Transfer Function Optimization for Volume Visualization Based on Visibility and Saliency

by

Shengzhou Luo

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Doctor of Philosophy

University of Dublin, Trinity College

September 2016

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Shengzhou Luo

September 15, 2016

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Shengzhou Luo

September 15, 2016

Acknowledgments

I wish to express my sincerest thanks and gratitude to my supervisor Dr John Dingliana, for all his help, encouragement and support. This work would not have been possible without him.

Thanks are also due to my friends and colleagues in the Graphics Vision and Visualisation group (GV2), past and present, for making my time there thoroughly enjoyable and memorable. I also wish to acknowledge the contribution of Tom Noonan. His collaboration on GPU-based visibility computation was a great help to my work.

Finally, I would like to especially thank my parents, my sister and Lu Han, to whom I am forever indebted for their support, encouragement and patience throughout this research.

Shengzhou Luo

University of Dublin, Trinity College
September 2016

Transfer Function Optimization for Volume Visualization Based on Visibility and Saliency

Publication No. ___________________

Shengzhou Luo, Ph.D.
University of Dublin, Trinity College, 2016

Supervisor: John Dingliana

Volume data is widely used in scientific and medical research, and volume visualization techniques aim to provide effective and flexible methods for analyzing and exploring complex structures in such data. However, obtaining clear visualization of features of interest in volume data is still a major challenge. One time-consuming and unitintuitive part of the process is the specification of an appropriate transfer function, which determines the classification and visibility of features. In practice, this is typically achieved using a trial-and-error approach: modifications are made to the transfer function and changes in the resulting visualization are carefully observed in order to inform further modifications to the transfer function.

This thesis proposes and investigates novel automated optimization techniques for transfer functions, in order to emphasize features of interest. These techniques exploit information content associated with volume data and objective measures based on visual saliency and visibility in volume visualization.

We describe a global optimization and two user-driven refinement methods for modulating transfer functions in order to assist the exploration of volume data. This optimization is dependent on the distribution of the scalar values of the volume data set and is designed to reduce general occlusion and improve the clarity of layers of structures in the resulting images.

In addition to view-independent information, we propose a novel view-dependent measure called visibility-weighted saliency in order to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest in a volume rendered images. This measure is based on a computational measure of perceptual importance of voxels and the visibility of features in volume rendered images.

Subsequently, we present an automated transfer function optimization method based on the visibility-weighted saliency metric. This method takes into account the perceptual importance of voxels and the visibility of features, and automatically adjusts the transfer function to match the target saliency levels specified by the user. In addition, a parallel line search strategy is presented to improve the performance of the optimization algorithm.

Finally, we describe a multivariate visualization approach which modulates focus, emphasizing important information, by adjusting saturation and brightness of voxels based on an importance measure derived from temporal and multivariate information.

Contents

List of Tables

List of Figures

Glossary

Entropy

Entropy (or Shannon entropy) is a basic concept in information theory to measure the uncertainty of a random variable.

Feature

A feature in a volume is a subset of voxels in the volume. In this thesis, features are specified by intensity ranges in transfer functions.

Multivariate volume data

Multivariate volume data are volume data that contain multiple variables in each voxel, e.g. a data set containing variables from multiple modalities such as CT, MRI, and PET.

Saliency

Saliency (or visual saliency), for images, is the distinct subjective perceptual quality which makes some regions in an image stand out from their neighbors and immediately grab the user's attention.

Saliency field

A saliency field for a volume data set represents the saliency (see above) of each voxel in the volume data set. This field itself could be represented in the form of a volume data set.

Saliency model

A saliency model (or saliency map) is a computational model that integrates the normalized information from feature filters for visual properties such as color, intensity and orientation into one global measure of visual saliency.

Saturation

Saturation is a concept in color theory, which refers to the perceived purity or vividness of a specific color.

Time-variant volume data

Time-variant volume is a form of data that typically consists of a series of time-steps (frames) of volume data representing how a system of model changes over time.

Transfer function

Transfer functions are mappings that assign visual properties to volume data. This is a recurrent component of most volume visualization techniques.

Visibility

The visibility of a voxel is the contribution of a voxel to the volume rendered image, which is determined by both the opacity of this voxel and the opacity of those voxels in front of the current voxel in the view direction.

Visibility field

A visibility field for a volume data set represents the visibility of each voxel in the volume data set. This field itself could be represented in the form of a volume data set.

Visibility histogram

Visibility histograms summarize the distribution of visibility of voxels from a given viewpoint. They are feedback mechanisms of volume visualization.

Volume data

Volume data are discretely sampled along 3D grids and contain scalar values that are usually acquired from medical imaging devices such as CT or MRI machines or computed from scientific simulations such as fluid simulations.

Voxel

Elements in volume data sets are called voxels, which are analogues to pixels in 2D images.

Chapter 1

Introduction

Volume visualization is an active branch of scientific visualization concerned with extracting meaningful information from volume data (3D discretely sampled data sets) using interactive graphics and imaging. The study of volume visualization involves volume data representation, modeling, manipulation and rendering [8] and it aims, in particular, to facilitate visual exploration of 3D structures allowing users to more deeply understand and analyze volume data sets. First introduced by Levoy [9] in 1988, volume visualization has been widely used in various sciences to create insightful visualizations from both simulated and measured data. However, recent advances in volume data acquisition and scientific simulations have led to dramatic increases in both size and complexity of data sets, which present new and ongoing challenges to be addressed [10].

The rendering of volume data requires every sample value (also called voxel, which is a volume element or volumetric pixel) to be mapped to visual properties (e.g. opacity and color). This mapping is done with a transfer function, which can be a simple ramp, a piecewise linear function or an arbitrary table. The design of an effective transfer function (see Section 2.2 for details) is essential for visualizing volume data.

With volume rendering, both the exterior and interior of a volume data set can be revealed semi-transparently by specifying appropriate transfer functions. However, because of 3D occlusion between structures and the indirect control over the final visualization, it is time-consuming and unintuitive for users to specify appropriate transfer functions. In practice, this is typically achieved using a trial-and-error approach: modifications are made to the transfer function and changes in the resulting visualization are carefully observed in order to inform further modifications to the transfer function [11]. The adjustments users make in transfer function specification are based on subjective perception of important features in a certain viewpoint.

1.1 Motivation

Objective measures such as voxel information [12], visibility histograms [13] [5], feature visibility [14], and visual saliency models [15] such as saliency maps [16] [17] for 2D images and saliency fields [18] for volumetric data, provide the basis for powerful feedback mechanisms in volume rendering. In current volume rendering systems, appropriate transfer functions are often obtained by trial-and-error [19]. It is desirable to take advantage of these objective measures in order to automate the specification of transfer functions for emphasizing features of interest in volume visualization.

The main goal of our research is to investigate the optimization of visualization parameters (in particular transfer functions) with information derived from volume data based on feedback mechanisms from the volume rendering process. We hypothesize that the importance of voxels (sample values in volume data) are associated with their information content. Therefore, the transfer functions of volume visualization can be optimized based on the information inherent within the data sets and user input which indicates the user's interest. Furthermore, we hypothesize that combining automated optimization techniques with feedback mechanisms such as visibility and visual saliency can provide a more intuitive means for obtaining clear visualization of features of interest in volume data.

1.2 Scope

The focus of this thesis is on methods for enhancing user understanding of features of interest in volume visualization by optimizing transfer functions based on information derived from volume data (e.g. entropy and saliency of voxels). In addition to the information inherent in the volume data, view-dependent information (e.g. visibility of voxels) obtained in the volume rendering process is also exploited in the optimization of transfer functions.

In this research, we focus on the visualization of volume data sets, particularly the scalar field data acquired from medical imaging (e.g. CT and MRI scans) and generated from flow simulations (e.g. computational fluid dynamics).

The features of interest in a volume data set are specified by initial user-defined transfer functions. Therefore, manual segmentation by domain experts or computationally expensive automatic segmentation techniques are not in the scope of this thesis. Moreover, this thesis focuses on direct volume rendering techniques. Indirect volume rendering techniques, which require the reconstruction of 3D surfaces, are not in scope of this thesis.

1.3 Contributions

We present a transfer function refinement approach, which exploits the entropy of voxels derived from volume to equalize the opacity transfer function, in order to reduce general occlusion and improve the clarity of layers of structures in the resulting images. Moreover, this approach assists the user in exploring and enhancing features of interest by interactively specifying different priority intensity ranges.

In addition to view-independent information (i.e. entropy of voxels), we propose visibility-weighted saliency for measuring the view-dependent saliency of features of interest for volume visualization. This metric aims to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. (The formal definition of a feature is provided in Section 4.3.1.)

Subsequently, we describe an automated transfer function optimization method based on the visibility-weighted saliency metric. This method takes into account the perceptual importance of voxels and the visibility of features, and automatically adjusts the transfer function to match the target saliency levels specified by the user. In addition, a parallel line search strategy is presented to improve the performance of the optimization algorithm.

Finally, we develop a novel visualization approach which modulates focus, emphasizing important information, by adjusting saturation and brightness of voxels based on an importance measure derived from temporal and multivariate information.

1.4 Summary of Chapters

The rest of this thesis is structured as follows:

Chapter 2 provides an overview of the background and related work in the field of volume visualization, with particular focus on the design and optimization of transfer functions.

Chapter 3 presents a novel approach for transfer function refinement, which is an optimization of transfer functions based on the distribution (i.e. the histogram) of the volume data. This optimization also allows the user to prioritize specific regions by generating weightings for transfer function components based on user-selected regions. The work described in this chapter has been published as a short paper in Eurographics 2014 [20] and as a full paper in Eurasia 2014 [21].

Chapter 4 describes visibility-weighted saliency as an important measure of visual saliency of features in volume rendered images, in order to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. Visibility-weighted saliency is based on a computational measure of perceptual importance of voxels and the visibility of features in volume rendered images. The visibility-weighted saliency metric has been published as a full paper in Computer Graphics & Visual Computing (CGVC) 2015 [22].

Chapter 5 provides a detailed description of an automated transfer function optimization approach based on the visibility-weighted saliency metric, which indicates the perceptual importance of voxels and the visibility of features in volume rendered images. The work described in this chapter has been presented as a poster at EG / VGTC Conference on Visualization (EuroVis) 2016 [23].

Chapter 6 outlines a novel visualization approach which modulates focus, emphasizing important information, by adjusting saturation and brightness of voxels based on an importance measure derived from temporal and multivariate information. The work described in this chapter has been presented as a poster at EG / VGTC Conference on Visualization (EuroVis) 2015 [24].

Chapter 7 summarizes our contributions and provides a discussion of possible avenues of future work.

Related Publications:

1. Shengzhou Luo and John Dingliana, “Visibility-Weighted Saliency for Volume Visualization”, Computer Graphics and Visual Computing (CGVC), London, UK, 2015.

2. Shengzhou Luo and John Dingliana, “Transfer Function Refinement for Exploring Volume Data”, Eurasia Graphics 2014: International Conference on Computer Graphics, Animation and Gaming Technologies, Ankara, Turkey, 2014, p. 17.

3. Shengzhou Luo and John Dingliana, “Information-Guided Transfer Function Refinement”, Eurographics (Short Papers), Strasbourg, France, 2014, pp. 61–64.

Related Posters:

1. Shengzhou Luo and John Dingliana, “Selective Saturation and Brightness for Visualizing Time-Varying Volume Data”, EG / VGTC Conference on Visualization (EuroVis) 2015 Posters, Cagliari, Italy, 25-29, May 2015.

2. Shengzhou Luo and John Dingliana, “Transfer Function Optimization Based on a Combined Model of Visibility and Saliency”, EG / VGTC Conference on Visualization (EuroVis) 2016 Posters, Groningen, Netherlands, 6-10 June 2016 (Best Poster Award).

Chapter 2

Related Work

In this chapter, we present a brief review of the literature related to the concepts that we discuss in this thesis.

2.1 Volume Rendering

Volume rendering is used to display a two-dimensional (2D) image of three-dimensional (3D) data set. It can be considered as a process of projecting a 3D volumetric data set to a 2D image [25]. The majority of volume data sets are discretely sampled along 3D grids and contain scalar values usually acquired from medical imaging devices such as CT or MRI machines or computed from scientific simulations such as fluid simulation. Volume data sets have the form of 3D arrays and elements in the data sets are called voxels, which are analogues to pixels in 2D images. Figure 2.1 illustrates how voxels (cubic elements) constitute a volume data set. An example of volume rendering is provided in Figure 2.4, which shows a sliced image and a volume rendered image of a head data set.

Traditionally, volume rendering techniques are categorized as either direct volume rendering or indirect volume rendering. Indirect volume rendering is actually surface rendering. It is done by extracting surfaces (polygon meshes) from volume data sets and rendering these surfaces to the screen. A typical method for extracting polygonal meshes from volume data is Marching Cubes [26].

In contrast to surface rendering, direct volume rendering displays images of a 3D

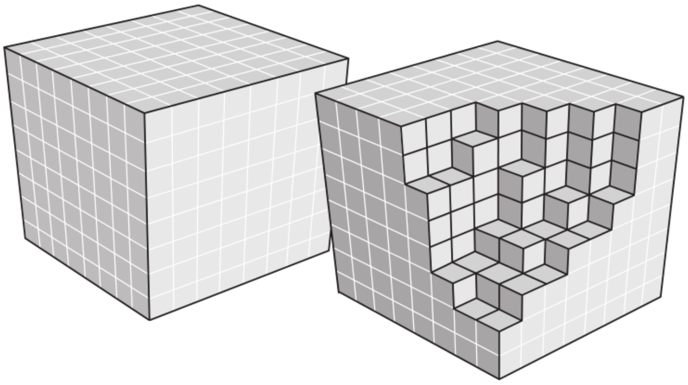


Figure 2.1: Voxels (cubic elements) constituting a volume data set [2].

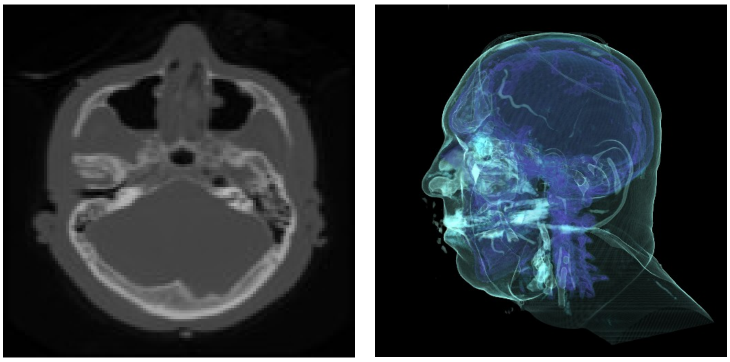


Figure 2.2: A sliced image of the data set

Figure 2.3: Volume rendering of the data set

Figure 2.4: The VisMale data set [3]

volume data set as a complete block of data without extracting geometric surfaces from the data [9]. Direct volume rendering consists of a variety of techniques for generating images from 3D scalar fields. Since indirect volume rendering is not in the scope of this thesis, direct volume rendering is henceforth referred to as volume rendering within this thesis.

Volume rendering was initially used in medical imaging, and later became an essential technique in many sciences for portraying complex phenomena such as clouds, water flows, and molecular and biological structure [27]. Volume visualization is a synonymous term for volume rendering, sometimes with emphasis on the effective expression of data rather than the realism of rendering.

Motivated by scientific visualization and medical imaging, where volume data is often acquired by devices such as CT and MRI scanners, or numerical simulation of natural phenomena, researchers have developed a wide variety of techniques to improve the performance and enhance the perception of volume visualization [28].

2.1.1 Volume Ray Casting

Volume ray casting is a volume rendering technique that can produce very high quality results. Since the advent of programmable graphics processing units (GPU) and 3D textures, volume ray casting techniques have been able to exploit the power of GPUs to achieve real-time performance. GPU-based volume ray casting has become one of the most popular techniques for visualizing volume data [29]. However, the sizes of volume data are increasing much faster than the sizes of memory available on GPUs [10]. Various scalable volume rendering methods [30] [31] have been proposed to address the challenge of dramatically increasing volume data sizes.

2.1.2 Splatting

Splatting is a volume rendering technique that trades quality for speed by combining volume projection with a sparse data representation [32]. In splatting, only voxels within certain intensity ranges need to be projected and the projection is done with efficient rasterization schemes [33].

Typical splatting approaches classify and shade voxels before the projection. Subsequently, each shaded voxel is projected to the screen using a 3D interpolation kernel. A

2D projection of the kernel is called a splat or footprint. Each voxel's splat is weighted by its color and opacity specified by the transfer function. Because of the pre-shaded scheme, projecting these fuzzy splats leads to a blurry appearance of object boundaries in the resulting image.

2.1.3 Shear Warp

The shear warp technique [34] consists of three steps. Firstly, a 3D shear is applied to the volume data in directions parallel to the data slices. Secondly, the sheared data is projected to a distorted intermediate image in 2D. Lastly, the intermediate image is warped to produce the final rendered volume image.

Shear warp is relatively fast in software implementation. However, compared to volume ray casting, shear warp has less accurate sampling and potentially worse image quality.

2.1.4 Texture-Based Volume Rendering

Texture-based volume rendering techniques render slices in the volume as texture-mapped polygons with the opacity and color of voxels determined by the transfer function and interpolation and then merge the polygons from back to front in compositing operations support by the hardware [35]. Because graphics cards are fast at texturing, texture-based volume rendering [36] can efficiently render slices of volume data with the texturing capability of graphics cards.

In 2D texture mapping approaches, slices are aligned with the original volume data. While in 3D texture mapping approaches, slices are aligned with the viewing plane and directly sampled from the 3D volume, which requires graphics hardware support for 3D textures [37].

Texture-based volume rendering techniques can produce images of reasonable quality. However, there may be noticeable transitions when the volume is rotated interactively [38]. Artifacts may also be noticeable at the edge of the image if the field-of-view angle is relatively large and the proxy geometry is based on planar slices [39].

