data into meaningful events and states, e.g., saccades, fixations, and blinks. A velocity-based algorithm was used to detect saccades, and periods between saccades were labeled as fixations. The parser used velocity and acceleration-based saccade detection methods. For each data sample, the parser computed its instantaneous velocity and acceleration. If either was above threshold, a saccade signal was generated. The velocity threshold was set to $30°/s$ and the acceleration threshold was $8000°/s^2$. Another saccadic motion threshold

was used to delay the onset of a saccade until the eye has moved significantly. This threshold was set to 0.1° of visual angle (see the EyeLink 1000 User Manual for more details).

The eye event data was output as EDF files by the EyeLink system after the recording session, which can be converted to other formats (e.g., text files) for analysis. To clearly reveal the attention distribution over an image, an individual fixation map was generated for each participant, which was a binary matrix of the same size as the image, and the value of an element was set to 1 if a fixation was directed to the corresponding location by the participant. An overall fixation map was generated by summing the individual fixation maps over all the participants, which was then blurred by a Gaussian filter with a cut-off frequency of 8 cycles per image and normalized to the range of $[0, 1]$. The resulting fixation map summarizes the attention distribution of all participants free-viewing the image and is used as the basis for the following analysis. In order to better inspect to which parts of the image viewers deploy their attention, the fixation map was mapped onto a cold-to-hot color spectrum (where hot colors represent large values) and overlaid with the stimulus image. The process is illustrated in Fig. 6.2.

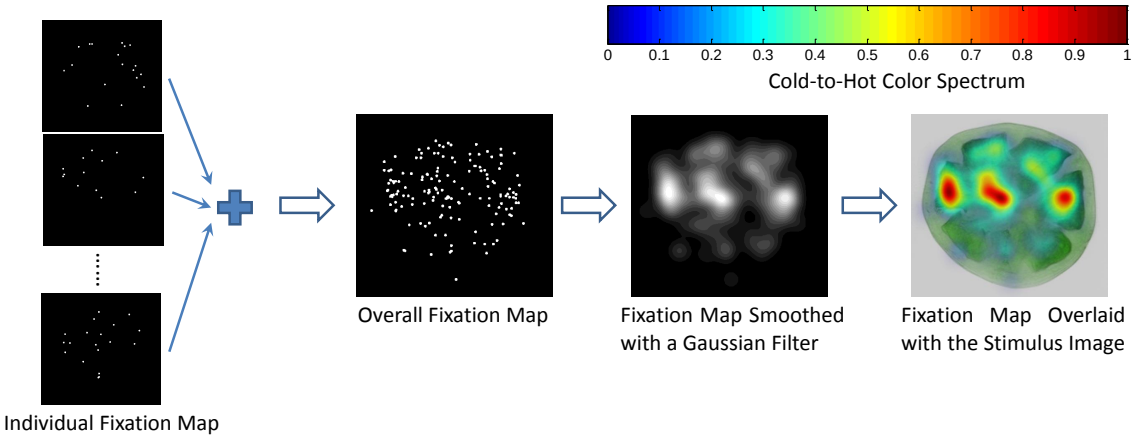


Figure 6.2: Fixation map generation pipeline.

6.2 Experimental Results

Figure 6.3 shows the fixation maps of four visualization images. It can be seen that attention is not distributed evenly within a visualization image. Instead, particular locations in the image tend to attract most attention from the viewer and appear as red spots. These locations typically have different optical properties from the surrounding structures. For example, the teeth of the Visible Male Head dataset is colored differently from the surrounding bone, the seeds in the Tomato dataset are rendered in higher opacities than the surrounding pulp, the left ankle of the Visible Female Feet dataset has distinct colors from the bone and skin, and the air bladder of the Carp dataset has clear boundaries separating it from the surrounding.

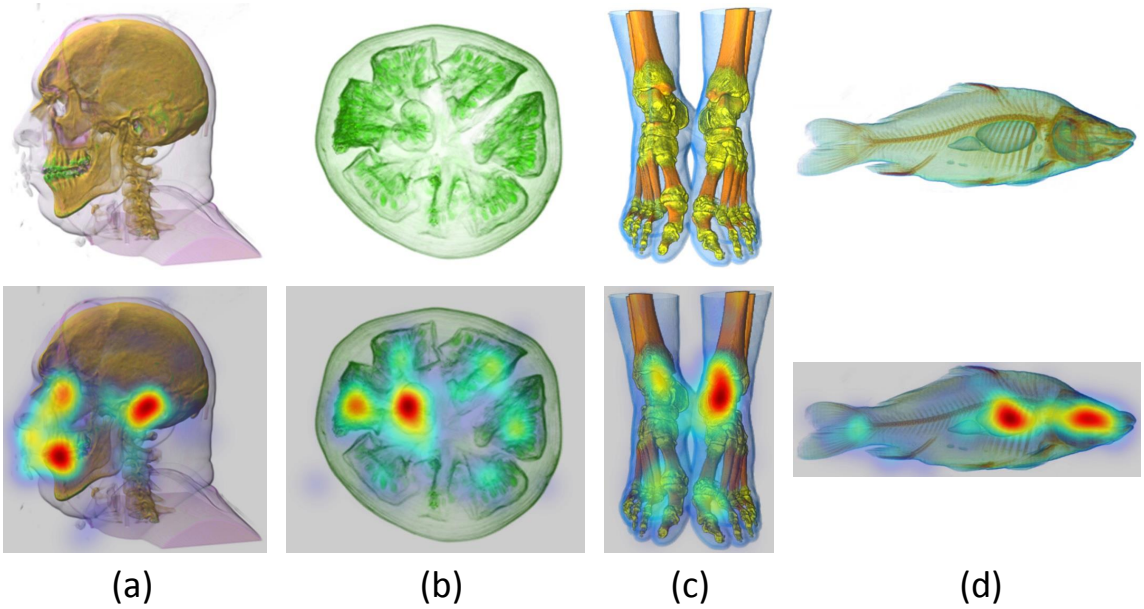


Figure 6.3: Attention distribution in volume visualization images. Top row: original visualization images; bottom row: visualization images overlaid with fixation maps. (a): the Visible Male Head dataset; (b) the Tomato dataset; (c) the Visible Female Feet dataset; (d) the Carp dataset.

In the following, the experimental results will be discussed with respect to the two hypotheses proposed in the beginning.

Hypothesis 1: The enhancement of opacity can help to attract more attention to the enhanced region The higher the opacity is, the more visually prominent a volumetric object appears on the rendered image. This object may be experienced as being more salient by the viewer and thus draw more visual attention. To test this hypothesis, we first compare the attention distribution in default visualization vs rule-enhanced visualization on the five evaluation datasets in Chapter 5. Here, the opacity for the object of interest is enhanced by rule evaluation results. From Fig. 6.4, it can be seen that for the Brain Tumor, MRI Woman Head and Visible Male Head datasets, the adjustment of opacity for the target tissue can effectively attract more attention to it. This effect is less obvious for the Aneurysm dataset, since the target tissue (i.e., the aneurysm) in the default visualization already attracts the greatest amount of attention. For the CT Head dataset, the enhancement of the brain does not help to draw more attention to that region, probably due to the fact that the brain appears homogeneous and does not exhibit clear folding structures unlike the MRI Woman Head and Visible Male Head datasets.

The results in Fig. 6.4 imply that while the enhancement of opacity can significantly attract more attention to the target tissue in some datasets, it may fail to achieve the same goal

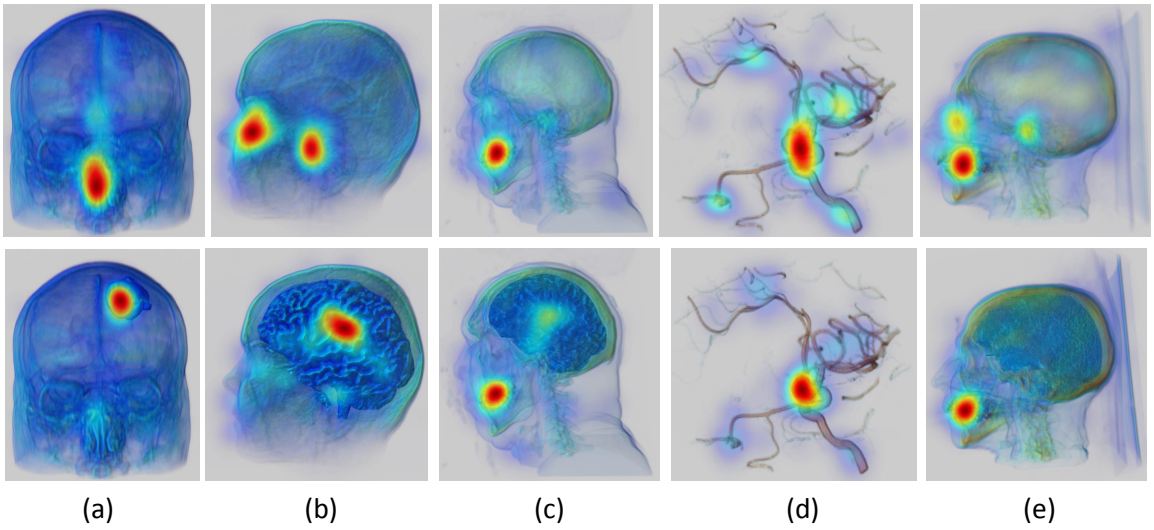


Figure 6.4: Comparing visual attention distribution for default vs rule-enhanced visualization. Top row: attention distribution of default visualization; bottom row: attention distribution of rule-enhanced visualization. (a) the Brain Tumor dataset; (b) the MRI Woman Head dataset; (c) the Visible Male Head dataset; (d) the Aneurysm dataset; (e) the CT Head dataset.

for other datasets. This is further illustrated by another example in Fig. 6.5. Here, the four visualizations of the Visible Male Head dataset were generated with the same set of parameters except for the threshold for zero target visibility, which is increased gradually from left to right and thus allocates more and more visibility (i.e., opacity) to the skull region (see Chapter 3 for more details). It can be seen that the skull region receives little attention though more and more visibility is assigned to it as the threshold value increases. Instead, attention tends to concentrate in three regions that exhibit strong contrast with the context, i.e., teeth, otic bone, frontal and maxillary sinuses for Figs. 6.5(a)(b), and the eye position for Figs. 6.5(c)(d). Also, the attention percentage within each region differs. For example, the attention directed to the sinuses decreases as the threshold increases, since more visibility is allocated to larger structures and the contrast between the sinuses and its context decreases.

Hypothesis 2: Attention distribution can indicate the visual quality of the visualization image The way a volume is visualized will directly determine how it is to be viewed. Conversely, how the rendered image is viewed would reflect how the volume is visualized. Our second hypothesis is that attention distribution can indicate the visual quality of the visualization image. In Fig. 6.5, we have shown that the attention percentage within an area of interest (AOI) differs among different visualizations. Generally, the more clearly and saliently an AOI is visualized, the more attention will be paid to it. Therefore,

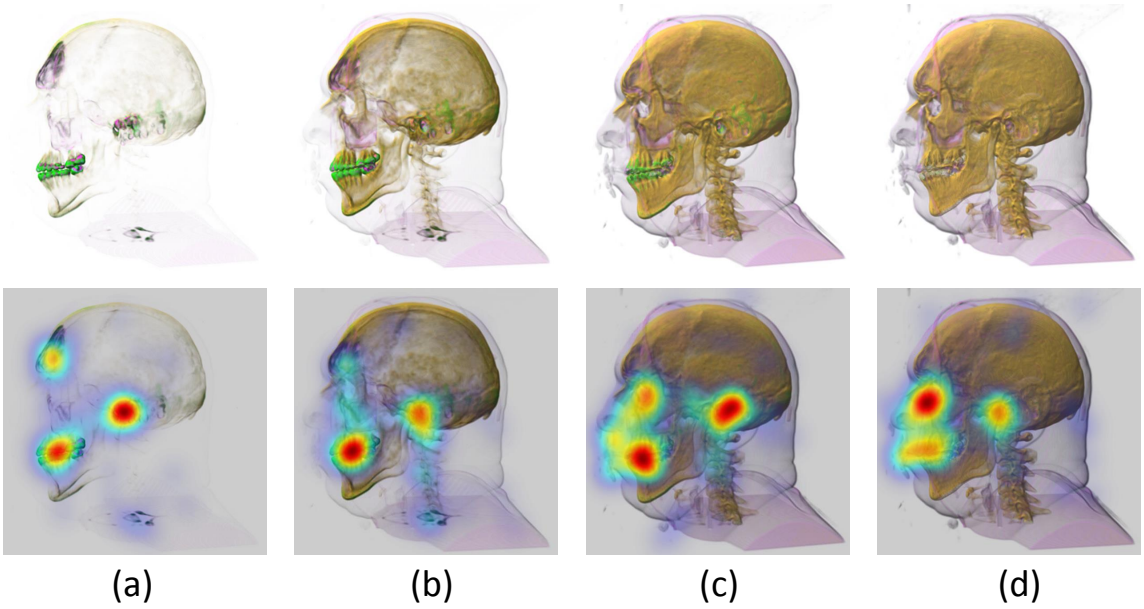


Figure 6.5: Comparing visual attention distribution for different visualizations of the Visible Male Head dataset. The dataset is visualized by the method introduced in Chapter 3. Top row: original visualization images; bottom row: visualization images overlaid with fixation maps. (a) Threshold = 10^{-7} × volume size; (b) Threshold = 10^{-6} × volume size; (c) Threshold = 10^{-5} × volume size; (d) Threshold = 10^{-4} × volume size.

the attention percentage within an AOI may be used as an indication of how well the AOI is visualized by the corresponding visualization method, which is computed as:

$$ap = \frac{\sum_{(x,y)\in\mathcal{A}} fm(x,y)}{\sum_{(x,y)\in\mathcal{M})} fm(x,y)}, \tag{6.1}$$

where fm is the fixation map, (x, y) is a position on the map, $\mathcal{A}$ is the set of coordinates within the AOI and $\mathcal{M}$ is the set of all feasible coordinates for the fixation map.

We test this hypothesis on the Visible Female Feet dataset, where the AOI is specified as the joint region of the left foot and the visualization results are ordered according to the attention percentage within the AOI. The attention percentages of the rendered images generated by various visualization methods are listed in Table 6.1. Figure 6.6 gives the ranking results of the visualization images generated in our own works (rendered images published in other works cannot be placed here due to copyright issues). The ranking results seem to be consistent with subjective visual quality assessment.

The hypothesis is tested on the visualization results of the Tooth dataset in Table 6.2 and Fig. 6.7 gives the ranking results of the images generated in our own works. The AOI is specified as the boundary between the enamel and dentine. For the top items of the ranking

Rank	1	2	3	4	5
Method	Fig. 6.6(a)	[WZJ+12]	[WCZ+11]	[MCWE09]	Fig. 6.6(b)
ap	14.87%	7.76%	7.12%	6.41%	3.50%

Table 6.1: Ranking the visualization results of the Visible Female Feet dataset according to the attention percentage within the AOI.