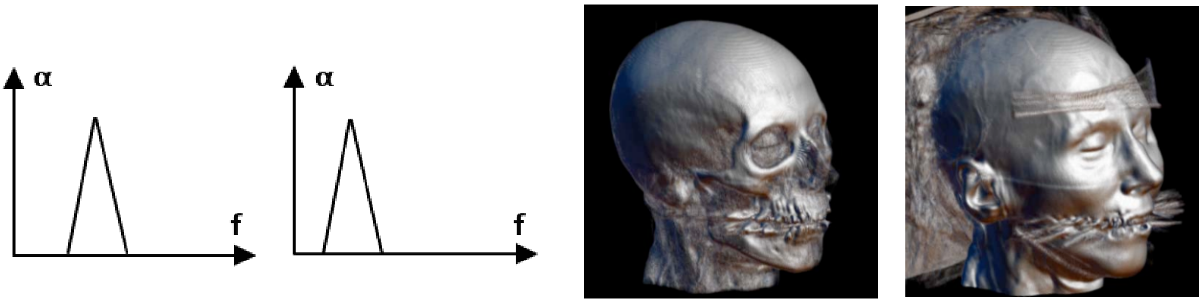
2.1.5 Illustrative Volume Visualization

Illustrative visualization, as a novel category of visualization, aims to visualize data in a clear and understandable way using techniques from traditional hand-crafted illustrations. Illustrative visualization has been successfully employed in medical visualization [40] [41] [42].

Illustration-based styles are believed to be effective in conveying information. Researchers in the field of computer graphics and visualization have applied illustration-based styles in order to produce effective and expressive visualization. Stompel et al. [43] introduced feature enhancement techniques, such as strokes-based, temporal domain enhancement, to enhance time-varying data obtained from the field of computational fluid dynamics (CFD).

In scientific visualization, features of interest may often comprise inner structures of the data sets, e.g. visualizing internal organs in anatomical data. In addition, depicting only the features of interest is not sufficient, because the user is often interested in exploring the features within the context of the whole volume data.

Inspired by techniques from illustration, various approaches have been proposed to reveal different levels of structures simultaneously in volume data sets. Two-level volume rendering [44] [45] [46], also related to focus and context visualization [47] [48] [49], is a method of merging two volume rendering techniques into a single rendering. A more detailed rendering method is used to emphasize the features of interest while the background information is kept in a de-emphasized form for contextual reference. This method can be useful when inner structures need to be rendered along with semitransparent outer parts. Cutaway techniques [50] [51], similar to those used by technical illustrators, involve selectively removing a spatial segment of the volume and can be used make inner structures clearly visible while preserving a sense of their spatial relationship with the surrounding material. In exploded views [52], the occluding objects are decomposed and displaced so that the internal details are visible. In contrast, structure extrusion techniques [53] make features of interest visible by extruding them off the clipping plane and preserving the context by adapting the extruded region.



(a) Two transfer functions (TF)

(b) The result from the TF on the left in (a)

(c) The result from the TF on the right in (a)

Figure 2.5: Slight changes in the transfer function causes significant difference in the resulting images [4]

2.2 Transfer Functions

Volume data are 3D entities with information inside them, but the data might not consist of surfaces and edges. Because of the lack of explicit geometric information, it is a major challenge to provide clear visualizations of the structures contained in a volume data set. Volume data may be rendered directly by mapping scalar values to visual properties (e.g. opacity and color), or an intermediate geometric representation may be extracted using techniques like Marching Cubes [26] and then rendered as geometric surfaces. The mapping, which assigns visual properties to volume data, is called a transfer function.

Transfer function specification is an essential part in volume visualization. A simple one-dimensional transfer function is a mapping from scalar values to RGB and alpha values. The resulting visualization largely depends on how well the transfer function captures features of interest [54]. However, it is non-trivial to obtain an effective transfer function. The specification is often achieved by a trial-and-error process, which involves a significant amount of tweaking of color and opacity. Figure 2.5 shows how slight changes in the transfer function lead to significant changes in the resulting images. The adjustment of transfer functions is unintuitive and often difficult.

In practice, major factors that have a great influence on transfer function setting are: partial volume effect [1], non-uniform distribution of materials and noise [56].

[1] During the acquisition of data, the finite resolution causes contributions of different materials combined into the value of a single voxel. This is generally referred to as the partial volume effect,

Among these, two challenging problems that need to be tackled could be elaborated as follows: firstly, for volume data sets, e.g. those obtained by MRI and CT, different tissues are represented in similar or even overlapping ranges of scalar values; secondly, interesting interior structures are often partly or completely occluded by surrounding tissue. Consequently, feature detection and understanding volume data become a big challenge.

These problems are handled by transfer functions, which have played a crucial role in volume visualization. Good transfer functions reveal important structures in the data without obscuring them with less important regions. The design of transfer functions to generate informative visualizations has been a significant challenge addressed by a number of researchers [19]. Various strategies have been proposed for transfer function design [39]. However, features with overlapping intensity intervals are difficult to extract and visualize with 1D transfer functions. When one intensity value or interval is associated with multiple boundaries, a 1D transfer function is unable to render them in isolation [54].

Classical approaches to this problem try to detect boundary information between tissues by introducing derived attributes such as first and second-order derivatives to isolate materials [57] [54]. In this case, the transfer functions are extended to multidimensional feature spaces. The introduction of multidimensional transfer functions alleviates the material separation problem. Instead of classifying a voxel based on a single scalar value, multidimensional transfer functions allow a voxel to be classified based on a combination of values. Multidimensional transfer functions are very effective means to extract materials and their boundaries for both scalar and multivariate data. Multidimensional transfer functions are discussed in more detail in the next subsection.

In addition, various user interfaces were proposed to simplify the design of multidimensional transfer functions [58] [59]. However, the parameter spaces of multidimensional transfer functions are more complex (compared to 1D transfer functions) and thus introduce problems such as requirement for large amount of user interaction, missing precision or the interaction being complex and unintuitive [60].

Another strategy is based on the selection of rendered images. This strategy lets the user select one or more favorite images to guide the further search of transfer

which results in blurred boundaries and hampers the detection of small or thin structures. [55]

functions [61] [62]. More recent approaches introduced visibility [5] [63] or measures derived from information theory [64] [65] [66] [67]. Zhou et al. studied the combination of 2D transfer functions with occlusion and size-based transfer functions [68].

Bruckner and Gröller introduced the concept of style transfer functions [69], which aim to produce more comprehensible images by using transfer functions that map input values to different non-photorealistic rendering styles.

Despite the advances of these methods, transfer function design for volume rendering is still an open research problem. The creation of transfer functions needs to be simplified and the functionality of transfer functions needs to be extended in order to realize the full potential of volume rendering. For instance, more sophisticated transfer functions are required in medical imaging, in order to address various domain specific visualization problems [70].

Moreover, transfer function specification in general is an unintuitive or even monotonous task for average users, because it usually involves an iterative process of trial and error. For instance, there are skin and fat tissues around the brain, and their intensities lie in the same range as the brain. If we want to visualize the brain by setting the scalar value range of the brain to opaque, the surrounding skin and fat tissue will also become opaque. Then the brain will be occluded by the surrounding soft tissues which make it difficult to explore the brain structure. Common approaches to this problem are to introduce explicit segmentation of structures of interest before the volume rendering process [71]. In fact, the process of applying the transfer function could be interpreted as a segmentation problem.

Multidimensional Transfer Functions

Multidimensional transfer functions [11], which are mappings from intensity and other variables, such as first and second derivatives to color and opacity, have demonstrated their effectiveness in distinguishing boundaries between materials in volume data.

In volume data, boundaries are regions between areas of relatively homogeneous material. It is difficult to detect boundaries because different materials often consist of overlapping intensity intervals. To address this problem, multidimensional transfer functions used derived attributes such as gradient magnitudes and second derivatives along with scalar values, in order to detect transitions between relatively homogeneous areas [57] [4]. In this case, the transfer functions are extended to multidimensional feature spaces. For higher-dimensional transfer functions, the generation of transfer functions could be memory intensive and costly to compute, and the interaction of transfer functions are more complex and unintuitive as the dimensionality becomes higher.

Therefore, two-dimensional (2D) histograms are often used in multidimensional transfer functions [72]. An example is a 2D histogram with axes representing a subset of the feature space (e.g. scalar value vs gradient magnitude), with each entry in the 2D histogram being the number of voxels for a given feature space pair. Even in the case of two-dimensional transfer functions, a considerable amount of user interaction is required in order to come up with meaningful results [60].

As one of the most common representations of voxel distributions, histograms are used in transfer function design to assign visual properties to voxels [19]. Bajaj et al. [73] introduced the contour spectrum to determine voxels corresponding to important isosurfaces in the volume. To overcome the difficulty of using one-dimensional transfer functions (solely based on scalar values stored in the voxels) to extract inner structures of interest from the volume data, Levoy [9] proposed the use of gradient magnitude to emphasize strong boundaries between different tissues.

The introduction of gradient magnitude as a data metric aims to detect voxels that are of large deviation compared with other voxels by approximating gradient magnitude at each sample point in the volume, because the exact distribution of data is unknown due to information lost in the discrete sampling process. Kindlmann and Durkin [57] extended Levoy's work [9] by introducing a higher dimensional transfer function domain based on gradient magnitudes and second derivatives. To emphasize different structures, Kniss et al. [11] presented a technique for interactively manipulating 2D histograms of gradient magnitudes and data values. In their work, material boundaries appear as arcs in the 2D histogram and can be selected with interactive widgets [54]. Kniss et al. [74] presented Gaussian transfer functions, which are suitable for the classification of narrow features in multidimensional domains.

Kindlmann et al. [75] proposed curvature-based transfer function to enhance the expressive and informative power of volume rendering. In their approach, volume data is rendered with contours to exhibit constant thickness in image space.

Šereda et al. [76] proposed LH histograms for improving the identification and selection of boundaries in 2D intensity-gradient transfer functions. Subsequently, Šereda et al. [77] presented a clustering method based on the LH histograms for semi-automatic transfer function design.

Haidacher et al. [78] described the statistical transfer function space, which is based on statistical properties such as mean and standard deviation of the data values (e.g. intensity and gradient magnitude for 2D transfer functions) in the neighborhood of each voxel. This approach can reduce the influence of noise and enhance visual appearance in volume rendering.

Wang et al. [79] described a clustering approach on 2D density plots for automatic transfer function design. Their approach allows the user to interactively explore the pre-computed clusters in the feature space and merge or remove uninterested features to improve visualization quality. Ip et al. [80] described a multilevel segmentation technique that mimics user exploration behaviors by recursively segmenting intensity-gradient histograms.

There are other multidimensional transfer function approaches, such as spatialized gradient-based transfer functions [81], distance-based transfer functions [82], size-based transfer function [83], texture-based transfer functions [84] [85].

In addition, parallel coordinates and dimensionality reduction algorithms (e.g. principal component analysis) have been employed to support the design of transfer functions in multidimensional parameter spaces [86] [87] [88].

2.3 Automated Transfer Function Generation

Researches have proposed various approaches to automate the design of transfer functions and provide acceptable suggestions which can be further edited by users. However, the usefulness of a transfer function mostly depends on the underlying question the user wants to answer. Moreover, users' tasks vary drastically from one domain to another. Therefore, most techniques work semi-automatically and very few techniques consider domain knowledge in the design process [89].

He et al. [90] addressed transfer function exploration as a parameter optimization problem and presented an approach to assist the user in exploring appropriate transfer functions using stochastic search techniques starting from an initial population. Another strategy is further tuning transfer functions based on user selection of favorable rendered images as feedback, in order to achieve desired results. Marks et al. [61] presented Design Gallery, which lets the user select one or more favorite images to guide the further search of transfer functions. Rezk-Salama et al. [91] presented high-level semantics to abstract parametric models of transfer functions in order to automatically assign transfer function templates.

Wu and Qu [62] developed a method that uses editing operations and stochastic search of the transfer function parameters to maximize the similarity between volume-rendered images given by the user. Maciejewski et al. [72] described a method to structure attribute space in order to guide users to regions of interest within the transfer function histogram. Chan et al. [92] developed a system to optimize transparency automatically in volume rendering based on Metelli's episcotister model to improve the perceptual quality of transparent structures. Correa and Ma [5] proposed the visibility histogram to guide the transfer function design.

Zhou and Takatsuka [93] presented an automated approach for generating transfer functions, which can depict inclusion relationships between structures in the volume, and maximize opacity and color differences among the structures. This approach uses a residue flow model based on Darcy's Law to differentiate the distribution of opacity between branches of a contour tree. Selver and Güzeliş [94] introduced a semi-automatic method for transfer function initialization and optimization using volume histogram stacks and radial basis function networks.

Inspired by how physicians interact with volume data to extract clinically relevant information, Läthén et al. [95] proposed an optimization method for shifting transfer function presets, in order to better visualize contrast enhanced blood vessels.

Maciejewski et al. proposed a non-parametric method to generate transfer functions [72]. In their later work [96], instead of using the attributes, metrics representing relationships and correlations in the underlying data were used in the method.

2.4 Visibility Histograms and Visibility-Driven Transfer Functions

Visibility has been studied in measuring viewpoint quality [12] and enhancing ghost and cutaway views [97] in volume visualization.

In traditional transfer function design, the visibility of structures revealed in volume rendering is a consequence of adjusting transfer function parameters, rather than a design parameter [98]. Correa and Ma [5] introduced visibility histograms to guide transfer function design for both manual and automatic adjustment. Visibility histograms (Figure 2.6), which summarize the distribution of visibility of voxels from a given viewpoint, are powerful feedback mechanisms of volume visualization [13]. Visibility histograms encode the information required to measure the efficacy of transfer functions and are advantageous in guiding and automating the manipulation of transfer functions.

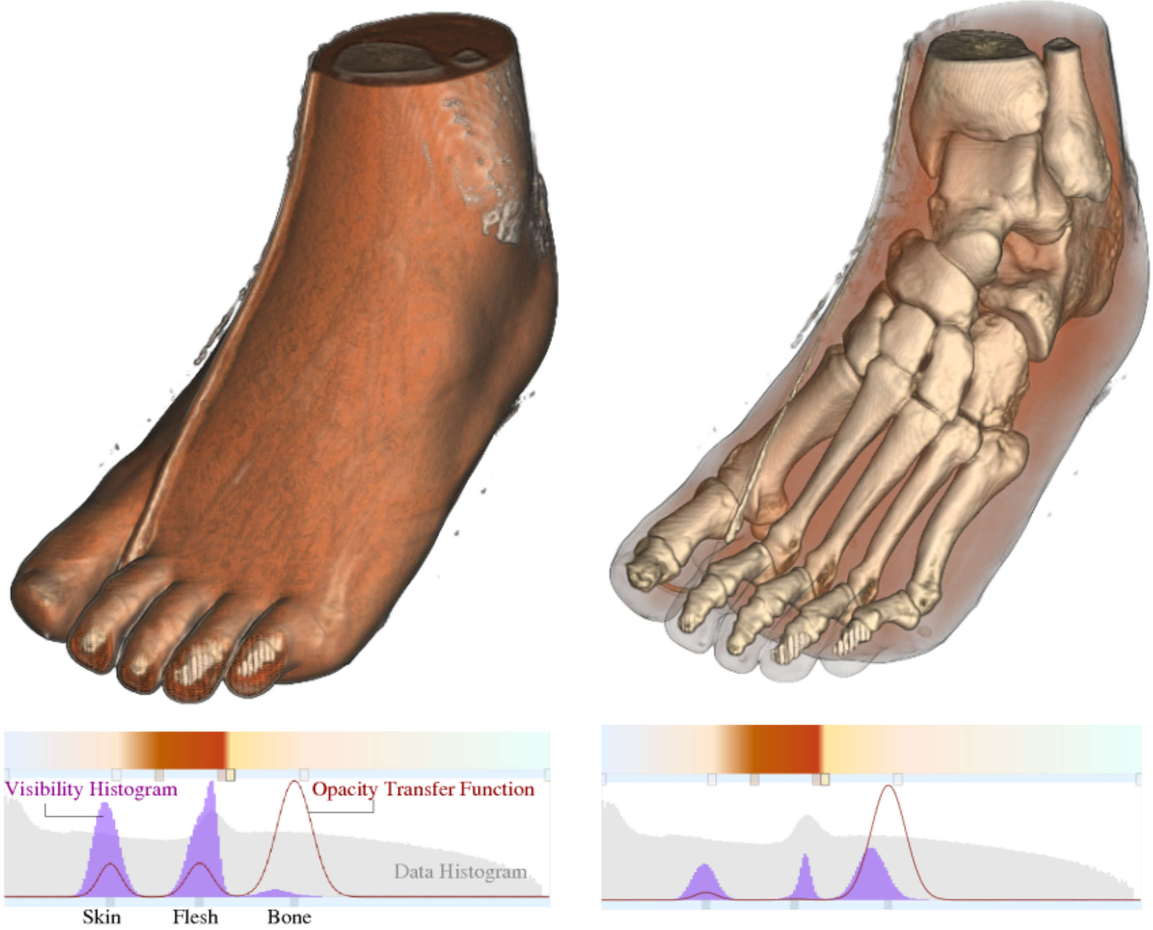
Wang et al. [6] extended the previous work on visibility histograms and proposed a feature visibility metric, in order to measure the influence of each feature to the volume rendered image. As shown in Figure 2.7, their approach allows the user to directly specify the desired visibility for the features of interest, and subsequently the opacity transfer function is optimized using an active set algorithm [99].

Ruiz et al. [66] proposed an information-theoretic framework which obtains opacity transfer functions by minimizing the Kullback-Leibler divergence between the observed visibility distribution and a target distribution provided by the user. Later, Bramon et al. [67] extended this approach to visualize multimodal volume data.

Cai et al. [100] described a method to derive opacity transfer functions by minimizing the Jensen-Shannon divergence between the observed visibility distribution and a user-defined target distribution. The target distribution can be defined using Gaussian function weighting.