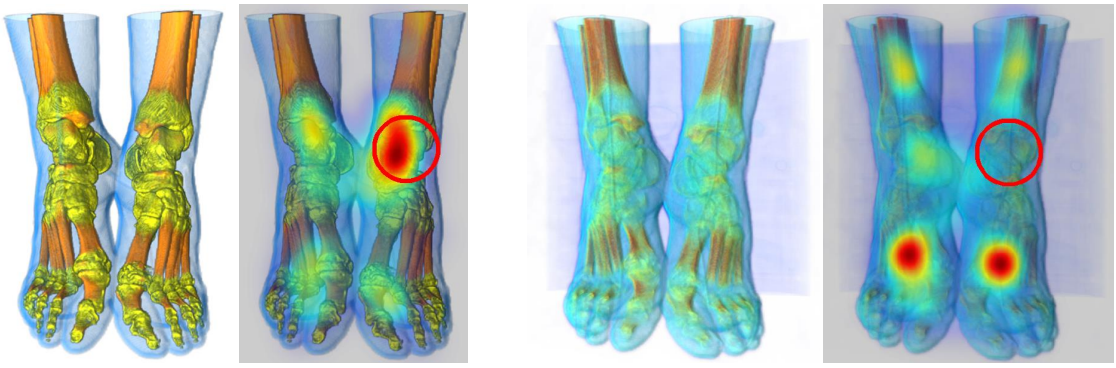


Figure 6.6: Ranking the visualization results generated in our own works for the Visible Female Feet dataset according to the attention percentage (indicated by the numbers below) within the AOI (specified by the red circles).

list, the AOI is clearly visualized, while for the bottom items, the AOI is hardly visible. It is interesting to note that the bottom item in Table 6.2 (i.e., [WCS+10]) actually performs well with regards to other quality metrics, e.g., sharpness, contrast, brightness and color balance. However, visualization is not just about creating pretty pictures. One primary function of visualization is to deliver important features of the dataset to the viewer. If a visualization cannot clearly reveal these features, it cannot be rated as good. From this perspective, attention percentages can serve as a complementary metric to other image quality metrics and evaluate the visual quality of the visualization from a novel aspect.

Rank	1	2	3	4	5	6	7	8	9
Method	[IVJ12]	Fig. 6.7(a)	[CM10]	Fig. 6.7(b)	[WZJ+12]	[vBSG06]	[RBBV11]	Fig. 6.7(c)	[WCS+10]
ap	31.61%	30.98%	28.84%	25.58%	21.36%	17.72%	16.82%	16.16%	8.03%

Table 6.2: Ranking the visualization results of the Tooth dataset according to the attention percentage within the AOI.

Table 6.3 presents another example on the Carp dataset and Fig. 6.8 gives the ranking results of the images generated in our own works. Here, the air bladder of the Carp dataset

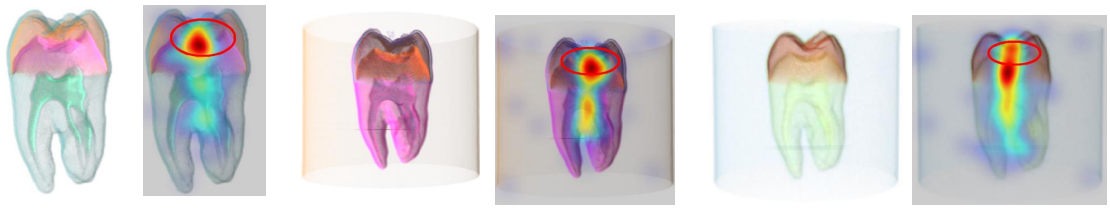


Figure 6.7: Ranking the visualization results generated in our own works for the Tooth dataset according to the attention percentage within the AOI.

is specified as the AOI. It can be seen that the attention percentages within the AOI do correlate with the visual quality of the AOI. Comparing the first two visualizations with the last two in Table 6.3, it is obvious that the air bladder region in the former is visualized more clearly than those in the latter. This suggests that the attention percentage within an AOI can provide useful information for visual quality evaluation.

Rank	1	2	3	4	5	6	7
Method	[KKG15]	Fig. 6.8(a)	[WQ07]	Fig. 6.8(b)	[WCZ+11]	[RBBV11]	[PF07]
ap	26.82%	25.29%	24.24%	21.91%	20.56%	16.74%	13.44%

Table 6.3: Ranking the visualization results of the Carp dataset according to the attention percentage within the air bladder region.

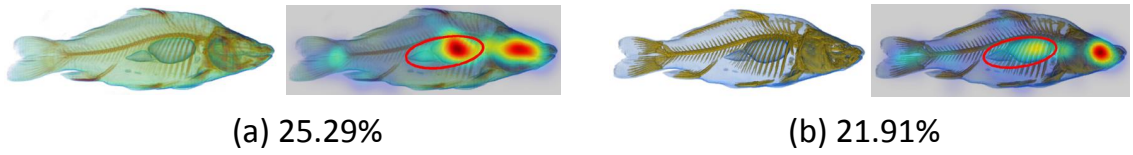


Figure 6.8: Ranking the visualization results generated in our own works for the Carp dataset according to the attention percentage within the air bladder region.

Since visual attention is a limited resource, allocating more attention to one location will result in less to others. Also, a visualization method may be able to visualize one part of the dataset clearly but fail to do so for other parts. Therefore, it is natural that different AOIs will give different ranking results. Table 6.4 presents the ranking results where the head of the carp is specified as the AOI. Discrepancy between Table 6.3 and Table 6.4 can be observed.

In many situations, the visualization designer knows which parts of the visualization are important and should draw attention from the viewer. An importance factor can be assigned to each AOI. We propose the final visual quality metric as a weighted sum of the attention

Rank	1	2	3	4	5	6	7
Method	[PF07]	[WQ07]	[WCZ+11]	Fig. 6.8(a)	[KKG15]	Fig. 6.8(b)	[RBBV11]
ap	38.83%	38.32%	35.60%	33.05%	31.98%	29.74%	25.49%

Table 6.4: Ranking the visualization results of the Carp dataset according to the attention percentage within the head region.

percentages:

$$Q = \sum_{i \in \mathcal{I}} imp_i \cdot ap_i, \tag{6.2}$$

where imp_i is the importance factor assigned to the i_{th} AOI and ap_i is the attention percentage within the i_{th} AOI. Equation 6.2 implies that the attention percentages of AOIs with higher importance values will contribute more to the quality metric. Table 6.5 displays the ranking results of the Carp datset according to the proposed quality metric, where the air bladder is assigned an importance factor of 0.7 and the head region 0.3. Though a user study is currently lacking to verify the proposed metric, readers should be able to find some consistency between the proposed ranking and human visual perception.

Rank	1	2	3	4	5	6	7
Method	[WQ07]	[KKG15]	Fig. 6.8(a)	[WCZ+11]	Fig. 6.8(b)	[PF07]	[RBBV11]
ap	28.46%	28.36%	27.62%	25.07%	24.26%	21.05%	19.37%

Table 6.5: The final ranking of the Carp dataset visualization results according to the proposed quality metric.

6.3 Discussion

In Fig. 6.3, it is shown that attention is not distributed evenly within a visualization image, but is directed to locations where local visual attributes differs significantly from the context. This is consistent with the findings from previous works, which stated that what matters in guiding bottom-up attention is local feature contrast rather than local absolute feature strength [IK01]. Hence the guideline for visualization experts is to enhance the local feature contrast of a region if they want to guide more attention to that region. It should be noted this cannot be always achieved by the adjustment of opacity, since opacity may only serve to enhance local feature strength when the region is large and appears homogeneous. This is illustrated by the brain region in Fig. 6.4(e) and the skull region in Fig. 6.5, which do not receive much attention though the feature strength (i.e., opacity) is enhanced, as the skull

and the brain appear as homogeneous regions and do not stand out from the context. This suggests that adjusting opacity alone does not always help to enhance saliency for the object of interest and we need to rely on other optical properties, e.g., color, lighting and shading, to bring out the object of interest. This justifies the extensive study on achieving focus+context visualization via the use of various graphics resources as reviewed in Section 2.1.1.

Our second hypothesis explores the possibility of evaluating the visual quality of a visualization by how much attention falls into desired regions. This is inspired by the fact that a good visualization should be able to draw viewer attention to important regions that the visualizer wants to highlight. Analysis based on the attention percentage within an AOI reveals that attention distribution can be potentially utilized for objective visual quality assessment in volume visualization, as the ranking results based on the proposed metrics seem to be consistent with human visual quality assessment. Nevertheless, a user study, where users are asked to rate the visualization images according to visual quality, should be conducted to validate the correlation between the rating scores given by human viewers and those derived from attention distribution.

Ranking the visualizations based on the percentage of attention within an AOI is similar to the idea of measuring visualization quality by how well visual attention matches with data importance, which has been investigated by Jänicke and Chen [JC10]. In their work, a user-defined data importance mask was compared to a saliency map to decide whether the visualization can guide the deployment of attention appropriately. The saliency map used in their work was generated using the Itti algorithm [IKN98]. While a computational saliency model helps to predict human attention without conducting a user study, its prediction is not always consistent with human behavior. Examples are given in Fig. 6.9. The first and second columns of Fig. 6.9 display the ground truth fixation maps and saliency maps predicted by the Itti model. It can be seen that the latter tends to elicit strong response at many locations that do not actually draw visual attention. The saliency maps estimated by another two saliency models are presented in the third and fourth columns. These two models are the BMS model [ZS13] and the SALICON model [JHDZ15], whose prediction accuracy currently ranks first and third on the MIT Saliency Benchmark. The discrepancy between the models' prediction and ground truth fixation data can be easily observed. For example, the BMS model predicts strong saliency for the bone region in BMS(a), while the ground truth shows that most attention is actually attracted to the joints. The SALICON model successfully predicts that the joints in SALICON(a)(b) should attract most attention, but the attention percentage drawn to each of the joint regions is not consistent with ground truth. These results imply that these computational saliency models may not be a good choice for attention prediction in visualization images, and visualization quality evaluation based on such models risks producing misleading results.

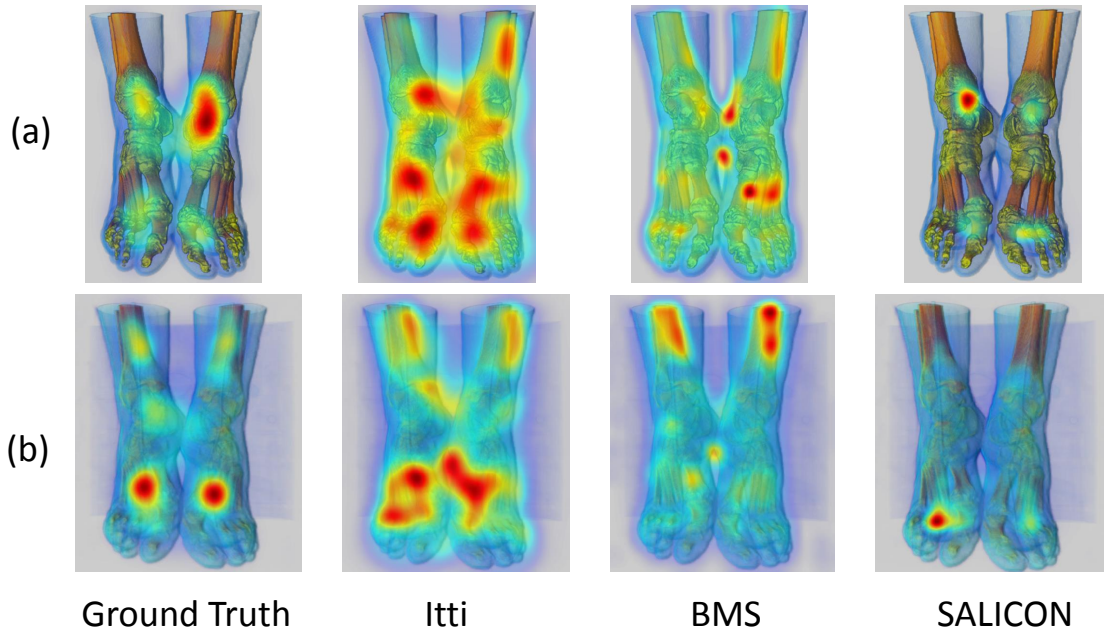


Figure 6.9: Comparing the expected attention distribution predicted by computational saliency models with ground truth fixation data. First column: ground truth fixation maps. Second column: saliency maps predicted by the Itti model [IKN98]. Third column: saliency maps predicted by the BMS model [ZS13]. Fourth column: saliency maps predicted by the SALICON model [JHDZ15].

6.4 Summary

In this work, an eye tracking experiment was conducted to study bottom-up visual attention deployment in volume visualization images. A fixation map that summarized the attention distribution of all participants free-viewing the stimulus image was generated by smoothing the overall fixation matrix with Gaussian filters. By analyzing and comparing the fixation maps of different visualization images, we arrive at the following findings:

1. Attention is not distributed evenly within a visualization image, but directed to a few salient locations of the image.
2. The saliency of a volumetric object is determined largely by local feature contrast rather than local absolute feature strength. The enhancement of opacity can not always help to attract more attention to the enhanced region, since it may only serve to enhance local feature strength when the region is large and appears homogeneous.
3. Attention distribution can indicate visual quality of the visualization image. Specifically, the attention percentage within an AOI can be used as an indication of how well the AOI is visualized by the corresponding visualization method.

4. The expected attention distribution predicted by computational saliency models is not always consistent with human's attentional deployment behavior in viewing volume visualization images.

Chapter 7

Conclusions and Future Work

In this thesis, three TF design methods were investigated aiming at better identifying and highlighting important structures in medical volumetric datasets. The first TF method introduced in Chapter 3 derived the opacity TF automatically by matching the observed visibility distribution to an importance-based target visibility distribution. The proposed method also presented an intuitive framework for clinicians to apply their domain knowledge to manipulate TFs. The experimental results demonstrated that the proposed visibility distribution-based TF design method can reveal important structures of the volume more clearly compared to assigning opacity directly from voxel intensity and gradient magnitude values.

The advantages of the first TF method is that it can generate an informative visualization of the volume with minimal user intervention and allows users to intuitively incorporate their domain knowledge in defining target visibility distributions. However, the classification accuracy of a 2-D TF is limited by the number of attributes used. This motivates us to extend the TF domain into higher dimensions in our second method. In Chapter 4, a two-level clustering approach was proposed to address the higher-dimensional TF design problem, where the first-level clustering by SOM served to project the high-dimensional feature data onto a low-dimensional space, and the second-level clustering by the normalized cut produced a manageable number of clusters to facilitate the detection and separation of meaningful volumetric objects in the TF space. The high-dimensional TF was shown to provide better separability of volume structures when compared with other TF design methods, and the proposed exploration scheme with top-down navigation was shown to provide useful guidance to the user exploring the volume.

While high dimensional features can generally help to improve the accuracy of volume structure classification, some volumetric objects may still not be separated well. The reason may be that the features are not distinct enough or the separation ability of high dimensional feature drops when projecting it to a lower-dimensional space. This causes voxels from different volumetric objects to be projected the same or nearby regions of the TF space, thus making it difficult for a TF to distinguish them. In the third method as presented in Chapter 5, we circumvented the challenges of TF design in a TF space. Instead, a set of rules

were defined based on the local frequency distribution of volume attributes. A rule-selection method based on a GA was proposed to learn the set of rules that can best distinguish the target tissue based on a small amount of segmentation data labelled in the spatial domain. In the rendering stage, voxels satisfying the selected rules were rendered with higher opacities in order to highlight the target tissue. The rule-based method allowed important structures in difficult datasets to be more effectively separated and highlighted.