In addition, various methods were proposed regarding the use of visibility for enhancing different aspects of volume visualization. Marchesin et al. [101] introduced a volume rendering technique that manipulates the voxel opacity values in a view-dependent way, in order to enhance visibility of internal structures in the volume data set. Bronstad et al. [102] described local opacity transfer functions with feature detection along the ray profile implemented on the GPU. In their approach, visibility histograms are employed to access the performance of the feature detection algorithm.

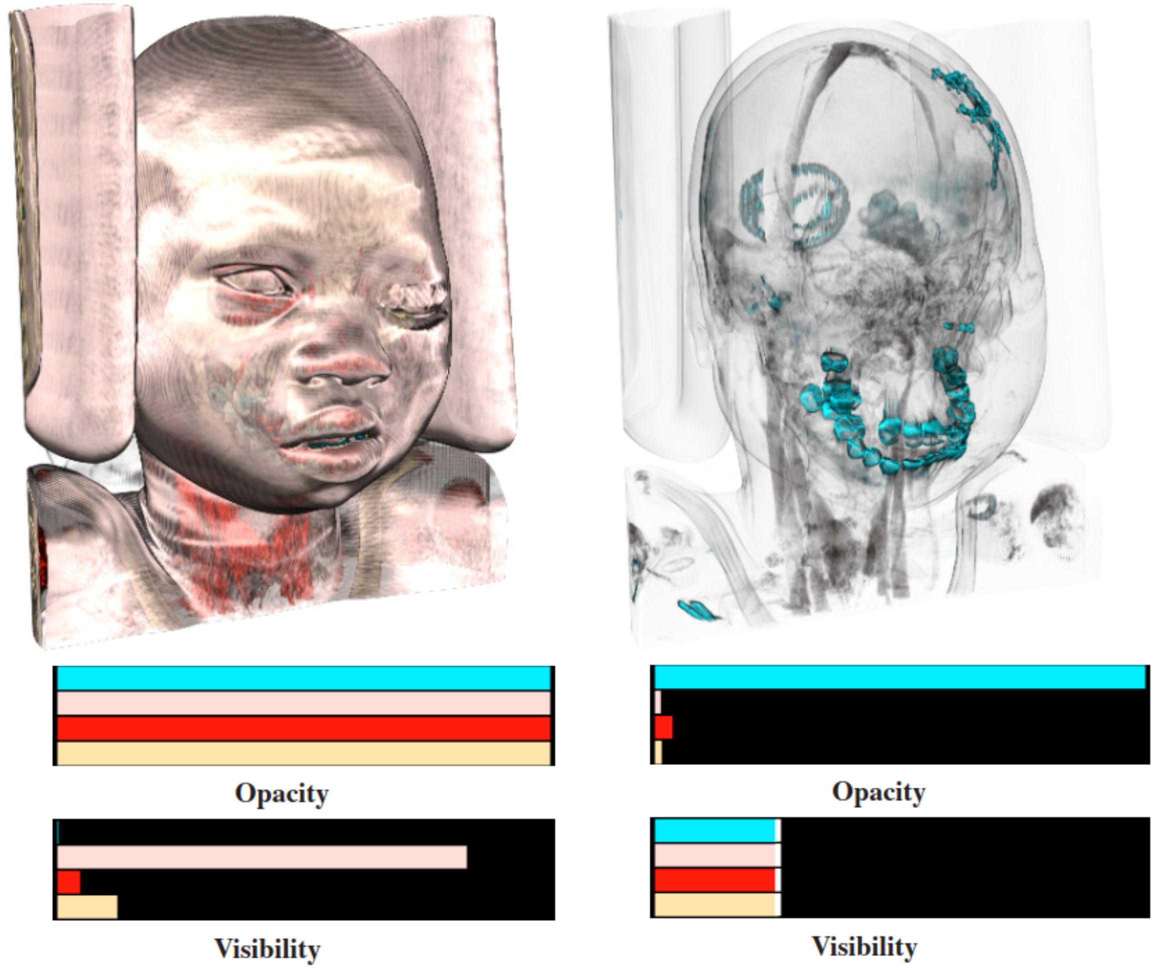
Jung et al. [103] presented a dual-modal visualization method, which uses visibility metrics to provide visual feedback regarding the occlusion caused by the volume data in one modal on the other modal. Jung et al. [104] extended visibility histograms to multimodal volume visualization. They demonstrated the use of visibility histograms



(a) A user-defined opacity transfer function and the initial visibility histogram

(b) Here the visibility histogram has been modified to match the user-defined opacity transfer function.

Figure 2.6: Visibility histograms [5]



(a) Feature opacities are equal

(b) Feature visibilities are equal

Figure 2.7: Opacities and feature visibilities of 4 features highlighted in different colors [6]

together with region of interest segmentation was effective in visualizing PET-CT volume data sets.

Instead of computing the visibility of all voxels, Zheng et al. [105] employed local visibility histograms to ensure both the features of interest and contextual information are visible in multimodal volume visualization. Schlegel and Pajarola [106] proposed a visibility-difference entropy metric. They presented an automated approach using this metric for generating a set of transfer function candidates with high ratings and are strongly distinct in what they reveal.

Qin et al. [107] presented the voxel visibility model as a quality metric for transfer function design. The voxel visibility model is a mapping function from data attributes of voxels to their visibility attributes. Instead of specifying transfer functions, this approach allows users to directly adjust the visibility of each voxel, and then the corresponding opacity transfer functions can be obtained by minimizing the distance between the desired voxel visibility distriubtion and the actual voxel visibility distribution.

2.5 Multivariate Volume Visualization

Analyzing multivariate data is an importance and challenging topic in many scientific disciplines. For instance, applications in medicine, engineering and meteorology often require analyzing multivariate data. However, multivariate volume data sets are usually mapped to a scalar dimension and visualized separately with standard volume rendering techniques. Because of the disparity in data resolutions and higher computation and storage requirements, simultaneous visualization of multiple variables in volume data is more challenging than standard volume visualization [108].

Stompel et al. [43] demonstrated feature enhancements using non-photorealistic rendering and hardware-accelerated techniques to generate perceptually effective visualization of multidimensional and multivariate volume data sets. Kniss and Hansen [109] applied volume rendering with multidimensional transfer function to visualize multivariate weather simulations. In their approach, they combined the temperature and humidity as a multivariate field in order to assist the meteorologists in identifying the frontal zones.

Akiba et al. [110] presented the use of time histograms for simultaneous classification of time-varying data in order to find transfer functions that classify all the time steps of the data set. Woodring and Shen [111] presented a method for the comparison of different data fields through the expression of a volume shader that composes data fields together with set operations. Wang et al. [112] introduced an importance measure based on conditional entropy and categorize temporal behaviors by clustering the importance curves over time.

Lee and Shen [113] introduced dynamic time warping (DTW) to measure the shape similarity between two time series with an optimal warping of time in order to account for the phase shift of the feature in time. Subsequently, they extended DTW to SUBDTW [114], in order to estimate when a trend appears and vanishes in a given time series. They modeled the temporal relationships as a state machine based on the beginning and ending times of the trends.

Khlebnikov et al. [115] described a novel method that allows simultaneous rendering of multivariate data by redistributing the opacity within a voxel. This method uses procedural texture synthesis [116] for opacity redistribution pattern and is similar in spirit to color weaving.

Data analysis techniques for high dimensional spaces, such as parallel coordinates [117] [118] and principal component analysis [119], were also investigated for exploring multivariate time-varying data sets.

2.6 Time-Varying Volume Visualization

Although researchers have developed a great number of visualization techniques for static volume data [19], how to effectively explore and understand time-varying volume data remains a challenging problem. Finding good transfer functions for time-varying volume data is more difficult than for static volume data, as data value ranges and distributions change over time.

2.6.1 Transfer Functions for Time-Varying Volume Visualization

Coherence is an important issue in transfer function design for time-varying volume data. Ideally, a single transfer function should be used for the whole time-varying data set in order to obtain coherent visualization. More than one color or opacity map can be misleading or physically meaningless, because the transition from one transfer function to another may cause sudden changes in the resulting images. However, the practice of using a single transfer function is not always applicable to general time-varying data sets. In some cases, the intensity distributions change significantly over time, thus applying a single transfer function to all frames becomes ineffective.

Volume data sets are inherently 3D representations. Automated analysis methods, such as temporal trends or statistical aggregates e.g. mean values and standard deviations, are often applied in order to abstract dynamic characteristics of the data sets [120]. Jankun-Kelly and Ma [121] examined how to combine transfer functions for different time-steps to generate a coherent transfer function. Woodring et al. [122] considered time-varying volume data as four-dimensional data field and provided a user interface to specify hyperplanes in 4D. Woodring and Shen [123] introduced an alternative approach to render multiple time-steps in a sequence with different colors into a single image. This approach provides the context of surrounding time steps but coherence of color among time-steps is hard to maintain.

Tikhonova et al. [124] presented an exploratory approach based on a compact representation of each time step of the data set in the form of ray attenuation functions. Ray attenuation functions are subsequently used for transfer function generation. Akiba et al. [110] introduced the time histogram which allows simultaneous classification and specification of temporal transfer functions for the entire time series.

A time-varying volume data set can be considered as a 3D array where each voxel contains a time-activity curve (TAC). Fang et al. [125] described an approach for classifying time-varying volume data based on the temporal behavior of voxels and three different similarity measures that can be used in their approach. Woodring and Shen [126] presented a method that filters time-varying volume data into several time scales using a wavelet transform and classifies the voxels by clustering the entire time series by time scale. Lee and Shen [113] proposed a method for classifying time-varying features using time activity curves with the dynamic time warping distance metric.

A single static transfer function may be able to capture dynamic features whose intensities change over time. To address this problem, Woodring et al. [127] utilized a method called temporal clustering and sequencing to find dynamic features and create dynamic transfer functions through time-series analysis (Figure 2.8).

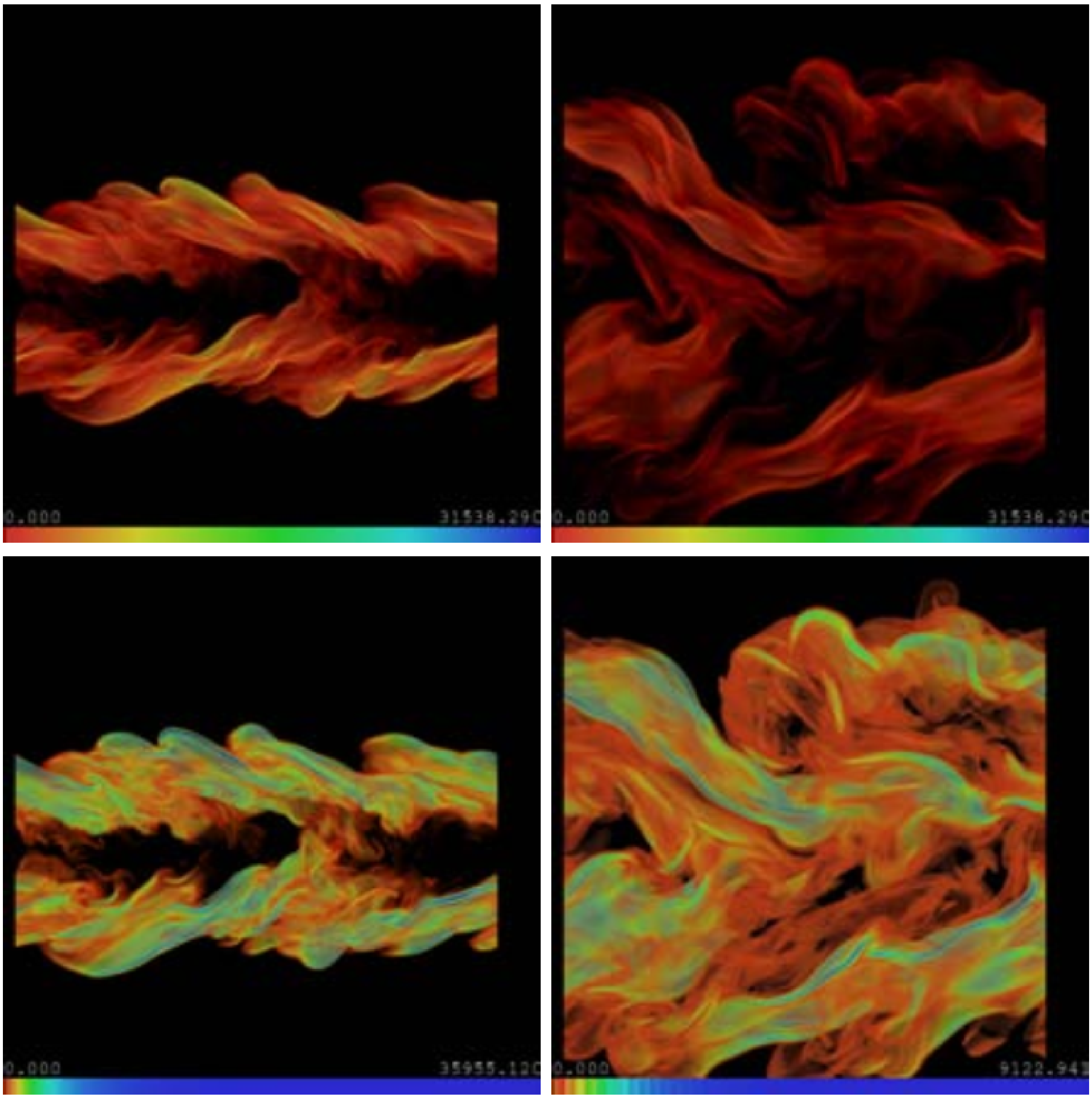


Figure 2.8: A single static transfer function cannot capture dynamic features. In the two images at the top, the features appear to vanish over time. On the other hand, the features are visible over time if a dynamic transfer function is used (the two images at the bottom) [127].

Ward and Guo [128] presented a method for visualizing time-series data that reveals a wide variety of features in the data, by mapping short sub-sequences of the time-varying volume data into a high-dimensional shape space, and then performing a dimension reduction process to allow projection into screen space.

Gu and Wang [129] proposed an approach to organize a time-varying data set into a hierarchical graph, which captures the transition relationships in the data set. This approach assists the user in comprehending the correspondence between volume regions over time and allows interaction of the graph through brushing and liking.

In order to create coherent and feature-prominent animations of time-varying volume data, Peng et al. [130] described an optimal color mapping strategy, which uses a two-phase optimization method with bilateral filtering and energy minimization.

2.6.2 Visualizing Time-Varying Volume Data with Non-Photorealistic Rendering

In contrast to traditional computer graphics, which has focused largely on creating photorealistic images of synthetic objects, non-photorealistic rendering is an area of computer graphics that focuses on creating abstract images with a wide variety of expressive styles [131]. In certain situations, non-photorealistic renderings are considered more effective and expressive than an equivalent photograph [132].

An essential problem in time-varying volume visualization is to visualize temporal variation and analysis of features. Traditionally, time-varying data has been visualized as snapshots of individual time steps or as an animation of snapshots of a sequence of time steps. These techniques are effective in making time-varying data understandable, however, they struggle when the complexity of data sets increased dramatically in recent years [133].

Compared to flow visualization, which is a well established branch of scientific visualization [133], general time-varying volume visualization is still a relative young field. Illustrations for time-varying rendering could be divided into two categories, one is to enhance time-invariant features and the other is to enhance temporal features of time-varying volume data. The techniques in the first category focus on enhancing structural perception of volume models through the amplification of features and the addition of illumination effects [134] [135]. Examples of these techniques include boundary enhancement, oriented feature enhancement (silhouettes, fading, and sketch lines). The techniques in the second category focus on illustrating dynamic aspects such as movement of features. A number of techniques have been proposed for this purpose. For example, there are speed lines, flow ribbons and strobe silhouettes, which are inspired by traditional animation [135] [136] [137]; and there are extended silhouette and boundary enhancement domains, which are inspired by the techniques used by illustrators and other artists [40]. Nevertheless, illustrations of temporal features of time-varying data requires more attention from researchers in the visualization community. The usefulness of illustrative approaches in time-varying volume visualization has not been studied as thoroughly as in other areas.

2.6.3 Vector Field Visualization

The visualization of vector fields plays a crucial role in visual interpretation and understanding of the underlying flow features and patterns [138] [139]. Since flow patterns also exist in time-varying volume data, certain techniques for visualizing vector fields could be incorporated into time-varying volume visualization, in order to depict the dynamic aspects of time-varying data.

Line drawings are effective ways to depict complex information with simple means [140]. Among vector field visualization techniques, streamline visualization is a simple but common way to convey the structure of 3D vector fields [141]. Streamlines have proven to be expressive in vector field visualization if they are combined with appropriate seeding strategies [142]. Texture patterns generated by line integral convolution are used in 2D vector field visualization. Rezk-Salama et al. [143] introduced volume rendering as a means to efficiently render 3D line integral convolution for visualizing 3D vector fields.

2.6.4 Feature Tracking

Feature extraction and tracking is an established technique for the analysis of time-varying data in various research fields, such as video analysis, computer vision and flow visualization [144]. In time-varying data, features are objects that evolve over time. Feature tracking aims to determine the correspondence between features in successive time steps and describe the evolution of features through time [145].

In practice, feature extraction and tracking are often employed in the exploration and analysis of time-varying volume visualization in order to better understand the dynamic nature of the underlying phenomena [146] [126] [113]. Feature extraction methods are often based on an analytic description of the feature of interest. Consequently, feature extraction and tracking could become manual-driven and trial-and-error processes when properties cannot be easily defined or unknown [147].

Tzeng and Ma [148] proposed an intelligent feature extraction and tracking algorithm, which applies machine learning to extract and track time-varying fluid flow features in volume data. They investigated the use of both neural networks and support vector machines in their approach and obtained promising results.

Based on the observation that regions in time-varying volume data present specific pattern and textural features, Caben et al. [149] introduced a texture-based feature tracking technique which analyzes local textural properties and finds correspondent features in the time series. Caban and Rheingans [84] extended this technique for transfer function design, in which the voxels' resulting opacity and color are based on local textural properties instead of intensity values.