Apart from the three TF design methods, this thesis also investigated on visual attention deployment in volume visualization based on an eye-tracking experiment, where eye movement data was collected from ten participants free-viewing volume visualization images. The results were analyzed based on a fixation map that summarized the attention distribution of all participants free-viewing the image. By inspecting and comparing the fixation maps of different visualization images, we found that attention was not distributed evenly within a visualization image, but was directed to locations where local visual attributes differed significantly from the context. Therefore, if visualization experts want to guide attention to a particular region in their visualization, they would need to enhance the local feature contrast of that region. We also explored the possibility of evaluating visualization quality by how well visual attention matched with data importance and pointed out that current computational saliency models may not be a good choice for attention prediction in volume visualization images.

7.1 Future Work

The automatic TF design method described in Chapter 3 operates in the most commonly used 2-D TF space, i.e., the intensity and gradient magnitude histogram. The algorithm can be extended to operate in other 2-D TF spaces, which would allow similar regions in difficult datasets to be more easily discriminated. It is also possible to extend the algorithm to perform in higher-dimensional TF space. The challenge here is that the number of bins in computing visibility distributions increases exponentially with the number of attributed used. For example, if seven attributes are used and each attribute is quantized into 256 bins, the dimension of the visibility histogram will be 256^7, leading to very expensive computational and storage costs. One solution would be to use fewer bins for less important attributes, but this is non-trivial and requires the investigation into feature importance.

The two-level clustering-based high-dimensional TF design method can take into consideration more features in the training of the SOM, which can further increase the separation ability of the clusters. As the number of features increases, selecting features empirically as adopted in current work would be of low efficiency, thus requiring investigation into an effective feature selection algorithm for volume data clustering. The goal is to find a subset of features with maximum relevance and minimum redundancy. There are two basic approaches to

perform feature selection for clustering: a wrapper approach, where the clustering algorithm is wrapped into the feature selection procedure to evaluate a feature subset, and a filter approach, where the feature set is filtered without applying the clustering algorithm. Due to the large size of volume data, a filter approach without running the clustering algorithm in the selection process will be more efficient. The challenge is to evaluate the feature subset by some criteria that will affect the ultimate performance of the clustering algorithm. Another direction of advancement is to study an automatic method to determine the suitable number of clusters for each volumetric dataset, which can increase the flexibility of the system.

For the rule-enhanced TF design method, it would be meaningful to explore methods other than local frequency distribution for rule definition, which would allow the target tissue to be more effectively separated and eliminate the additional step to refine the rule-evaluation result. Also, in the current rule selection method, each candidate rule takes up one bit in the genotype of the GA and the search space will increase exponentially with the number of candidate rules increasing. It would be useful to investigate other methods for rule selection that can perform more effectively on a large rule base.

Measuring visualization quality by how well visual attention matches with data importance is an interesting direction for future investigation. It can not only serve as a means for visual quality evaluation, but also provide feedback on improving an existing visualization. Ranking the visualizations based on the attention percentage within an AOI is a simple implementation of this idea. Future work can investigate on more sophisticated methods to define AOI, data importance and scoring metrics. A user study, where users are asked to rate the visualization images according to visual quality, can be conducted to validate the correlation between the rating scores given by human viewers and those derived from attention distribution. Another potential work is to develop a more precise computational saliency model to predict visual attention in volume visualization images. This model can leverage on the fact that the visualization images are generated from a 3-D volume via the volume rendering pipeline, which implies that not only the output image, but also the information in other steps of the rendering pipeline can be utilized for saliency prediction. For example, the shading step adds lighting effect to the rendered image and the parts shaded with more details look more realistic and thus attract more attention. This can be considered by the saliency model to improve the prediction accuracy.

Bibliography

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, pages 207–216. ACM, 1993.

[AS+94] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proceedings of 20th International Conferences on Very Large Data Bases (VLDB)*, volume 1215, pages 487–499, 1994.

[BBR08] Dominik Brugger, Martin Bogdan, and Wolfgang Rosenstiel. Automatic cluster detection in Kohonen's SOM. *IEEE Transactions on Neural Networks*, 19(3):442–459, 2008.

[BG06] Stefan Bruckner and M Eduard Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 2006.

[BGKG05] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M Eduard Gröller. Illustrative context-preserving volume rendering. In *Proceedings of EuroVis*, pages 69–76, 2005.

[BGKG06] Stefan Bruckner, Soren Grimm, Armin Kanitsar, and M Eduard Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 2006.

[BI15] Ali Borji and Laurent Itti. CAT2000: A large scale fixation dataset for boosting saliency research. *arXiv preprint arXiv:1505.03581*, 2015.

[BJB+] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. MIT Saliency Benchmark. http://saliency.mit.edu/.

[BK02] Matjaz Bevk and Igor Kononenko. A statistical approach to texture description of medical images: A preliminary study. In *Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems*, pages 239–244. IEEE, 2002.

[BKKW08] Kai Burger, Polina Kondratieva, Jens Kruger, and Rüdiger Westermann. Importance-driven particle techniques for flow visualization. In *IEEE Pacific Visualization Symposium, 2008 (PacificVIS'08)*, pages 71–78. IEEE, 2008.

[BM10] Stefan Bruckner and Torsten Möller. Isosurface similarity maps. In *Computer Graphics Forum*, volume 29, pages 773–782. Wiley Online Library, 2010.

[BOBS89] Virginia E Barker, Dennis E O'Connor, Judith Bachant, and Elliot Soloway. Expert systems for configuration at digital: XCON and beyond. *Communications of the ACM*, 32(3):298–318, 1989.

[BPS97] C.L. Bajaj, V. Pascucci, and D.R. Schikore. The contour spectrum. In *Proceedings of Visualization '97.*, pages 167–173, 1997.

[BR01] Martin Bogdan and Wolfgang Rosenstiel. Detection of cluster in self-organizing maps for controlling a prostheses using nerve signals. In *Proceedings of European Symposium on Artificial Neural Networks*, pages 131–136, 2001.

[BS+84] Bruce G Buchanan, Edward Hance Shortliffe, et al. *Rule-based expert systems*, volume 3. Addison-Wesley Reading, MA, 1984.

[BS05] U.D. Bordoloi and H.W. Shen. View selection for volume rendering. In *IEEE Visualization (VIS '05)*, pages 487–494. IEEE, 2005.

[BSF68] Bruce Buchanan, Georgia Sutherland, and Edward A Feigenbaum. *Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry*. Defense Technical Information Center, 1968.

[BT07] Neil Bruce and John Tsotsos. Attention based on information maximization. *Journal of Vision*, 7(9):950–950, 2007.

[C.E48] C.E.Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[CH03] Chad Creighton and Samir Hanash. Mining gene expression databases for association rules. *Bioinformatics*, 19(1):79–86, 2003.

[CM08] Carlos D. Correa and Kwan-Liu Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, 2008.

[CM09a] Carlos D. Correa and Kwan-Liu Ma. The occlusion spectrum for volume visualization and classification. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1465–1472, 2009.

[CM09b] Carlos D. Correa and Kwan-Liu Ma. Visibility driven transfer functions. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 177–184, 2009.

[CM10] Carlos D. Correa and Kwan-Liu Ma. Visibility histograms and visibility-driven transfer functions. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):1077–2626, 2010.

[CMH+01] Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas Konig, and Eduard Gröller. Fast visualization of object contours by non-photorealistic volume

rendering. In *Computer Graphics Forum*, volume 20, pages 452–460. Citeseer, 2001.

[CR08] Jesus J Caban and Penny Rheingans. Texture-based transfer functions for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1364–1371, 2008.

[DGH79] Richard Duda, John Gaschnig, and Peter Hart. Model design in the Prospector consultant system for mineral exploration. *Expert Systems in the Microelectronic Age*, 1234:153–167, 1979.

[dMPF10] Francisco de Moura Pinto and Carla M. D. S. Freitas. Importance-aware composition for illustrative volume rendering. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 134–141. IEEE, 2010.

[DRDH12] C. De Runz, E. Desjardin, and M. Herbin. Unsupervised visual data mining using self-organizing maps and a data-driven color mapping. In *Proceedings of 16th International Conference on Information Visualisation (IV)*, pages 241–245, 2012.