Gu and Wang [150] described the TransGraph, which is a graph-based representation for visualizing hierarchical state transition relationships. This approach provides an occlusion-free overview map and allows controllable interaction to help users to track data transition over space and time.

Hsieh et al. [151] introduced a particle swarm method for transfer function design. This method maps the frequency of particle occurrences to color and opacity values, and extracts features after the particles finish searching for features in the volume data.

Johnson and Huang [152] introduced the use of local frequency distribution of intensity values in broader neighborhoods around each voxel to define features and found frequency distributions to contain meaningful information relevant to various feature queries. Cai et al. [153] presented a rule-enhanced transfer function generation method to separate and highlight important structures in volume visualization. They defined a set of rules based on the frequency distribution of data attributes and employed a genetic algorithm to select a set of rules that are most effective in distinguishing the target tissue from other tissues.

2.7 Information Theory in Visualization

Information theory [154] was originally introduced to study the fundamental limit of reliable transmission of messages through a noisy communication channel. In information theory, information is considered as a set of possible messages, where the goal is to send the messages through a noisy communication channel and then to have the messages reconstructed with low probability of error.

Traditional applications of information theory, such as data compression and data communication, focus on the efficient throughput of a communication channel, while visualization focuses on the effectiveness in aiding the perceptual and cognitive process for data understanding and knowledge discovery [155].

In recent years, there is an emerging direction towards using the principles of information theory to solve challenging problems in scientific visualization [14]. These problems include view selection [12] [156] [157], streamline seeding and selection [158] [159], transfer function for multimodal data [160], representative isosurface selection [161], time-varying and multivariate data analysis [112] and information channel between objects and viewpoints [162].

Chen and Jänicke [163] presented an information-theoretic framework for visualization. Examining the theoretical aspect of information and its relation to data communication, they interpreted different stages of the visualization pipeline using the taxonomy of information.

Haidacher et al. [64] proposed an approach for transfer function specification for multimodal data visualization. They considered the joint occurrence of multiple features from one or multiple variables in order to separate statistical features that only occur in a single variable from those that are present in both.

Wang et al. [112] introduced an approach to characterize the dynamic temporal behaviors of spatial blocks using importance curves, which are based on conditional entropy. Clustering is performed on the importance curves of all the spatial blocks to classify the underlying volume data set.

Bruckner and Möller [65] presented isosurface similarity maps as a similarity measurement of two isofurfaces based on the mutual information of their respective distance transforms. Furthermore, they developed an automatic method for identifying representative isovalues using isosurface similarity maps.

Ruiz et al. [66] presented an approach to generate transfer functions from a target distribution provided by the user. Their approach is based on a communication channel between a set of viewpoints and a set of bins of a volume data set, and supports both 1D and 2D transfer functions including the gradient information.

Bramon et al. [67] proposed an automatic method to visualize multimodal data by combining several information-theoretic strategies to define colors and opacity values of the multimodal transfer function. They set an information channel between two registered input data sets to define the fused color and minimize the informational divergence between the visibility distribution captured by a set of viewpoints and a target distribution proposed by the user to obtain the opacity.

2.8 Computational Saliency in Visualization

Predicting salient regions in images [164] can be exploited as useful feedback for improving visualizations. Inspired by mechanisms of the human visual system, various computational models of visual saliency have been proposed to predict gaze allocation in an image [16] [165] [17] [166] [167] [168].

Jänicke and Chen [169] described a quality metric for analyzing the saliency of visualization images and demonstrated its usefulness with examples from information visualization, volume visualization and flow visualization. Corcoran et al. [46] compared the effectiveness of various stylized volume visualization techniques on the basis of shape perception.

Kim et al. [170] extended their previous work on detecting salient regions in images [171] to detect salient regions in both images and videos. Both approaches use the self-ordinal resemblance measure to compute the spatial saliency. Then the spatiotemporal saliency map is generated by combining the spatial saliency with the temporal saliency, which is the sum of absolute differences of temporal gradients of the center and the surrounding regions.

Lee et al. [172] presented mesh saliency, which is defined in a scale-dependent manner using a center-surround operator on Gaussian-weighted mean curvatures. They observed that their approach was able to capture the most visually interesting regions on a mesh. Kim and Varshney [18] introduced the use of center-surround operators to compute saliency fields of volume data. In their user study, they found that their approach was better at eliciting viewer attention than the traditional Gaussian regional enhancement approaches.

Shen et al. [173] proposed the use of saliency to assist volume exploration. They described a method for inferring interaction position in volume visualization, in order to help users pick focused features conveniently. Shen et al. [174] described spatiotemporal volume saliency, which extended the saliency field [18] to time-varying volume data.

2.9 Perceptual Evaluation

Traditional, visualization research has an emphasis on solving problems with modeling and optimization using engineering and mathematics tools. A recent focus in visualization has become measuring the effectiveness of a proposed method with adequate user studies [175]. Due to the complex nature of the data being studied, simply displaying all available information does not adequately meet the demands of domain scientists [176]. User studies can be used to evaluate the strengths and weaknesses of visualization methods [177]. The evaluation of visualization methods that focus on human factors often employ user studies or expert evaluations to determine their effects on interpretation and usability.

There are a number of different evaluation strategies, such as measuring user performance, accuracy and experience [178]. Laidlaw et al. [179] compared six methods for visualizing 2D vector fields and measured user performance on three flow-related tasks for each of the six methods. They used the evaluation results to identify what makes a 2D vector fields visualization effective. Joshi and Rheingans [136] evaluated the effective of their illustrative techniques by measuring user accuracy, time required to perform a task and user confidence.

Lu [180] described a method for automatically selecting rendering parameters to simplify user interaction and improve usability. Subsequently, a user study was conducted to evaluate the effectiveness of this method. Two data sets were rendered in three styles with three predefined portions of the data sets were highlighted respectively. The users' eye gaze patterns were analyzed to determine if they were able to accurately identify the highlighted areas in the images.

Kersten-Oertel et al. [181] presented empirical studies on the effect of six different perceptual cues for enhancing depth. In the user study, the subjects were asked to determine which one of two indicated vessels was closer to them, and were asked to respond as accurately and quickly as possible. Both the percentage of correct answers and response times were analyzed.

Also for depth perception, Díaz et al. [182] conducted a user study to investigate the impact of well-known volumetric shading models [183] in stereoscopic desktop-based environments. In the results, the average time spent and average correctness of answers were analyzed.

2.10 Summary

We have presented a review of literature in the field of volume visualization. Specific areas which were reviewed in detail include transfer functions, automated transfer function generation, visibility histograms, information theory and computational saliency. Transfer functions are relevant to all of the following chapters as transfer function optimization is a central theme of this thesis. Information theory is relevant to Chapter 3. Visibility and computational saliency are relevant to Chapter 4 and Chapter 5. Transfer functions for time-varying volume visualization relates to Chapter 6

The review suggests that it is feasible to optimize the parameters of volume visualization based on the information within volume data. Existing research on automated transfer function generation, information theory, visibility histograms and computational saliency of volume data provides us the foundation to explore and understand how this optimization may be achieved. Further studies are required in order to better integrate visibility and computational saliency of volume data into the visualization pipeline and enhance the expressiveness by exploiting the information within volume data sets.

Chapter 3

Information-Guided Transfer Function Refinement for Exploring Volume Data

In this chapter, we describe a global optimization and two user-driven refinement methods for modulating transfer functions in order to assist the exploration of volume data. This optimization is dependent on the distribution of scalar values of the volume data set and is designed to reduce general occlusion and improve the clarity of layers of structures in the resulting images. The user can explore a volume by interactively specifying different priority intensity ranges and observe which layers of structures are revealed. In addition, we show how the technique can be applied to time-varying volume data sets by adaptively refining the transfer function based on the histogram of each time-step. Experimental results on various data sets are presented to demonstrate the effectiveness of our method.

3.1 Introduction

Volume visualization is a powerful technique for depicting layered structures in 3D volume data sets. Transfer functions play an essential role in volume visualization. Through the assignment of visual properties, including color and opacity, to the volume data being visualized, transfer functions impact the final rendering of the data set and thus affect which structures will be visible to the user. However, obtaining an effective transfer function is a non-trivial task, which often entails time-consuming tweaking until a desired aesthetic quality is achieved in the resulting rendering. Although a number of automatic or semi-automatic approaches have been developed, transfer function design remains a challenging problem [19] [60].

For end users, such as physicians, who may not have much experience in volume rendering and transfer function design, a user-friendly approach that allows them to intuitively explore volume data sets is very desirable. As fully automatic approaches cannot currently guarantee satisfactory results in every case, exploratory approaches with simple and efficient interaction are highly desirable.

We propose a novel approach to refine the transfer function based on the distribution of the scalar values of the volume data set. Firstly, we propose an automatic step to refine the transfer function that improves the rendering of volume data by reducing overall occlusions with no previous assumptions of the data set. Furthermore, we propose two interactive methods that extend on the optimization technique in order to enhance specific intensity ranges within the data as identified by the user. The process is fast and intuitive and allows the user to provide customized views of the data to aid in visual exploration of the volume data set.

3.2 Related Work

A number of approaches have been proposed to automate the design of transfer functions, and these are discussed in detail in Section 2.2. Here, we briefly discuss the most closely related previous works to the contribution of this chapter.

Maciejewski et al. [72] described a method to structure attribute space in order to guide users to regions of interest within the transfer function histogram. Chan et al. [92] developed a system to optimize transparency automatically in volume rendering based on Metelli's episcotister model to improve the perceptual quality of transparent structures. Correa and Ma [5] proposed the visibility histogram to guide the transfer function design. In a later work [63], they generalized the visibility histogram and proposed a semi-automatic method for generating transfer functions by maximizing the visibility of important structures based on the visibility histogram, which represents the contribution of voxels to the resulting image. Ruiz et al. [66] also used visibility

air	fat	soft tissue	bone (cancellous/dense)
-1000	-100 to -50	+100 to +300	+700 to +3000

Table 3.1: Hounsfield units of some typical substances [1]

as a main parameter for the transfer function specification. Their method obtains the opacity transfer function by minimizing the informational divergence between the visibility distribution captured by a set of viewpoints and a target distribution defined by the user. Later, Bramon et al. [67] extended this approach to deal with multimodal information.

3.3 Background

3.3.1 Transfer Function Specification

In the specification of a 1D (intensity-based) transfer function, the user essentially assigns a color and/or opacity to a certain point in the histogram of scalar values in the data set. In practice, the user would be presented with an interface that allows them to set up several control points which corresponds to a certain kind of material or structure. The user then defines a mapping from each control point to some visual property (e.g. color) resulting in voxels of the corresponding intensity to be rendered in that color. Figure 3.1 displays four typical shapes used in transfer function design. If a volume data set contains complex structures, tent-like shapes are desirable in revealing isosurfaces of structures and seeing through inner structures. Otherwise, the ramp shape and other shapes can also reveal structures effectively.

In order to design transfer functions effectively, it is commonly required that users have prior knowledge about which intensity ranges are relevant or which regions should be emphasized in the data. This is especially the case in medical visualization. For instance, in computed tomography (CT) data the intensity ranges are determined by the Hounsfield scale (Table 3.1). The user may expect the constituent's intensity of CT data to follow the Hounsfield scale and thus set up control points accordingly.

Another consideration is that interior structures are likely to comprise far fewer voxels and are often occluded by the surrounding material. Consider the transfer function in Figure 3.2. The user finds three intensity intervals of interest and then sets

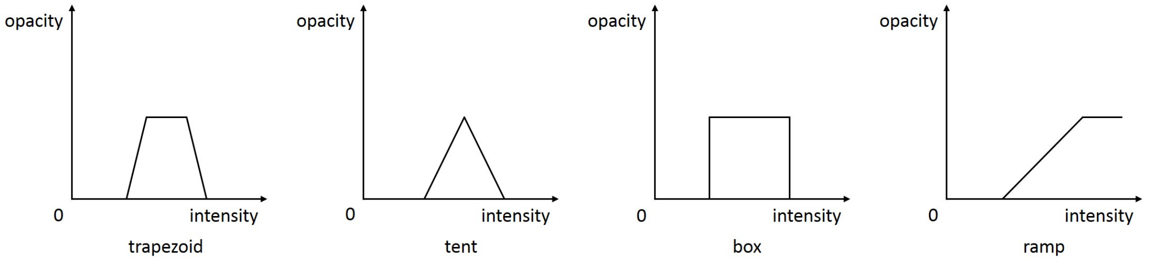


Figure 3.1: Typical transfer function shapes [7]

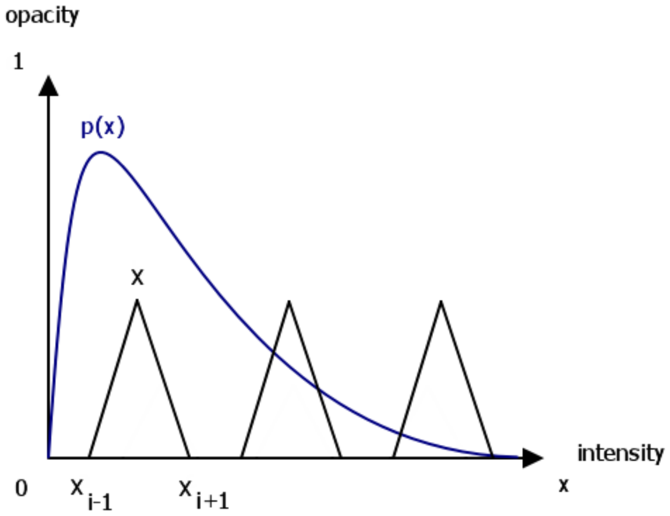


Figure 3.2: A transfer function with tent-like shapes

up three sets of control points in order to visualize these intensity intervals. The opacity of the three peak control points are assigned equally as they are equally important. However, if the distribution of voxels follows $p(x)$ (the blue curve), the voxels of the leftmost intensity intervals may completely occlude voxels of the other two intensity intervals in the resulting image. The global optimization in our approach aims at reducing this kind of occlusion by modulating the opacity of the transfer function based on the entropy of volume data, which is described in Section 3.3.2. Although many shapes can be used in transfer function design, tent-like shapes are often sufficient to model the user's intent [7]. In this chapter, results of both transfer functions of tent-like shapes and continuous transfer functions are used as input of the optimization.

3.3.2 Entropy of Volume Data

In computer graphics, information-theoretic measures, such as entropy and mutual information, have been applied to solve multiple problems in areas such as view selection [12] [184], flow visualization [158], multi-modal visualization [64] [67] and transfer function design [65] [80]. Information theory provides a theoretic framework to measure the information content (or uncertainty) of a random variable represented as a distribution [14]. Consider a discrete random variable X which has a set of possible values $\{a_0, a_1, ..., a_{n-1}\}$ with probabilities of occurrence $\{p_0, p_1, ..., p_{n-1}\}$, we can measure the uncertainty of the outcome with the entropy H(X), which is defined by

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{3.1}$$

where the summation is over the corresponding alphabet and the convention $0 \log 0 = 0$ is taken. The term $-\log p(x)$ represents the information content associated with the result x. If the entire volume data set is treated as a random variable, $I(a_x) = -\log p(x)$ represents the information content of a voxel a_x with intensity x, and the entropy gives us the average amount of information of a volume data. The probability p(x) is defined by $p(x) = \frac{n_x}{n}$, where n_x is the number of voxels with intensity x and n is the total number of voxels in the volume data.

Bordoloi and Shen [12] described a noteworthiness factor to denote the significance of the voxel to the visualization. The noteworthiness should be high for the voxels which are desired to be seen, and vice versa. The noteworthiness of voxel j is defined as $W_j = \alpha_j I_j = -\alpha_j log f_j$, where α_j is the opacity of voxel j looked up from the transfer function, I_j is the information carried by voxel j, which can be derived from the frequency of its histogram bin f_j. -log f_j represents the amount of information associated with voxel j.

3.4 Method

In this section, we present a transfer function refinement approach for modulating the opacity associated with the control points in a transfer function and combine it with user interaction to specify priority areas or intensity values of importance in the resulting image. In addition, an interaction widget (as in Figure 3.8) is introduced to allow users to explore the data sets by emphasizing certain intensity values and see the optimized output immediately.

In our approach, the user has control of the transfer functions by setting up control points as input for the optimization or tweaking the resulting transfer functions after the optimization. For example, the user can leave out less relevant data ranges by not covering the data ranges with shapes formed by control points (as in Figure 3.1). In the case of refining existing transfer functions, users also have the flexibility to refine the input transfer functions partially and keep certain control points constant during the optimization.

3.4.1 Weighting of Transfer Function Components

The goal of our transfer function refinement approach is to balance the opacity settings so that voxels of more significance contribute more and voxels of less significance contribute less to the resulting images. Given control points $v_1, v_2, ..., v_n$ of the transfer function with intensity values $x_1, x_2, ..., x_n$ and corresponding opacity values $\alpha(x_1), \alpha(x_2), ..., \alpha(x_n)$, the intensity range of the transfer function is normalized to $[0, 1]$. For the convenience of discussion, two control points v_0 and v_{n+1} are added to the lower bound and the upper bound respectively, and $x_0 = 0$, $\alpha(x_0) = 0$, $x_{n+1} = 1$ and $\alpha(x_{n+1}) = 0$.

Similar to the noteworthiness factor by Bordoloi and Shen [12], opacity and probability (derived from the intensity histogram) are also used in our weighting. We define the significance factor of the intensity x as

$$s(x) = -\alpha(x)p(x)\log p(x), x \in [0,1] \tag{3.2}$$

In the significance factor $s(x)$, $p(x)$ is computed from the histogram of the data set, and $\alpha(x)$ is the opacity function that we want to modulate. The significance factor should be high for the voxels which are desired to be seen, and vice versa. Then we define the weight of the i-th edge (the segment between v_i and v_{i+1}) as

$$e(i) = \int_{x\in[x_i,x_{i+1}]} s(x)dx \tag{3.3}$$

where $i \in [0,n]$ and $x \in [0,1]$.

Hence, the energy function of the transfer function can be defined as the variance of edge weights

$$E = \sum_{i=0}^{n}(e(i) - \overline{e(i)})^2 \tag{3.4}$$

where $\overline{e(i)}$ is the mean of edge weights, i.e.

$$\overline{e(i)} = \frac{\sum_{i=0}^{n} e(i)}{n} \tag{3.5}$$

Consequently, minimizing the energy function is equivalent to flattening the curve of the edge weights.

3.4.2 Optimizer

Constraints are introduced in the search of the parameter space. Control points would only be moved vertically in the transfer function domain. In other words, only the opacity associated with control points would be changed. The intensity of control points remains the same. Also, those control points that are marked as constant would not be updated in the optimization process. These constraints are based on our assumption that the intensity intervals associated with control points are the user's intensity intervals of interest. The user has explored the volume data and set up the transfer function according to his/her needs. Our algorithm aims to help the user

data set	transfer function (TF)	time for 500 iterations	time for 1000 iterations
CT-Knee	TF in Figure 3.6 (b)	0.002	0.005
CT-Knee	TF in Figure 3.4 (b)	0.005	0.009
VisMale	TF in Figure 3.6 (b)	0.002	0.005
VisMale	TF in Figure 3.4 (b)	0.005	0.008

Table 3.2: Computation time (seconds) for optimizations

reduce occlusion while preserving the user's knowledge or judgments of the data set.

A greedy strategy is employed in our algorithm to minimize the energy function. In each iteration, two operations are performed:

- Find the edge with the highest weight in the transfer function and reduce the opacity of the control point at its upper end (the vertex with a larger significance factor in the edge's two adjacent vertices).
- Find the edge with the lowest weight in the transfer function and increase the opacity of the control point at its lower end (the vertex with a smaller significance factor in the edge's two adjacent vertices).

In our implementation, the optimizer terminates when the energy function becomes stable, i.e. further iterations do not change the resulting transfer function. For the sample data sets we have tested, we observed that there is no further change to the resulting transfer function after 500 iterations. In addition, we measured the computation time of the optimization on two data sets with a continuous transfer function and a transfer function of tent-like shapes respectively. The tests in Table 3.2 all finished within 0.01 seconds, which shows that our approach is very lightweight. This suggests that, in practice, it would be viable to let the optimization continue to a conservatively high iteration count to ensure reasonable chance of convergence, without affecting interactivity.

The two step sizes in reducing opacity and increasing opacity can both be user-specified, or the first one is user-specified and the second one is computed based on the first one and the ratio of the significance factors of the two chosen control points. The ratio of the two step sizes affects the overall opacity of the resulting image, for instance, the image becomes more opaque or translucent. See Section 3.5.1 for results of this global refinement.

3.4.3 Prioritizing Intensity Ranges

The above described optimizer is an approach to balance the global opacity and thus reduce occlusions in the rendered images. In other words, we de-emphasize the most prevalent voxels, which are considered to have a high probability of occluding the rest of the scene and in particular interior structures of the data.

Although global optimization can help deliver images with better overall visibility, small details may be under-enhanced in the global optimization and certain structures in the image may have to be further enhanced for specific purposes. For instance, in an anatomical data set, the global optimization may guarantee that all structures of materials such as skin, bone and flesh are all visible however if the task of the user is specifically to study skin, this may be counter-productive. Thus, it is clear that a flexible method guided by user interactions is necessary to achieve various visualization goals.

In this section, we describe two alternative methods for prioritizing specific intensity ranges in the volume data. The first approach allows the user to interactively select a specific intensity range target that they are interested in and a hue-based distance factor is used to emphasize different intensity ranges based on the target identified by the user. Secondly, we describe a region-based optimization to provide an intuitive method of interaction by choosing regions of interest in the image that are then enhanced in the final rendering.

Distance Factors for User-Selected Intensity Values

We introduce distance factors for intensity values to prioritize the specific intensity ranges. In our implementation, users can select intensity values that they would like to enhance by clicking on a color palette, which uses the same color map in the transfer function. Assume each control point is assigned a unique intensity value, we can get the difference between a specific intensity value and the intensity value of a control point.

Given the selected intensity x_0, we define the distance factor of the i-th control point $x(i)$ as

$$D_h(i) = |x_0 - x(i)|, i \in [0, n+1] \tag{3.6}$$

Linear interpolation is used to obtain the distance factor $d_h(x)$ for the intensity $x \in$

$[x_i, x_{i+1})$

$$d_h(x) = D_h(i) + (D_h(i+1) - D_h(i))\frac{x - x_i}{x_{i+1} - xi} \tag{3.7}$$

Therefore, we define the weight of the i-th edge (the segment between v_i and v_{i+1}) with the distance factors as

$$e_h(i) = \int_{x\in[x_i, x_{i+1}]} d_h(x)s(x)dx \tag{3.8}$$

where $s(x) = -\alpha(x)p(x)\log p(x), x \in [0,1], i \in [0,n]$.

Hence, the energy function with distance factors for the intensity x_0 of the transfer function can be defined as the variance of edge weights with the distance factors

$$E_h = \sum_{i=0}^{n}(e_h(i) - \overline{e_h(i)})^2 \tag{3.9}$$

where $\overline{e_h(i)}$ is the mean of edge weights $e_h(i)$.

To use the energy function E_h described in this section, we simply need to replace the original energy function (mentioned in Section 3.4.1) with E_h in the previously described optimization algorithm.

Distance Factors for User-Selected Regions

Selecting a specific target intensity value as in the previous section provides a means of prioritizing the visualization towards the user's requirement. However, in practice, the user often has an imprecise idea of the exact intensity range that he/she wishes to focus on. Thus, we provide a more intuitive option that allows the user to prioritize the visualization by selecting a region on the screen containing voxels in the intensity range of interest to the user.

We introduce region-based distance factors to prioritize the user's region of interest. Assume each control point is assigned a unique color, we can get the difference between the color of a pixel in the region (which is selected in image space) and the color of a control point (HSV color space is used in our implementation).

Given the color of the i-th control point is $c(i)$, the distance between the color of a pixel r in the region R and the color of the i-th control point is denoted by $d(r, c(i))$. In our approach, the sum of the distances D between each pixel in the region R and the i-th control point is used to measure the difference between the region and the control point.

$$D(R,i) = \sum_{r \in R} d(r, c(i)), i \in [0, n+1] \tag{3.10}$$

Given the selected region R, we define the distance factor of the i-th control point as

$$W_R(i) = \frac{D(R,i)}{\sum_{i=1}^{n} D(R,i)}, i \in [0, n+1] \tag{3.11}$$

Linear interpolation is used to obtain the distance factor $w_R(x)$ for the intensity $x \in [x_i, x_{i+1})$

$$w_R(x) = W_R(i) + (W_R(i+1) - W_R(i)) \frac{x - x_i}{x_{i+1} - xi} \tag{3.12}$$

Therefore, we define the weight of the i-th edge (the segment between v_i and v_{i+1}) with the distance factors as

$$e_R(i) = \int_{x \in [x_i, x_{i+1}]} w_R(x) s(x) dx \tag{3.13}$$

where $s(x) = -\alpha(x) p(x) \log p(x), x \in [0,1], i \in [0, n]$.

Hence, the energy function with distance factors for the region R of the transfer function can be defined as the variance of edge weights with the distance factors

$$E_R = \sum_{i=0}^{n} (e_R(i) - \overline{e_R(i)})^2 \tag{3.14}$$

where $\overline{e_R(i)}$ is the mean of edge weights $e_R(i)$.

Similarly, in order to use the energy function E_R, we need to replace the original energy function with E_R in the previously described optimization algorithm.

The distance factors described in this section measure the dissimilarity between a selected region and a control point. Therefore, the distance factor would be small if the region has an overall color similar to the color of the control point. Since we are minimizing the energy function, which is the variance of the edge weights, reducing the distance factors of those control points, which are related to the selected region, will result in their opacity values being increased. As a result, the features (in this case, the intensity intervals) in the selected regions will be enhanced and other features will be de-emphasized in the rendered image. See Section 3.5.3 for sample results of this optimization.

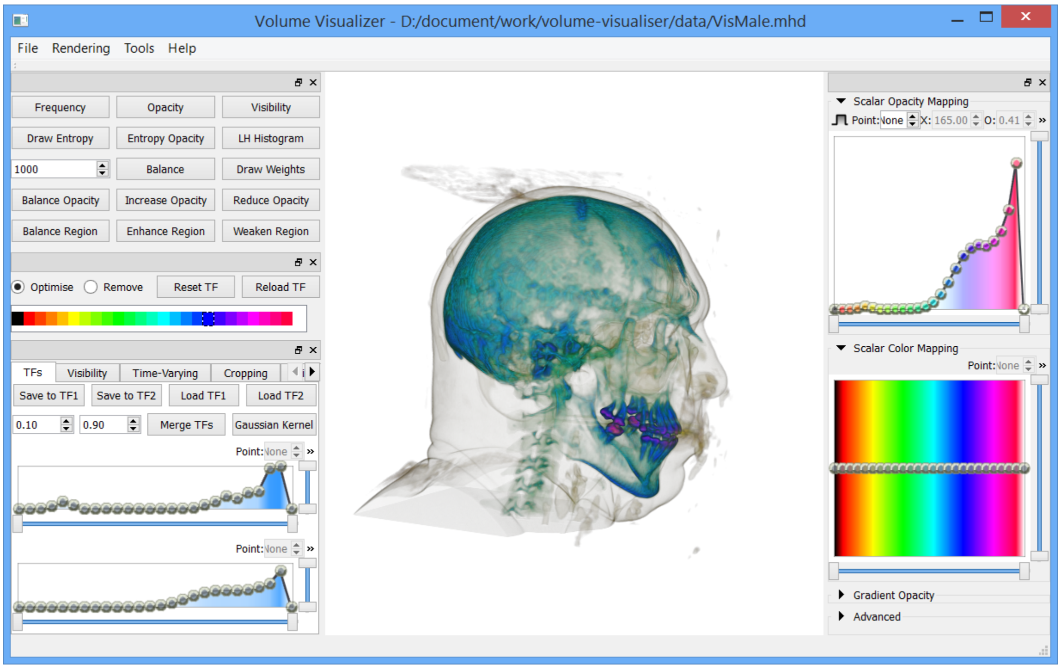
3.4.4 Adaptive Transfer Functions for Time-Varying Data Sets

In time-varying data sets, the data value ranges and distributions change among time-steps. A single global transfer function may not be able to adequately catch the details of the data set. Therefore, we exploit the transfer function optimizer (as discussed in Section 3.4.2) to locally refine the transfer function for each time-step in the data set. In this case, the user specifies a transfer function for a single time-step of the time-varying data set and using either of the two interaction methods (as discussed in Section 3.4.3) to specify priority intensity ranges. The transfer function designed for this time-step is taken as an input transfer function and optimized again based on the histogram of the next time-step. Subsequently, the output of next time-step is taken as input of the time-step after it and so forth.

3.5 Results and Discussions

In this section, we present some results to demonstrate the effectiveness of our approach on the CT-knee ($379 \times 229 \times 305$) and VisMale head ($128 \times 256 \times 256$) data sets [3] and a time-varying data set of a simulated turbulent vortex flow ($128 \times 128 \times 128$, 100 time-steps) [185]. Results were generated in our volume rendering system (Figure 3.3) on a computer equipped with an Intel Core i5-2410M CPU, 8GB of RAM and an NVIDIA GeForce GT 540M graphics card.

Automatically generated transfer functions with ramps and tent-like shapes are provided as initial input to the optimizer. Figure 3.4 displays a continuous transfer function. The ramps are formed by a series of control points with corresponding colors from the color map. Figure 3.6 displays a transfer function with several evenly distributed tent-like shapes. Each tent-like shape consists of a peak control point and two bottom control points. The peak control points are movable while the bottom control



1000

● Optimise ○ Remove

0.10 0.90

Point: None

Point: None X: 165.00 O: 0.41

Figure 3.3: A screenshot of our volume rendering system

points are marked as constant to maintain the tent-like shapes.

Note that the transfer functions consisting of equidistant and equal height tent-like shapes (as in Figure 3.6) are just provided as naive examples of user designed transfer functions and used as input to the optimizer. In practice, users would design transfer functions (which consist of ramps, tent-like or other shapes) according to the characteristic of the data sets and the intensity ranges that they are interested in.

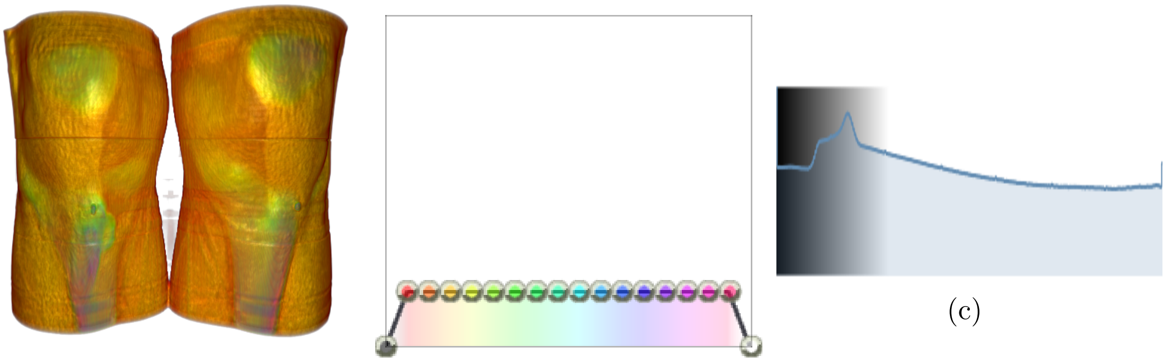
The initial opacity values of control points will affect the overall opacity level of the resulting image after optimization. Because there are omitted intensity ranges (the gaps) in transfer functions with tent-like shapes, the initial opacity values should be higher in transfer functions with tent-like shapes than in continuous transfer functions. In transfer functions with tent-like shapes (as discussed in Section 3.3), the opacity of the top control points are set to 1/2 (medium level of opacity) and the opacity of the bottom control points are set to 0. The bottom control points are fixed to 0 in order to keep the tent-like shapes in the transfer functions. By contrast, all control points in continuous transfer functions are movable vertically except that control points v_0 and v_{n+1} are fixed and serve as the boundary. The opacity of control points are set to 1/6 (an empirical value to make the volume rendering images to be of similar opacity level to tent-like transfer functions with opacity of 1/2) in continuous transfer functions. The color maps used in the transfer functions in our system are evenly sampled from a spectrum (with hue from 0° to 360° in HSV color space). In the optimization, the two step sizes for reducing opacity and increasing are both set to 1/256.

3.5.1 Automatic Transfer Function Refinement

Firstly, we demonstrate the global optimization discussed in Section 3.4.1 and Section 3.4.2 with continuous transfer functions. In Figure 3.4, the CT-Knee data set is rendered with a naive transfer function consisting of 6 tent-like shapes of various colors with equal opacity. Figure 3.5 shows the resulting image rendered with the optimized transfer function. We tested this specific example as joints are popular regions of interest in medical visualization. The knee in particular is a commonly studied joint. In Figure 3.4, only parts of the skeleton are visible. The rest is occluded by the surrounding material (such as the skin and muscles). After optimization (Figure 3.5), the surrounding tissues become translucent, hence the skeleton is exposed and the knee joint is visible, while the overall context is preserved. We argue that in the absence of any previous assumptions on what the user is looking for, the global optimization provides a more balanced initial view before deeper exploration of the data.

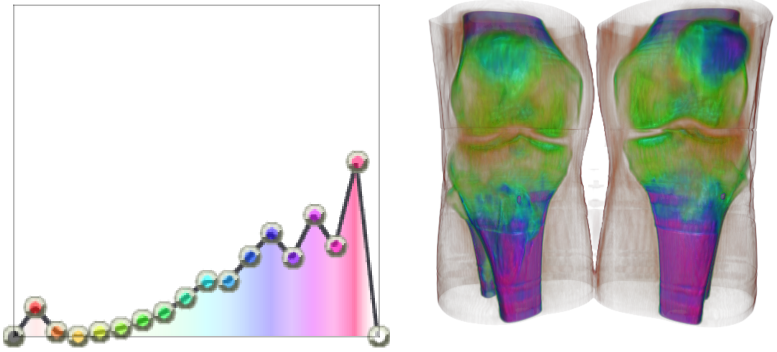
Although the continuous transfer function is useful in automatic transfer function generation, in typical volume visualization programs, users often prefer much more simplified transfer functions, e.g. a few control points such as in transfer functions with tent-like shapes. Figure 3.6 is the CT-Knee data set rendered with the transfer function. We show in Figure 3.7 that our optimization can also benefit such simpler transfer functions.

In practice we observed that the energy function usually converges to a small but non-zero value. As the number of control points increases, it takes more iterations for the optimization to achieve a stable state - a number of tent-like shapes ranging from 4 to 16 was found to be the most effective.