[Duc07] Andrew Duchowski. *Eye tracking methodology: Theory and practice*, volume 373. Springer Science and Business Media, 2007.

[FAT99] I. Fujishiro, T. Azuma, and Y. Takeshima. Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In *Proceedings of Visualization '99.*, pages 467–563, 1999.

[Fur86] George W Furnas. *Generalized fisheye views*, volume 17. ACM, 1986.

[GH10] Joseph H Goldberg and Jonathan I Helfman. Comparing information graphics: A critical look at eye tracking. In *Proceedings of the 3rd BELIV'10 Workshop: Beyond time and errors: Novel evaluation methods for information visualization*, pages 71–78. ACM, 2010.

[GH11] Joseph Goldberg and Jonathan Helfman. Eye tracking for visualization evaluation: Reading values on linear versus radial graphs. *Information Visualization*, 10(3):182–195, 2011.

[GLY14] Hanqi Guo, Wei Li, and Xiaoru Yuan. Transfer function map. In *2014 IEEE Pacific Visualization Symposium (PacificVis)*, pages 262–266. IEEE, 2014.

[GMY11] Hanqi Guo, Ningyu Mao, and Xiaoru Yuan. WYSIWYG (what you see is what you get) volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2106–2114, 2011.

[GP11] Petr Gajdoš and Jan Platoš. GPU based parallelism for self-organizing map. In *Proceedings of the Third International Conference on Intelligent Human*

Computer Interaction (IHCI 2011), volume 179 of *Advances in Intelligent Systems and Computing*, pages 231–242. Springer Berlin Heidelberg, 2011.

[GRW+00] Donna L Gresh, Bernice E Rogowitz, Raimond L Winslow, David F Scollan, and Christina K Yung. WEAVE: A system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the IEEE Conference on Visualization '00*, pages 489–492. IEEE Computer Society Press, 2000.

[GXY11] Hanqi Guo, He Xiao, and Xiaoru Yuan. Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. In *Proceedings of Pacific Visualization Symposium (PacificVis)*, pages 19–26. IEEE, 2011.

[GY13] Hanqi Guo and Xiaoru Yuan. Local WYSIWYG volume visualization. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 65–72. IEEE, 2013.

[Hai07] Martin Haidacher. *Importance-driven rendering in interventional imaging.* 2007.

[Hau06] Helwig Hauser. Generalizing focus+context visualization. In *Scientific visualization: The visual extraction of knowledge from data*, pages 305–327. Springer, 2006.

[HBKG08] Martin Haidacher, Stefan Bruckner, Armin Kanitsar, and M Eduard Gröller. Information-based transfer functions for multimodal visualization. In *Proceedings of Eurographics Workshop on Visual Computing for Biomedicine*, pages 101–108. Eurographics Association, 2008.

[Hd14] Eric Heitz and Eugene d'Eon. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In *Computer Graphics Forum*, volume 33, pages 103–112. Wiley Online Library, 2014.

[HDKS05] Ming C Hao, Umeshwar Dayal, Daniel A Keim, and Tobias Schreck. Importance-driven visualization layouts for large time series data. In *IEEE Symposium on Information Visualization, 2005 (INFOVIS 2005)*, pages 203–210. IEEE, 2005.

[HF95] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the 21th International Conference on Very Large Data Bases (VLDB '95)*, volume 95, pages 420–431, 1995.

[HHK12] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):194–201, 2012.

[HHKP96] Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. Generation of transfer functions with stochastic search techniques. In *Proceedings of the 7th IEEE Conference on Visualization*, VIS '96, pages 227–234, 1996.

[HKG00] Jiří Hladuvka, Andreas König, and Eduard Gröller. Curvature-based transfer functions for direct volume rendering. In *Proceedings of Spring Conference on Computer Graphics*, volume 16, pages 58–65. Citeseer, 2000.

[HMBG01] Helwig Hauser, Lukas Mroz, Gian Italo Bischi, and M Eduard Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, 2001.

[HNA+11] Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. *Eye tracking: A comprehensive guide to methods and measures*. Oxford University Press, 2011.

[HPB+10] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M.E. Gröller. Volume visualization based on statistical transfer function spaces. In *Proceedings of IEEE Pacific Visualization Symposium (PacificVis)*, pages 17–24, 2010.

[HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM SIGMOD Record*, volume 29, pages 1–12. ACM, 2000.

[HYL+15] Hsin-Yang Ho, I Yeh, Yu-Chi Lai, Wen-Chieh Lin, Fu-Yin Cherng, et al. Evaluating 2D flow visualization using eye tracking. In *Computer Graphics Forum*, volume 34, pages 501–510. Wiley Online Library, 2015.

[HZ03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2003.

[IK00] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10):1489–1506, 2000.

[IK01] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (11):1254–1259, 1998.

[IVJ12] Cheuk Yiu Ip, Amitabh Varshney, and Joseph JaJa. Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms. *IEEE Transactions on Visualization and Computer Graphics*, 18:2355–2363, 2012.

[JC10] Heike Jänicke and Min Chen. A salience-based quality metric for visualization. In *Computer Graphics Forum*, volume 29, pages 1183–1192. Wiley Online Library, 2010.

[JDT12] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012.

[JEDT09] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *IEEE 12th International Conference on Computer Vision*, pages 2106–2113. IEEE, 2009.

[Jen95] Henrik Wann Jensen. Importance driven path tracing using the photon map. In *Rendering Techniques '95*, pages 326–335. Springer, 1995.

[JH09] C Ryan Johnson and Jian Huang. Distribution-driven visualization of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):734–746, 2009.

[JHDZ15] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. SALICON: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2015.

[KD98] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of IEEE Symposium on Volume Visualization*, pages 79–86, 1998.

[Kia01] Melody Y. Kiang. Extending the Kohonen self-organizing map networks for clustering analysis. *Computational Statistics and Data Analysis*, 38(2):161 – 180, 2001.

[Kin02] Gordon Kindlmann. Transfer functions in direct volume rendering: Design, interface, interaction. *Course notes of ACM SIGGRAPH*, 2002.

[KKG13] Naimul Mefraz Khan, Matthew Kyan, and Ling Guan. Intuitive volume exploration through spherical self-organizing map. In *Advances in Self-Organizing Maps*, pages 75–84. Springer, 2013.

[KKG15] Naimul Mefraz Khan, Matthew Kyan, and Ling Guan. Intuitive volume exploration through spherical self-organizing map and color harmonization. *Neurocomputing*, 147:160–173, 2015.

[KKH01] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the IEEE Conference on Visualization '01*, pages 255–262. IEEE Computer Society, 2001.

[KKH02] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[KSC+10] Han Suk Kim, Jürgen P Schulze, Angela C Cone, Gina E Sosinsky, and Maryann E Martone. Dimensionality reduction on multi-dimensional transfer functions for multi-channel volume data sets. *Information Visualization*, 9(3):167–180, 2010.

[KSW06] Jens Krüger, Jens Schneider, and Rüdiger Westermann. Clearview: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):941–948, 2006.

[KU87] Christof Koch and Shimon Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. In *Matters of Intelligence*, pages 115–141. Springer, 1987.

[KV06] Youngmin Kim and Amitabh Varshney. Saliency-guided enhancement for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):925–932, 2006.

[KVK00] Samuel Kaski, Jarkko Venna, and Teuvo Kohonen. Coloring that reveals cluster structures in multivariate data. *Australian Journal of Intelligent Information Processing Systems*, 6(2):82–88, 2000.

[LCC14] David P Lau, Chin Boon Chng, and Chee Kong Chui. New device for single-stage in-office secondary tracheoesophageal puncture: Animal studies. *Head and Neck*, 2014.

[Lin91] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37:145–151, 1991.