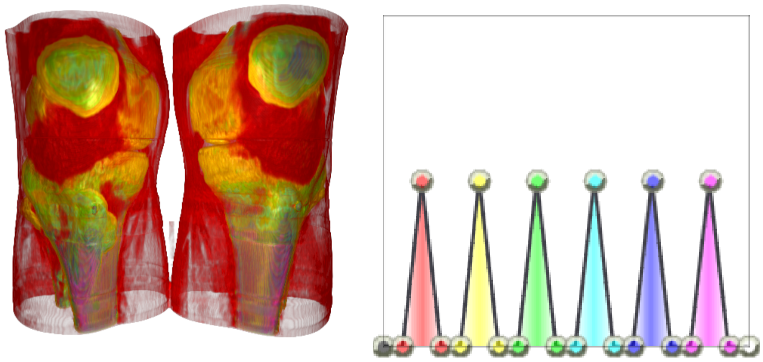
(a) (b) (c)

Figure 3.4: Before optimization: CT-Knee with a continuous transfer function (a) Preliminary view of data set (b) A continuous transfer function with a ramp (c) Histogram of the data set



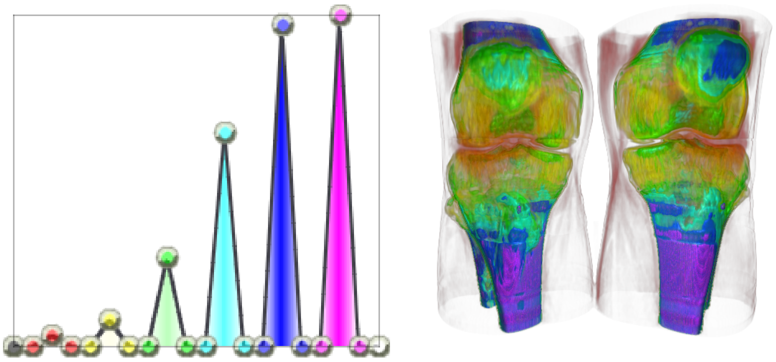
(a) (b)

Figure 3.5: The transfer function from Figure 3.4 after optimization: (a) Optimized transfer function) (b) Optimized output



(a) (b)

Figure 3.6: Before optimization: CT-Knee rendered with a transfer function consisting of tent-like shapes (a) Preliminary view of data set (b) A transfer function with 6 tent-like shapes



(a) (b)

Figure 3.7: The transfer function from Figure 3.6 after optimization: (a) Optimized transfer function (b) Optimized output

3.5.2 Transfer Function Refinement with User-Selected Intensity Values

Figure 3.9 shows three images of a CT-Knee data set where different colors are selected for optimization based on the approach discussed in Section 3.4.3. Since the colors of the transfer function are generated in HSV color space by varying the hue component, the difference of intensity values are mapped to the difference of hue in HSV color space. By clicking on the color palette (Figure 3.8), the transfer function is instantly optimized for the corresponding intensity values.

3.5.3 Transfer Function Refinement with User-Selected Regions

Figure 3.10a shows the VisMale data set with a generated transfer function of 4 tent-like shapes. Figure 3.10c shows the intensity histogram of the data set. After the optimization (Figure 3.10b) based on Section 3.4.2, the outside of the head is less opaque so the inner structures are revealed to the user. However, the intermediate material (i.e. the skull) also becomes less clear. If the goal is to make the skull more visible, the user could select a region consisting of parts of the skull to generate a weighting and perform further optimization of the transfer function based on the approach discussed in Section 3.4.3. If the material of interest is occluded by surrounding materials, the user could use an axis-aligned clipping plane in order to accurately select voxels of the skull while minimizing the accidental tagging of the surrounding material (Figure 3.10d). As shown in Figure 3.10e, the skull becomes more clear after the region-based optimization.

3.5.4 Adaptive Transfer Functions for Time-Varying Data Sets

We demonstrate our approach on a turbulent vortex data set [186], which consists of 100 time-steps. Our optimizer adaptively propagates transfer functions for the time-varying data set. Specifically, the transfer function of the previous time-step is taken as input to generate the transfer function for the next time-step. Therefore, the transfer

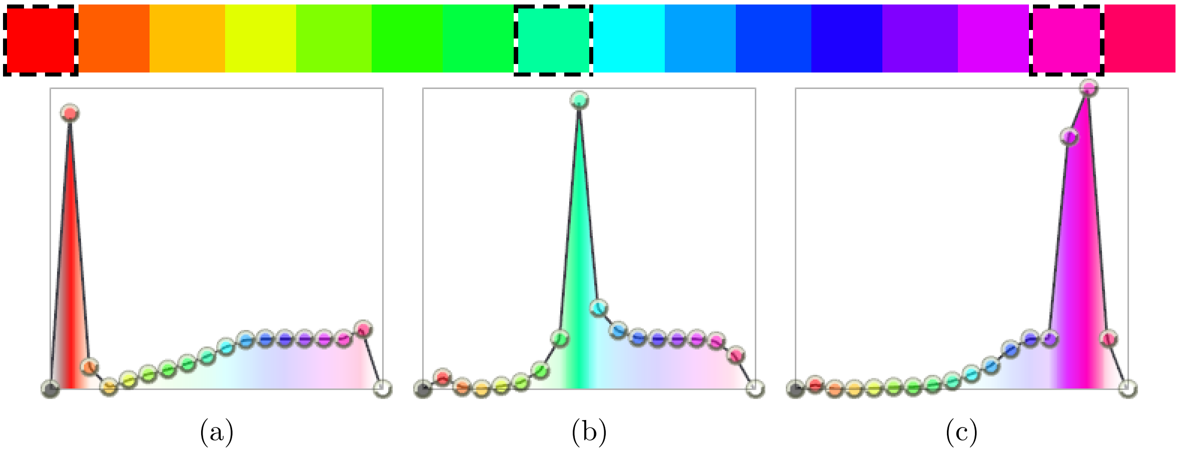


Figure 3.8: The 3 chosen colors (corresponding to different intensity values) and the transfer functions after optimization for each intensity value. The same transfer function as shown in Figure 3.4 is used as input to the optimizer. Note how the transfer functions are enhanced for the specific intensity ranges as compared to the result of global optimization in Figure 3.5.

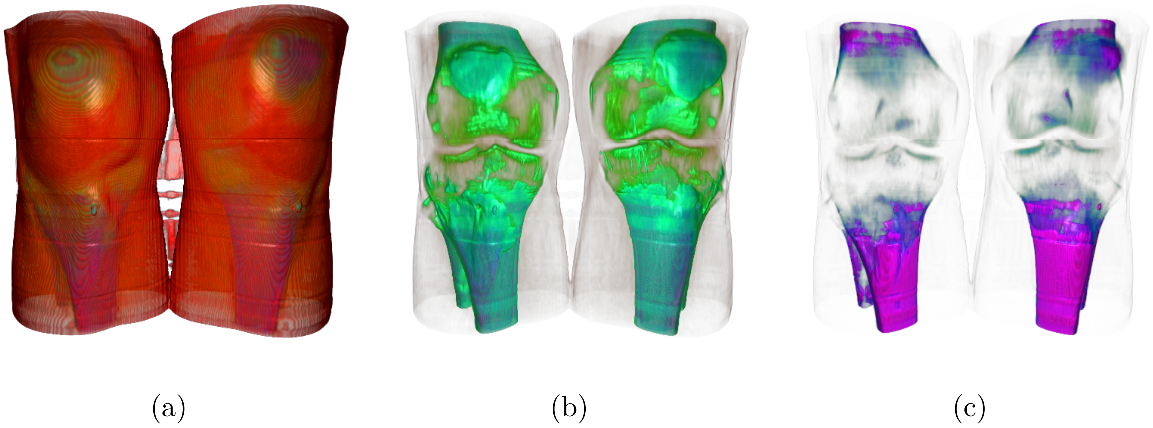
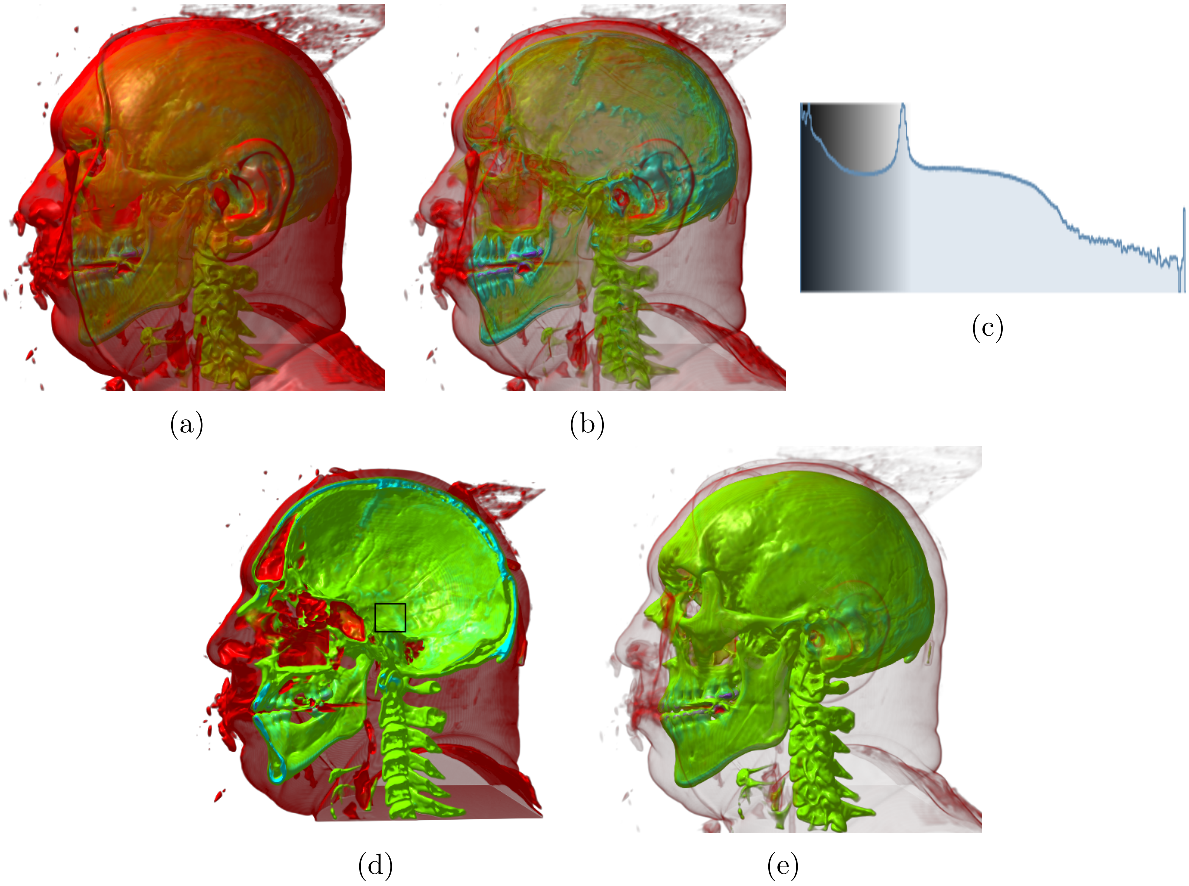


Figure 3.9: The CT-Knee data with transfer functions optimized for the 3 colors in Figure 3.8. (a) The materials with intensity values mapped to red are enhanced. Similarly, the materials in green and magenta are enhanced respectively in (b) and (c).



(a)

(b)

(c)

(d)

(e)

Figure 3.10: (a) Preliminary view of the VisMale data set. (b) Optimized output. (c) Histogram of the data set. (d) The user selects a region on the skull under a clipping plane. (e) Optimized output: skull is enhanced and the outer layer is de-emphasized.

functions are locally optimized for each time-step of the time-varying data set, but as the input transfer function comes from the immediate preceding time-step, the resulting transfer functions also exhibit reasonable temporal coherency assuming the data set is also coherent. As the difference of intensity histograms among consecutive time-steps is hard to notice, only the images of the first time-step and a time-step in the middle of the data set are displayed here. Figure 3.12 shows three images of the vortex data set at time-step 0 while optimized for the three colors chosen in the color palette in Figure 3.11. Figure 3.13 displays the vortex data set at time-step 50 with the transfer functions optimized for the three chosen colors. Figure 3.14 shows the intensity histograms of the two time-steps discussed above and the corresponding optimized transfer functions for the colors selected in Figure 3.11.

3.6 Conclusions

In this chapter, we presented a global optimization technique which aims to alleviate excessive occlusion problems in volume rendering. However, instead of computing the view-dependent visibility of each voxel as is necessitated in other similar approaches [63] [66], we achieve this by balancing the opacity of voxels based on the distribution of intensity values. Our view-independent approach is relatively lightweight and should have better performance in contrast to other techniques. In addition, we propose two interactive methods that extend on the optimization technique in order to enhance specific intensity ranges within the data as identified by the user. This mechanism provides the ability for users to intuitively specify priority intensity ranges, thus facilitating the exploration of both static and time-varying volume data sets.

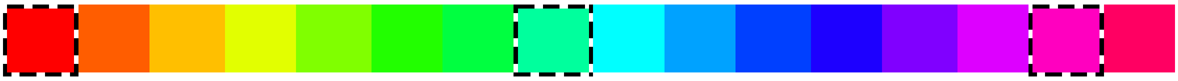
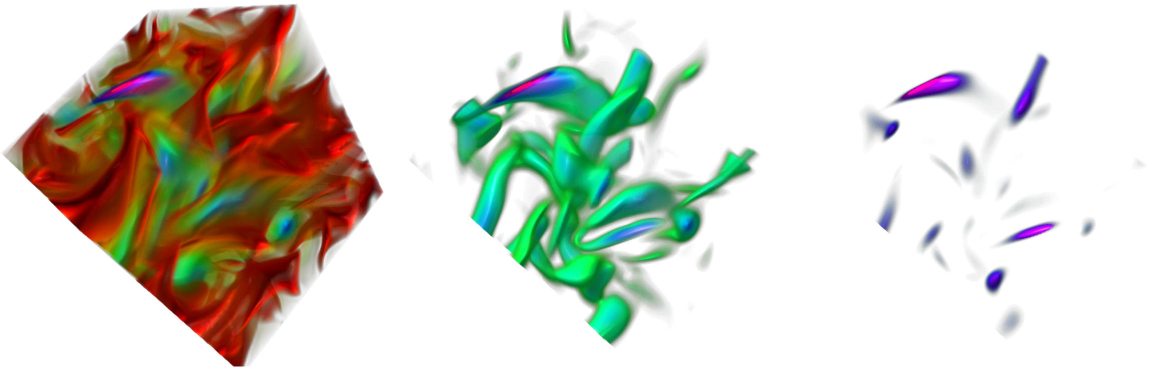
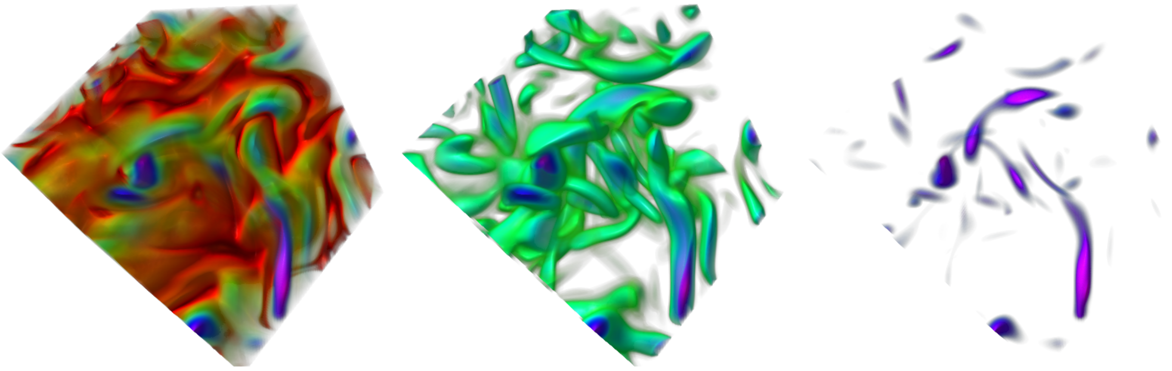


Figure 3.11: The 3 chosen colors (corresponding to different intensity values) for optimization. Note how different parts of the data set are enhanced respectively in Figure 3.12 and Figure 3.13.



(a) (b) (c)

Figure 3.12: The vortex data at time-step 0. (a) The materials with intensity values mapped to red are enhanced. Similarly, the materials in green and magenta are enhanced respectively in (b) and (c).



(a) (b) (c)

Figure 3.13: The vortex data at time-step 50. These images show similar results as those for time-step 0, because there are only limited changes among the histograms of different time-steps.

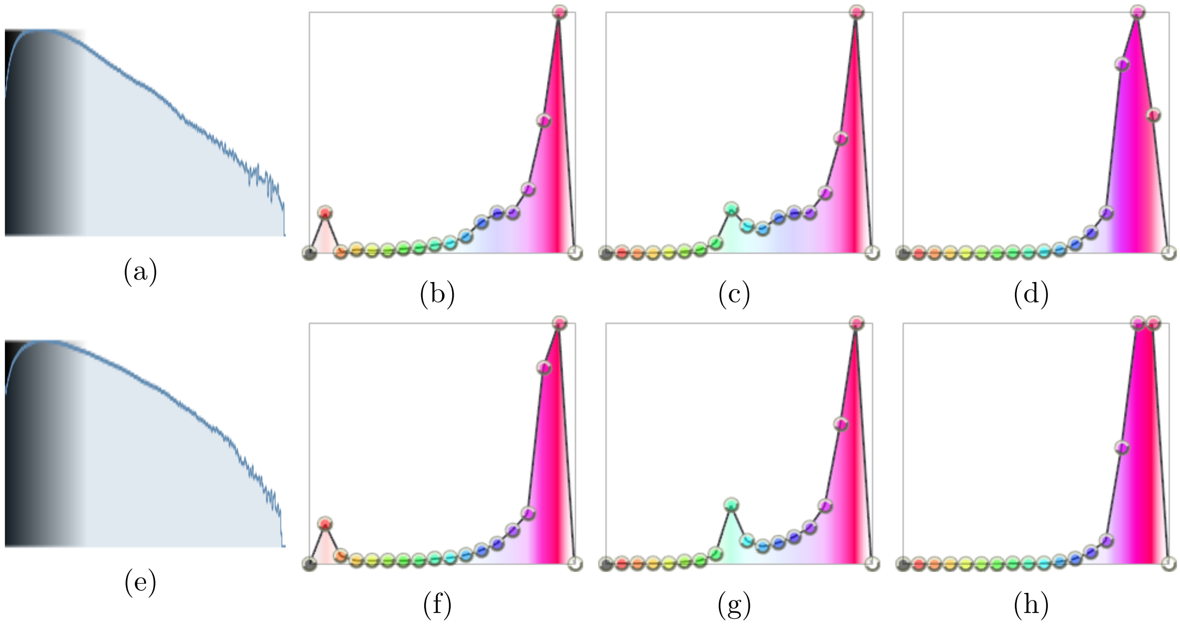


Figure 3.14: (a) Histogram of time-step 0. (b) Transfer function (TF) for Figure 3.12a. (c) TF for Figure 3.12b. (d) TF for Figure 3.12c. (e) Histogram of time-step 50. (f) TF for Figure 3.13a. (g) TF for Figure 3.13b. (h) TF for Figure 3.13c.

Chapter 4

Visibility-Weighted Saliency for Volume Visualization

In volume visualization, the clarity of features depends on the transfer function, the viewpoint and spatial distribution of features in the volume data set. In this chapter, we propose visibility-weighted saliency as a measure of visual saliency of features in volume rendered images, in order to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. Visibility-weighted saliency is based on a computational measure of perceptual importance of voxels and the visibility of features in volume rendered images. The effectiveness of this scheme is demonstrated by test results on two volume data sets.

4.1 Introduction

A crucial step in volume visualization is transfer function specification. Transfer functions assign visual properties, including color and opacity, to the volume data being visualized. Hence, transfer functions determine which structures will be visible and how they will be rendered. An appropriate transfer function can quickly reveal large amounts of information of the data set to the viewer. However, obtaining an effective transfer function is a non-trivial task, which involves a significant amount of tweaking of color and opacity. A cause of this problem is the lack of an objective measure to quantify the quality of transfer functions [63].

Although user studies are useful in evaluating some fundamental characteristics of visualization techniques, it is not possible to conduct a user study for each individual visualization every time it is created. Several computational measures of visual saliency that model human attention have been developed [16] [17]. Kim and Varshney [18] introduced the saliency field, which measures visual saliency of voxels using the center-surround operator based on the difference of Gaussian-weighted averages at a fine and a coarse scale. However, salient voxels may be occluded by other voxels close to the viewer in certain viewpoints and thus these salient voxels become invisible in the volume rendered image. In order to measure the visual saliency of features in volume rendered images, it is necessary to consider both the saliency and the visibility of the voxels which form the feature.

In this chapter, we propose visibility-weighted saliency as an improved measure of the visual saliency of features in volume rendered images. Visibility-weighted saliency is a combination of feature visibility [6] and saliency fields [18]. Feature visibility measures the contribution of each feature to the volume rendered image and saliency fields measure how strongly each voxel stands out in its local neighborhood. The visibility-weighted saliency is presented in two different ways, i.e. visibility-weighted saliency fields and feature saliency histograms. Visibility-weighted saliency fields display the spatial distribution of visual saliency of features and feature saliency histograms provide quantitative information about the perceptual importance of the features. With visibility-weighted saliency, the saliency of features rendered in different viewpoints with different transfer functions can be measured in a quantitative and fully automated way. Thus, this technique can be used to guide users in choosing appropriate viewpoints and designing effective transfer functions for the features of interest in volume visualization. This technique is also useful for understanding how much different parts of the volume contribute to the final image and how different tissues occlude each other and interfere with each other's visibility.

4.2 Related Work

Several computational models of visual saliency for modeling human attention have been developed. Itti et al. [16] developed a computational model of visual attention based on the center-surround operators in an image. This center-surround mechanism has the intuitive appeal of being able to identify regions that are different from their surrounding context. Based on the perceptual principles, Chen et al. [92] introduced several image quality measures to enhance the perceived quality of semitransparent features. Jänicke and Chen [169] described a quality metric for analyzing the saliency of visualization images and demonstrated its usefulness with examples from information visualization, volume visualization and flow visualization.

Lee et al. [172] proposed saliency for meshes based on a multi-scale center-surround mechanism that operates on local curvature. Kim and Varshney [18] presented the use of a center-surround operator using the Laplacian of Gaussian-weighted averages of appearance attributes to enhance selected regions of a volume and validated their work using an eye-tracking user study. Shen et al. [174] extended this technique to spatiotemporal volume saliency to detect both spatial and temporal changes.

Visibility measures the impact of individual voxels on the image generated by a volumetric object and visibility distribution can be utilized as a measure on the quality of transfer functions as users explore the transfer function space. Visibility has been studied to measure the quality of a given viewpoint [12] [97] and to enhance the rendering process with cutaway views. Correa and Ma [63] introduced visibility histogram, which describes the accumulated visibility of each intensity value in the transfer function.

Ruiz et al. [66] proposed an automatic method to generate a transfer function by minimizing the Kullback-Leibler divergence between the observed visibility distribution and a target distribution provided by the user. Wang et al. [6] extended the idea of the visibility histogram to feature visibility and introduced an interaction scheme where the opacity of each feature was generated automatically based on user-defined visibility values. Visibility distribution is also used in automating color mapping [100] and 2D transfer functions [107].

4.3 Method

For 2D images, intensity and color are the most important attributes. In volume visualization, the intensity and color in the final images result from the blending of alpha and color determined by user-specified transfer functions in a specific viewpoint. The saliency field is a view-independent scalar field that contains the visual saliency of each voxel in the volume data. The visual saliency of voxels represents the perceptual importance in 3D space, however it does not reflect how visible the voxels are in the final 2D images.

In order to take into account both the visual saliency of voxels in 3D space and the contribution of the voxels to the final 2D images of the rendered data set, we propose a visibility-based saliency metric, which attempts to measure the impact of individual voxels as well as user-specified features on volume rendered images. This technique aims to assist users in gaining insight into the internal structure of the data set and understanding the contribution of different features to the final image.

In this section, we describe in detail the concepts of visibility fields, saliency fields, visibility-weighted saliency fields and visibility-weighted saliency. In order to better illustrate the effects of these techniques in our discussion, we present the results of applying these techniques to a synthetic data set from two different viewpoints.

4.3.1 Feature Definition

Before discussion of the proposed methods, we first define a feature F as a subset of voxels in the volume V, i.e. $F \subseteq V$. In the case of intensity-based 1D transfer functions, a feature can also be defined using an intensity interval, which is

$$F = \{a \leq I(v) < b | v \in V\} \tag{4.1}$$

where $I(v)$ is the intensity of voxel v and $[a, b)$ is the intensity interval that contains all the voxels of feature F.

4.3.2 Visibility Fields

Direct volume rendering (e.g. ray-casting) is a technique that renders a 2D projection of a 3D volume data set. The rendering of a volume, which essentially is a block of 3D data, involves alpha blending and color composition of voxels. The resulting 2D image is acquired by blending the color and opacity of voxels along the view direction. The transfer function determines the color and opacity of individual voxels based on their data attributes such as intensity. However, the contribution of a voxel to the rendered image is determined by both the opacity of this voxel and the opacity of those voxels in front of the current voxel in the view direction. This mechanism is described in the front-to-back compositing equations.

$$C_i = (1 - A_{i-1})c_i + C_{i-1} \tag{4.2}$$

$$A_i = (1 - A_{i-1})a_i + A_{i-1} \tag{4.3}$$

where a_i and c_i are opacity and color of voxel i, and A_i and C_i are the accumulated opacity and color at voxel i.

Therefore, the visibility of voxel i [13] can be calculated as

$$v_i = A_i - A_{i-1} = (1 - A_{i-1})a_i \tag{4.4}$$

and the visibility field is simply the visibility of all the voxels in the volume V

$$V = \{\, v_i \mid i \in V \,\} \tag{4.5}$$

The visibility field is dependent on both the viewpoint and the transfer function, therefore it can be used to analyze the visualization of the volume data. The visibility field is particularly useful for understanding what parts of the data set are being rendered and how different tissues occlude each other (Figure 4.1).

In terms of implementation, the computation of visibility fields can be performed in real-time on a GPU. Correa and Ma [63] employed a scattering approach for GPU-assisted computation of the visibility histogram, which scatters the pixel points to the right bin in the histogram. Wang et al. [6] used the multiple rendering targets (MRT) extension of OpenGL 2.0 and above to achieve the computation of visibility for up to 32 features. Instead of grouping visibility values into intensity bins to acquire visibility distribution over intensity ranges (histograms), we are interested in the actual spatial visibility distribution, i.e. visibility field. We perform slice-based rendering on a GPU by rendering a series of quads which are parallel to the viewing plane, one for each slice. The fragments which do not belong to the volume are discarded. Then the visibility values are computed by subtracting the accumulated opacity of the previous slice from that of the current slice. After collecting the visibility values of all voxels, the visibility field can be constructed.

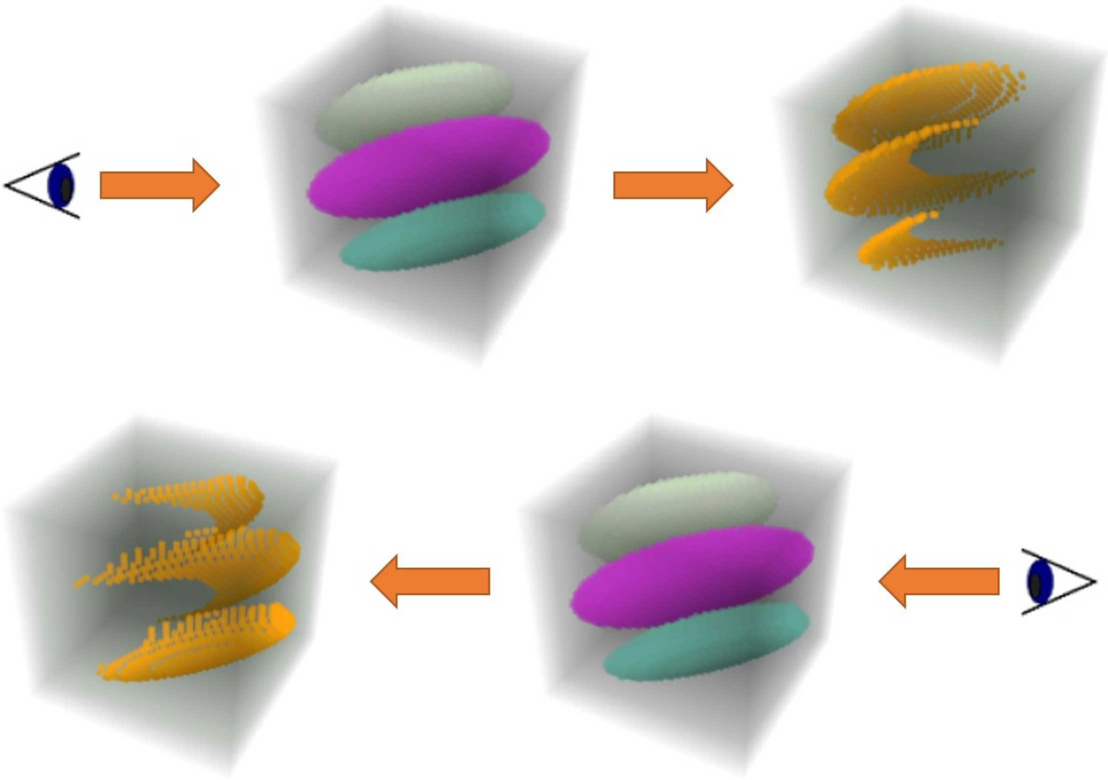


Figure 4.1: A synthetic volume data set consisting of three solid disk-like features is shown. The images in the first row show the final image and the corresponding visibility field from a viewpoint on the left. The images in the second row shows the final image and visibility field from a viewpoint on the right. The visibility fields display what parts of the volume contribute most to the images and how tissues in the front occlude those in the back.

4.3.3 Saliency Fields

Because viewers pay greater visual attention to regions that they find salient [187], many models of visual attention and saliency have been evaluated by their ability to predict eye movements. The saliency for a volume can be computed either by using eye-tracking data or through computational models of human perception. Once the saliency for a volume is acquired, it can be used to better inform the visualization process.

We use a center-surround operator that is similar to the work by Shen et al. [174] to compute the saliency field. Let the neighborhood $N(i, \sigma)$ for a voxel i be the set of voxels within a distance σ. Thus, $N(i, \sigma) = \{\, j \mid \|j - i\| < \sigma \,\}$, where j is a voxel. Let $G(O, i, \sigma)$ denote the Gaussian weighted average, then we have

$$G(O, i, \sigma) = \sum_{j \in N(i,\sigma)} O(j) g(i, j, \sigma) \tag{4.6}$$

where

$$g(i, j, \sigma) = \frac{exp[-\|j - i\|^2/(2\sigma)^2]}{\sum_{k \in N(i,\sigma)} exp[-\|k - i\|]^2/(2\sigma)^2} \tag{4.7}$$

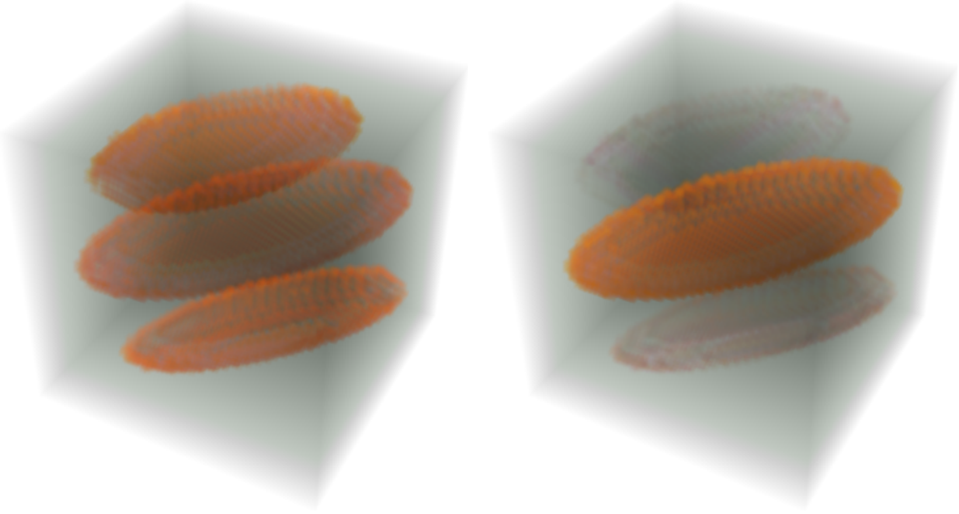
and O is a field of appearance attributes of every voxel in the volume and $O(j)$ is the appearance attribute of voxel j.

Then the saliency field is defined as the absolute difference of Gaussian-weighted averages

$$L(O, i, \sigma) = |w_1 G(O, i, \sigma) - w_2 G(O, i, 2\sigma)| \tag{4.8}$$

where w_1 and w_2 indicate the weights of the Gaussian-weighted averages at a fine scale and a coarse scale respectively.

Visual properties such as opacity and color values (e.g. brightness, saturation, hue) can be used as appearance attributes in the computation of a saliency field. Figure 4.2 displays the saliency fields computed from brightness and saturation of voxels respectively. Although opacity is an important visual property, the visibility field described in the previous section is derived from alpha blending, which has already taken the opacity of voxels into account. Therefore, we compute the saliency fields using brightness and saturation instead of opacity. Brightness and saturation are also the appearance attributes Kim and Varshney [18] used in their saliency-based enhancement operator.



(a) (b)

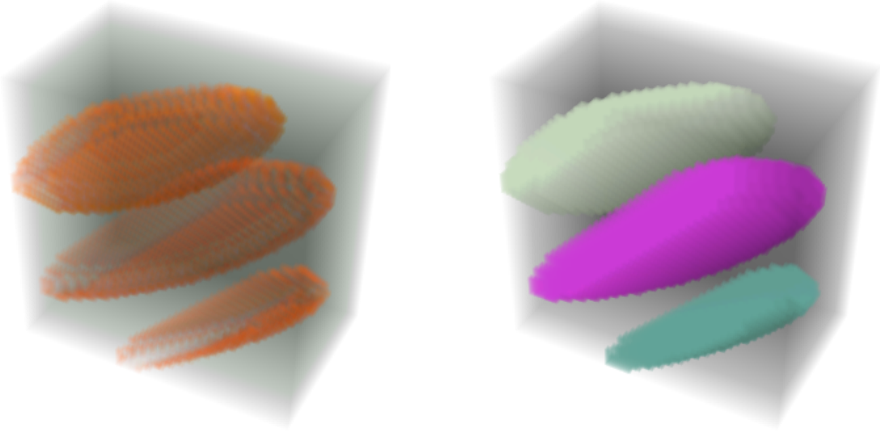
Figure 4.2: The saliency fields computed from brightness (a) and saturation (b) of voxels respectively.

The saliency field is acquired by applying the center-surround operator to appearance attributes of voxels. The effect of the center-surround operator is to emphasize the center and de-emphasize the surroundings of voxels. This is shown in the exploded view of the saliency field (computed from brightness) and the volume data in Figure 4.3.

In our implementation, we use perceptually uniform color spaces, e.g. CIELab and CIELCh. In CIELCh, instead of Cartesian coordinates a*, b*, the cylindrical coordinates C* (chroma, relative saturation) and h (hue angle in the CIELab color wheel) are specified, and the brightness L* remains the same. The advantage of using perceptually uniform color spaces is that the relative perceptual differences between two colors can be approximated by the Euclidean distance between the two colors in a three-dimensional space consisting of the three color components [188].