[LLL+10] Stefan Lindholm, Patric Ljung, Claes Lundström, Anders Persson, and Anders Ynnerman. Spatial conditioning of transfer functions using local material distributions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1301–1310, 2010.

[LLY06] Claes Lundström, Patric Ljung, and Anders Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1570–1579, 2006.

[LVLRR08] L. Linsen, T. Van Long, P. Rosenthal, and S. Rosswog. Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization.

IEEE Transactions on Visualization and Computer Graphics, 14(6):1483–1490, 2008.

[LW90] Marc Levoy and Ross Whitaker. Gaze-directed volume rendering. *ACM SIGGRAPH Computer Graphics*, 24(2):217–223, 1990.

[Ma99] Kwan-Liu Ma. Image graphs: A novel approach to visual data exploration. In *Proceedings of the IEEE Conference on Visualization '99: celebrating ten years*, pages 81–88. IEEE Computer Society Press, 1999.

[MAB+97] Joe Marks, Brad Andalman, Paul A Beardsley, William Freeman, Sarah Gibson, Jessica Hodgins, Thomas Kang, Brian Mirtich, Hanspeter Pfister, Wheeler Ruml, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 389–400. ACM Press/Addison-Wesley Publishing Co., 1997.

[MCWE09] Ross Maciejewski, Wei Chen, Insoo Woo, and David S. Ebert. Structuring feature space - A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480, 2009.

[MHB+00] Michael Meißner, Jian Huang, Dirk Bartz, Klaus Mueller, and Roger Crawfis. A practical evaluation of popular volume rendering algorithms. In *Proceedings of the 2000 IEEE Symposium on Volume Visualization*, pages 81–90. ACM, 2000.

[MJW+13] Ross Maciejewski, Yun Jang, Insoo Woo, Heike Janicke, Kelly P Gaither, and David S Ebert. Abstracting attribute space for transfer function exploration and design. *IEEE Transactions on Visualization and Computer Graphics*, 19(1):94–107, 2013.

[MK13] Victor Matvienko and Jorg Kruger. A metric for the evaluation of dense vector field visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1122–1132, 2013.

[MMKI14] Ann McNamara, Katerina Mania, George Koulieris, and Laurent Itti. Attention-aware rendering, mobile graphics and games. In *ACM SIGGRAPH 2014 Courses*. ACM, 2014.

[MU+05] Fabien Moutarde, Alfred Ultsch, et al. U∗ F clustering: A new performant "cluster-mining" method based on segmentation of self-organizing maps. In *Proceedings of Workshop on Self-Organizing Maps*, 2005.

[NBW14] Rudolf Netzel, Michel Burch, and Daniel Weiskopf. Comparative eye tracking study on node-link visualizations of trajectories. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2221–2230, 2014.

[NTCO12] B.P. Nguyen, W.-L. Tay, C.-K. Chui, and S.-H. Ong. A clustering-based system to automate transfer function design for medical image visualization. *The Visual Computer*, 28(2):181–191, 2012.

[NXC09] Eric WT Ngai, Li Xiu, and Dorothy CK Chau. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2):2592–2602, 2009.

[OAJ14] Mershack Okoe, Sayeed Safayet Alam, and Radu Jianu. A gaze-enabled graph visualization to improve graph reading tasks. In *Computer Graphics Forum*, volume 33, pages 251–260. Wiley Online Library, 2014.

[OM04] David Opolon and Fabien Moutarde. Fast semi-automatic segmentation algorithm for self-organizing maps. In *Proceedings of European Symposium on Artificial Neural Networks*, 2004.

[PF07] Francisco de Moura Pinto and Carla MDS Freitas. Design of multi-dimensional transfer functions using dimensional reduction. In *Proceedings of the 9th Joint Eurographics/IEEE VGTC Conference on Visualization*, pages 131–138. Eurographics Association, 2007.

[PH08] Dimitris Petrilis and Constantin Halatsis. Two-level clustering of web sites using self-organizing maps. *Neural Processing Letters*, 27(1):85–95, 2008.

[PLB+01] Hanspeter Pfister, Bill Lorensen, Chandrajit Bajaj, Gordon Kindlmann, Will Schroeder, Raghu Machiraju, and Jinho Lee. The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, 2001.

[PRMH10] J-S Praßni, Timo Ropinski, Jörg Mensmann, and Klaus Hinrichs. Shape-based transfer functions for volume visualization. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 9–16. IEEE, 2010.

[PWH01] V. Pekar, R. Wiemker, and D. Hempel. Fast detection of meaningful isosurfaces for volume data visualization. In *Proceedings of Visualization (VIS '01)*, pages 223–230, 2001.

[RBBV11] M. Ruiz, A. Bardera, I. Boada, and I. Viola. Automatic transfer functions based on informational divergence. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1932–1941, 2011.

[RBG07] Peter Rautek, Stefan Bruckner, and Eduard Gröller. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1336–1343, 2007.

[RBS05] Stefan Röettger, Michael Bauer, and Marc Stamminger. Spatialized transfer functions. In *Proceedings of EUROGRAPHICS-IEEE VGTC Symposium on Visualization*, pages 271–278, 2005.

[RPSH08] Timo Ropinski, J Praßni, Frank Steinicke, and Klaus Hinrichs. Stroke-based transfer function design. In *Volume Graphics*, pages 41–48, 2008.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *ACM SIGMOD Record*, volume 25, pages 1–12. ACM, 1996.

[SAS92] Brian E Smits, James R Arvo, and David H Salesin. An importance-driven radiosity algorithm. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 273–282. ACM, 1992.

[SB98] Lisa Streit and John Buchanan. Importance driven halftoning. In *Computer Graphics Forum*, volume 17, pages 207–217. Wiley Online Library, 1998.

[SCDT00] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12–23, 2000.

[SG00] Dario D Salvucci and Joseph H Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *Proceedings of the 2000 Symposium on Eye Tracking Research and Applications*, pages 71–78. ACM, 2000.

[SG09] M Alper Selver and Cüneyt Güzeliş. Semiautomatic transfer function initialization for abdominal visualization using self-generating hierarchical radial basis function networks. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):395–409, 2009.

[Sho76] Edward H Shortliffe. MYCIN: Computer-based medical consultations, 1976.

[SKK06] Christof Rezk Salama, Maik Keller, and Peter Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.

[SLC99] Mu-Chun Su, Ta-Kang Liu, and Hsiao-Te Chang. An efficient initialization scheme for the self-organizing feature map algorithm. In *Proceedings of International Joint Conference on Neural Networks*, volume 3, pages 1906–1910. IEEE, 1999.

[SM00] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:888–905, 2000.

[SM04] Jianbo Shi and Jitendra Malik. Normalized cut segmentation code. *Copyright 2004 University of Pennsylvania, Computer and Information Science Department*, 2004.

[SS15] Krishna Prasad Soundararajan and Thomas Schultz. Learning probabilistic transfer functions: A comparative study of classifiers. In *Computer Graphics Forum*, volume 34, pages 111–120. Wiley Online Library, 2015.

[SzBBKN14] Christian Schulte zu Berge, Maximilian Baust, Ajay Kapoor, and Nassir Navab. Predicate-based focus-and-context visualization for 3D ultrasound. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2379–2387, 2014.

[TAK+05] Melanie Tory, M Stella Atkins, Arthur E Kirkpatrick, Marios Nicolaou, and Guang-Zhong Yang. Eyegaze analysis of displays with combined 2D and 3D views. In *Proceedings of the IEEE Conference on Visualization '05*, pages 519–526. IEEE, 2005.

[TCSC13] Dereck Toker, Cristina Conati, Ben Steichen, and Giuseppe Carenini. Individual user characteristics and information visualization: Connecting the dots through eye tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 295–304. ACM, 2013.

[TG80] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[TLM03] Fan-Yin Tzeng, Eric B Lum, and Kwan-Liu Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of the 14th IEEE Visualization (VIS '03)*, page 66. IEEE Computer Society, 2003.

[TLM05] F-Y Tzeng, Eric B Lum, and K-L Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.

[TM04] Fan-Yin Tzeng and Kwan-Liu Ma. A cluster-space visual interface for arbitrary dimensional classification of volume data. In *Proceedings of IEEE/Eurographics Symposium on Visualization*, pages 17–24, 2004.

[TM09] Kadim Tasdemir and Erzsébet Merényi. Exploiting data topology in visualization and clustering of self-organizing maps. *IEEE Transactions on Neural Networks*, 20(4):549–562, 2009.

[TPD06] Andreas Tappenbeck, Bernhard Preim, and Volker Dicken. Distance-based transfer function design: Specification methods and applications. In *Simulation und Visualisierung*, pages 259–274. Citeseer, 2006.

[Ult93] Alfred Ultsch. Self-organizing neural networks for visualisation and classification. In *Information and Classification*, pages 307–313. Springer, 1993.

[Ult03a] Alfred Ultsch. Maps for the visualization of high-dimensional data spaces. In *Proceedings of Workshop on Self-Organizing Maps*, pages 225–230, 2003.

[Ult03b] Alfred Ultsch. *U∗-matrix: a tool to visualize clusters in high dimensional data.* Fachbereich Mathematik und Informatik Berlin, 2003.

[VA00] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, 2000.

[vBSG06] Petr Šereda, Anna Vilanova Bartroli, Iwo W. O. Serlie, and Frans A. Gerritsen. Visualization of boundaries in volumetric data sets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):208–218, 2006.

[VDC14] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2798–2805. IEEE, 2014.

[VFSG06] Ivan Viola, Miquel Feixas, Mateu Sbert, and Meister Eduard Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, 2006.

[VKG04] Ivan Viola, Armin Kanitsar, and Meister Eduard Gröller. Importance-driven volume rendering. In *Proceedings of the IEEE Conference on Visualization '04*, pages 139–146. IEEE Computer Society, 2004.

[VKG05] Ivan Viola, Armin Kanitsar, and M Eduard Gröller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.

[VSBO11] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. The city trip planner: An expert system for tourists. *Expert Systems with Applications*, 38(6):6540–6546, 2011.

[vVG06] Petr Šereda, Anna Vilanova, and Frans A. Gerritsen. Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In *Proceedings of IEEE/Eurographics Symposium on Visualization*, pages 243–250, 2006.

[WC04] Sitao Wu and Tommy WS Chow. Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern Recognition*, 37(2):175–188, 2004.

[WCS+10] Yunhai Wang, Wei Chen, Guihua Shan, Tingxin Dong, and Xuebin Chi. Volume exploration using ellipsoidal gaussian transfer functions. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 25–32. IEEE, 2010.

[WCZ+11] Yunhai Wang, Wei Chen, Jian Zhang, Tingxing Dong, Guihua Shan, and Xuebin Chi. Efficient volume exploration using the Gaussian mixture model. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1560 –1573, 2011.

[WDC+07] Gunther H Weber, Scott E Dillard, Hamish Carr, Valerio Pascucci, and Bernd Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):330–341, 2007.

[WEE03] Daniel Weiskopf, Klaus Engel, and Thomas Ertl. Interactive clipping techniques for texture-based volume visualization and volume shading. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):298–312, 2003.

[Win09] G Winter. Xia2: An expert system for macromolecular crystallography data reduction. *Journal of Applied Crystallography*, 43(1):186–190, 2009.

[WK09] S. Wesarg and M. Kirschner. 3D visualization of medical image data employing 2D histograms. In *Proceedings of Second International Conference in Visualisation (VIZ '09)*, pages 153 –158, 2009.

[WK12] Lei Wang and Arie Kaufman. Importance driven automatic color design for direct volume rendering. In *Computer Graphics Forum*, volume 31, pages 1305–1314. Wiley Online Library, 2012.

[WLMB+14] Manuela Waldner, Mathieu Le Muzic, Michael Bernhard, Werner Purgathofer, and Ivan Viola. Attractive Flicker - Guiding attention in dynamic narrative visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2456–2465, 2014.

[WQ07] Yingcai Wu and Huamin Qu. Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1027–1040, 2007.

[WS11] C. Wang and H.W. Shen. Information theory in scientific visualization. *Entropy*, 13(1):254–273, 2011.

[WYM08] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.

[WZC+11] Y. Wang, J. Zhang, W. Chen, H. Zhang, and X. Chi. Efficient opacity specification based on feature visibilities in direct volume rendering. *Computer Graphics Forum*, 30(7):2117 –2126, 2011.

[WZJ+12] Yunhai Wang, Jian Zhang, Dirk J.Lehmann, Holger Theisel, and Xuebin Chi. Automating transfer function design with valley cell-based clustering of 2D density plots. In *Proceedings of Eurographics Conference on Visualization*, pages 1295–1304, 2012.

[WZK12] Lei Wang, Xin Zhao, and Arie E Kaufman. Modified dendrogram of attribute space for multidimensional transfer function design. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):121–131, 2012.

[WZMK05] Lujin Wang, Ye Zhao, Klaus Mueller, and Arie Kaufman. The magic volume lens: An interactive focus+context technique for volume rendering. In *Proceedings of IEEE Visualization '05*, pages 367–374. IEEE, 2005.

[ZDT04] Jianlong Zhou, Andreas Döring, and Klaus D Tönnies. Distance based enhancement for focal region based volume rendering. In *Bildverarbeitung für die Medizin 2004*, pages 199–203. Springer, 2004.

[ZH13] Liang Zhou and Charles Hansen. Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 73–80. IEEE, 2013.

[ZK10] Xin Zhao and Arie Kaufman. Multi-dimensional reduction and transfer function design using parallel coordinates. In *Proceedings of the 8th IEEE/EG International Conference on Volume Graphics*, pages 69–76. Eurographics Association, 2010.

[ZS13] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *IEEE International Conference on Computer Vision (ICCV)*, pages 153–160. IEEE, 2013.

[ZTM+08] Lingyun Zhang, Matthew H Tong, Tim K Marks, Honghao Shan, and Garrison W Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):32, 2008.

List of Publications

1. **Lile Cai**, Binh P. Nguyen, Chee-Kong Chui, Sim-Heng Ong. A Two-Level Clustering Approach for Multidimensional Transfer Function Specification in Volume Visualization. In *The Visual Computer*, 2015. (doi:10.1007/s00371-015-1167-y)

2. **Lile Cai**, Binh P. Nguyen, Chee-Kong Chui, Sim-Heng Ong. Rule-Enhanced Transfer Function Generation for Medical Volume Visualization. In *Proceedings of the 17th Joint Eurographics/IEEE VGTC Conference on Visualization (EuroVis 2015).* (published in *Computer Graphics Forum*, Vol. 34, No. 3, pp. 121-130, 2015.)

3. Binh P. Nguyen, Rong Wen, **Lile Cai**, Jichuan Wu, Sim-Heng Ong, Chee-Leong Teo, Chee-Kong Chui. Accelerated Medical Computing Toolkit and GPU-Accelerated Importance-Driven Volume Visualization. In *GPU Technology Conference*, Silicon Valley, 17-20 March 2015.

4. Binh P. Nguyen, **Lile Cai**, Chee-Kong Chui, Sim-Heng Ong. GPU-Accelerated Transfer Function Generation for Importance-Driven Volume Visualization. In *GPU Technology Workshop South East Asia (GTW SEA 2014)*, Singapore, 10 July 2014. (Best Poster Award)

5. **Lile Cai**, Wei-Liang Tay, Binh P. Nguyen, Chee-Kong Chui, Sim-Heng Ong. Automatic Transfer Function Design for Medical Visualization Using Visibility Distributions and Projective Color Mapping. In *Computerized Medical Imaging and Graphics*, 37(7-8), Oct-Dec 2013, pp. 450-459.