4.3.4 Visibility-Weighted Saliency Fields of Features

The visibility field indicates the contribution of voxels, which is how much each voxel contributes to the final image, and the saliency field indicates the conspicuity of voxels, which is how much each voxel stands out from its surroundings. The conspicuity in a 3D volume is similar to that in a 2D image and can be measured by the difference of



(a) (b)

Figure 4.3: The saliency fields emphasize the center and de-emphasize the surroundings of voxels. As in the clipped views of the saliency field (a) and the volume data set (b), the three solid disks are represented as hollow shapes in the saliency field.

visible properties between each location (voxel) and its surroundings [168]. It would be desirable to have an indicator that represents both the contribution and conspicuity of the voxels. Therefore, we propose a visibility-weighted saliency field, by weighting the saliency of voxels by their visibility, given the volume is rendered with a transfer function from a specific viewpoint. The visibility-weighted saliency for voxel i is

$$s_i(O, i, \sigma) = v_i L(O, i, \sigma) \tag{4.9}$$

Hence we define S as the visibility-weighted saliency field of the volume V.

$$S = \{\, s_i(O, i, \sigma) \mid i \in V \,\} \tag{4.10}$$

Therefore, we define visibility-weighted saliency field of a feature F in the volume V ($F \subseteq V$) as.

$$S_F = \{\, s_i(O, i, \sigma) \mid i \in F \,\} \tag{4.11}$$

Then we define visibility-weighted saliency of feature F as

$$W_F(O, i, \sigma) = \frac{\sum_{i \in F} s_i(O, i, \sigma)}{\sum_{i \in V} s_i(O, i, \sigma)} \tag{4.12}$$

Since F is a subset of V, $W_F(O, i, \sigma)$ must be in the interval $[0, 1]$. S_F can be used as a score to indicate the saliency of feature F in terms of the appearance attribute O.

Features can be defined by user-specified transfer functions or segmentation of the volume data. Figure 4.4 illustrates the visibility-weighted saliency fields of the three disk-like features.

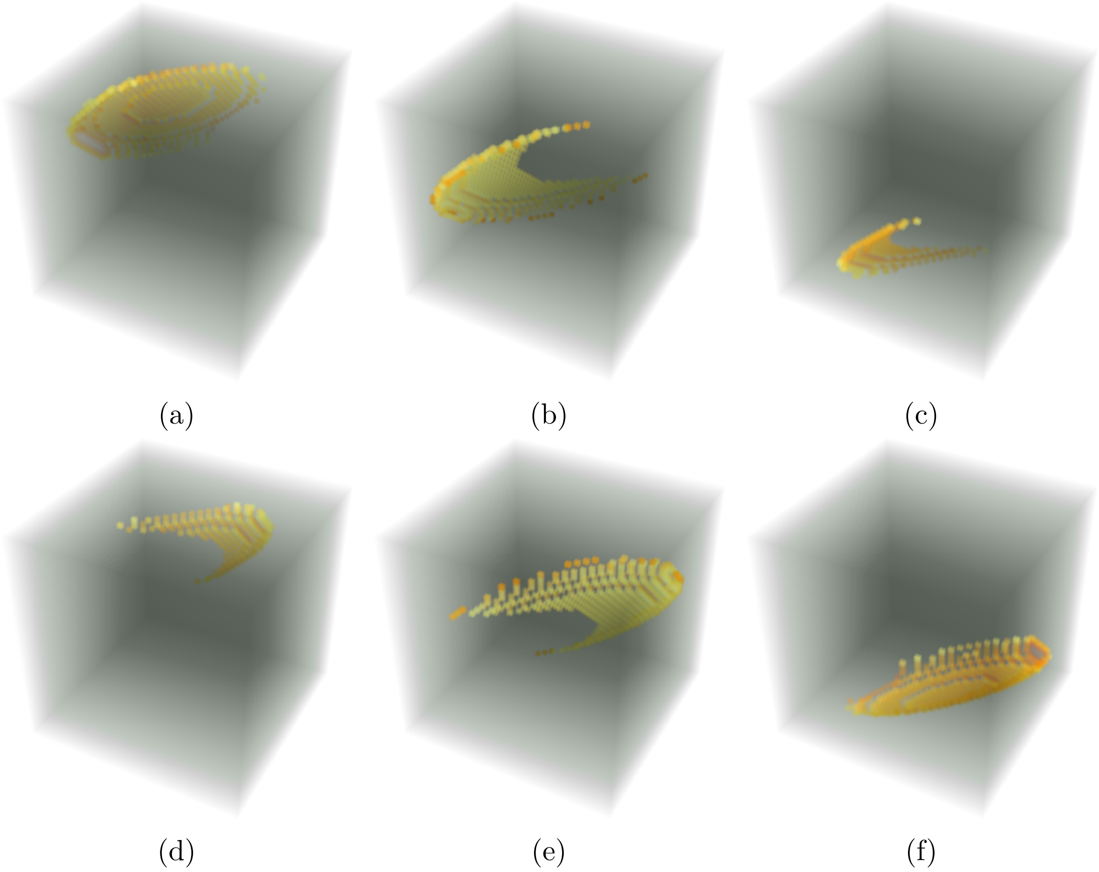
4.3.5 Visibility-Weighted Saliency (VWS) Histograms

As mentioned in Section 4.3.3, the saliency field can be computed from different appearance attributes. Multiple saliency fields computed from different appearance attributes can be combined together in order to represent different aspects of the visual saliency of voxels. In our implementation, we use brightness and saturation respectively to compute visibility-weighted saliency fields and define the weighted sum of the two sets of feature saliency as visibility-weighted saliency.

$$W_F = u_1 W_F(O_b, i, \sigma) + u_2 W_F(O_s, i, \sigma) \tag{4.13}$$

where u_1 and u_2 are weights of different appearance attributes. u_1 and u_2 are both in the interval $[0, 1]$ and $u_1 + u_2 = 1$. $W_F(O_b, i, \sigma)$ is the visibility-weighted saliency of feature F computed using brightness of voxels and similarly $W_F(O_s, i, \sigma)$ is the visibility-weighted saliency of feature F from saturation of voxels.

Figure 4.5 and Figure 4.6 display bar charts of our visibility-weighted saliency of the three features and the feature visibility by Wang et al. [6] for comparison. We compute the saliency fields using brightness and saturation respectively and thus acquire two sets of feature saliency of the three features in the synthetic data set. In Figure 4.5 and Figure 4.6, the feature saliency based on the brightness component shows similar patterns as the feature visibility. However, the feature saliency based on the saturation component gives the highest score to the middle disk (magenta color), which indicates the middle disk is significantly more salient than the other two (light green and dark green) in terms of saturation. The visibility-weighted saliency combines the feature

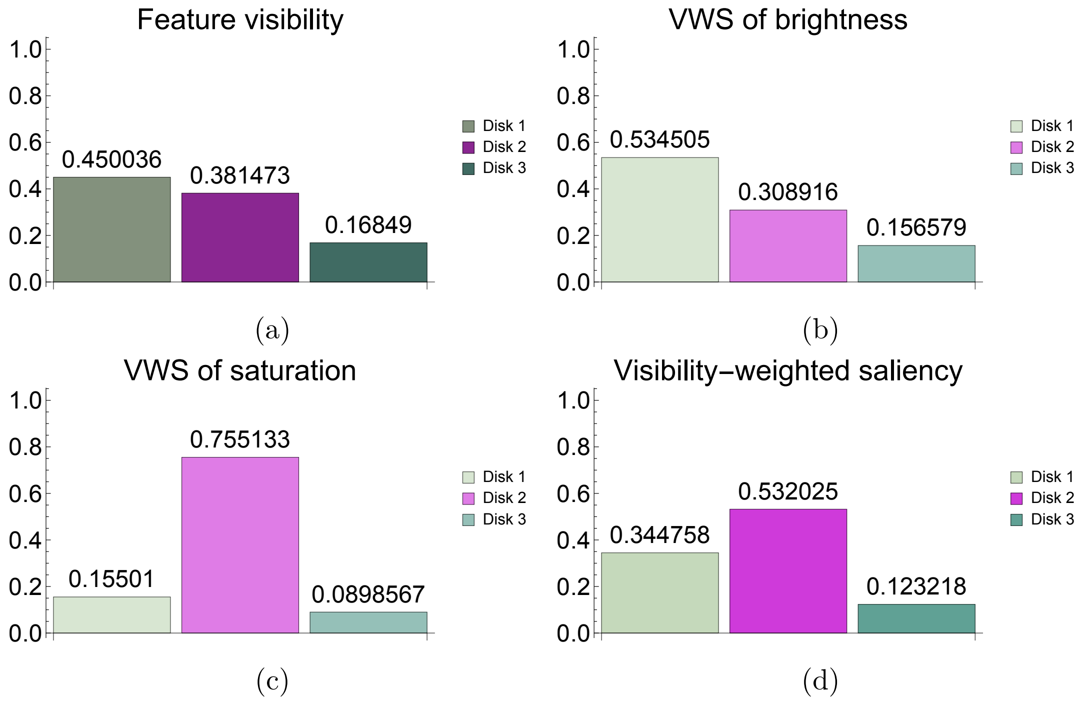


(a) (b) (c)

(d) (e) (f)

Figure 4.4: Visibility-weighted saliency fields of the three disks. The first column ((a) and (d)) shows the saliency fields of the top disk in the two viewpoints in Figure 4.1. The second column ((b) and (e)) and the third column ((c) and (f)) show the saliency fields for the middle disk and the bottom disk in the two viewpoints respectively.

saliency from brightness and saturation with user-specified weights. The visibility-weighted saliency can be used as a measure to indicate the saliency of features in volume rendered images. Equal weights of appearance attributes are assumed for now in our implementation, which is an equivalent assumption to that made in 2D saliency [16]. However, more detailed perceptual studies may determine more ideal weights.



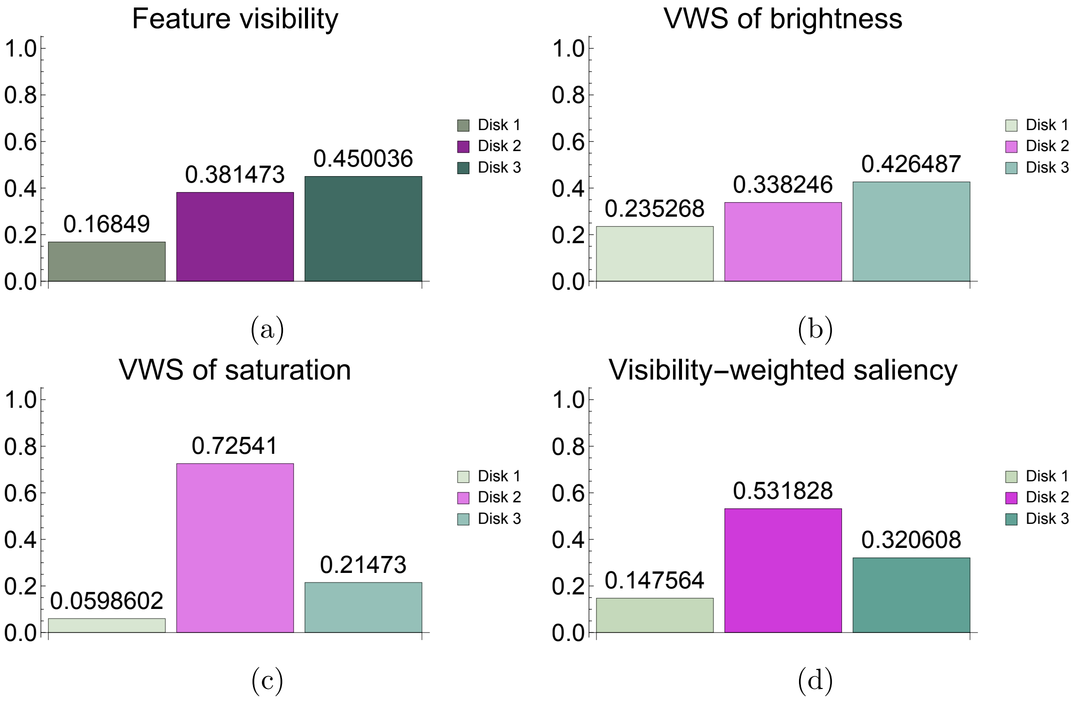
(a)

(b)

(c)

(d)

Figure 4.5: (a) The feature visibility [6] histogram shows the sum of visibility values of all the voxels belonging to each feature. (b) & (c) The two histograms of visibility-weighted saliency based on brightness and saturation respectively show the sum of visibility-weighted saliency of all the voxels belonging to each feature ($W_F(O, i, \sigma)$ in Section 4.3.4). (d) The histogram of visibility-weighted saliency shows the visibility-weighted saliency with equal weights of brightness and saturation (W_F in Section 4.3.5). Feature visibility (a) and visibility-weighted saliency from brightness (b) both suggest that the top disk is the most visible and the bottom disk is the least visible. However, the middle disk with magenta color is significantly more salient than the other two disks (light green and dark green) in terms of saturation (c).



(a)

(b)

(c)

(d)

Figure 4.6: Similar to Figure 4.5, in the viewpoint on the right in Figure 4.1, the bottom disk is the most visible according to feature visibility (a) and most salient according to feature saliency from brightness (b). However, feature saliency from saturation (c) suggests that the middle disk (magenta color) is significantly more salient than the other two (light green and dark green) in terms of saturation. The histogram (d) is the visibility-weighted saliency with equal weights of brightness and saturation.

4.4 Use Case: Measuring Feature Saliency Resulting from Different Transfer Functions

In this section, we present results of using our approach to measure visual saliency of features of a volume data set with two different transfer functions.

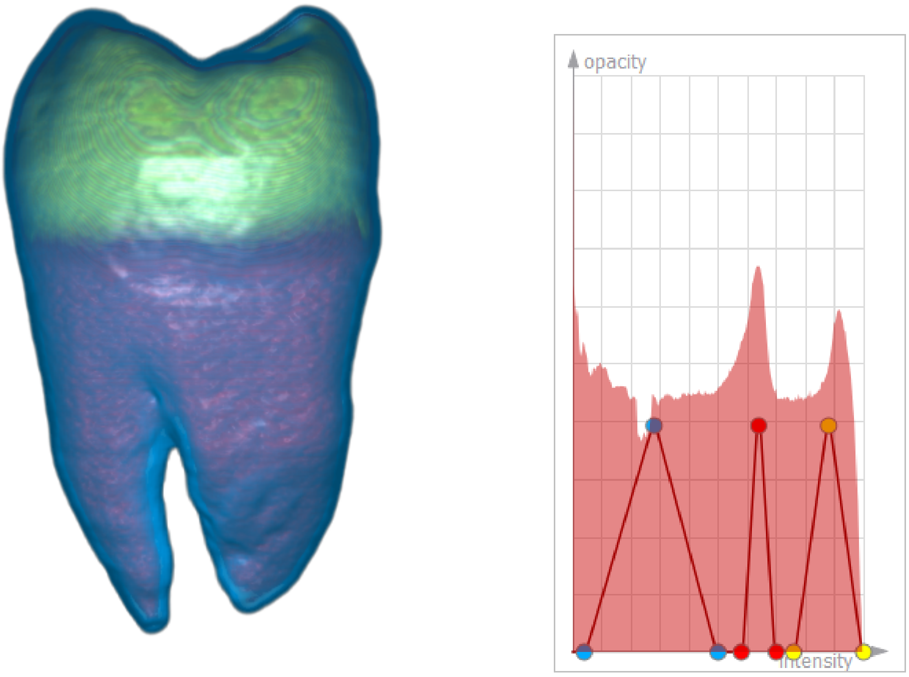
A tooth data set [3] is rendered with two different transfer functions to demonstrate the effectiveness of our approach. The first transfer function (Figure 4.7) assigns equal opacity to the three features. The second transfer function (Figure 4.10) is designed to emphasize the enamel (the yellow material), thus it assigns high opacity the enamel and low opacity to the other two features (cementum & pulp chamber and dentine).

By observation, it is clear that the transfer function in Figure 4.10 is better in terms of visualizing the enamel than Figure 4.7. The purpose of our approach is to provide an automated objective measure to make this comparison. This is demonstrated through the output of the visibility-weighted saliency fields of the two transfer functions (Figure 4.8 and Figure 4.11).

In the visibility-weighted saliency fields of the first transfer function (Figure 4.8), all three features are reasonably salient. On the other hand, the visibility-weighted saliency fields of the second transfer function (Figure 4.11) suggest the enamel has significantly higher visual saliency in the volume rendered image. The feature visibility and visibility-weighted saliency in Figure 4.9 and Figure 4.12 summarize the visibility and visual saliency of the three features specified by the transfer functions.

4.5 Experiment

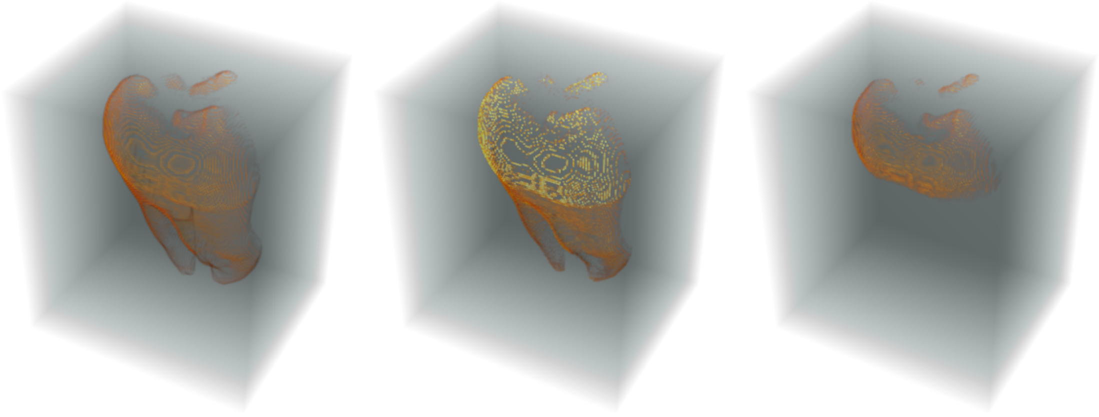
An experiment was performed to investigate how the visual saliency of objects in volume visualization was perceived by human users. A typical objective in volume visualization is to provide a clear impression of a certain part (i.e. a feature) of a volume data set while showing the rest of the data set for reference [178]. For this reason, it is necessary to test how well the shapes of features are perceived by human users. The aim of this experiment is to gather subjective opinion scores regarding how clear and distinct the features are in the images shown to the participants, and then compare the user opinion scores against the proposed computational visual saliency metric.



(a)

(b)

Figure 4.7: A tooth data set (a) with a transfer function (b) revealing three features: cementum & pulp chamber (blue), dentine (red) and enamel (yellow). Equal opacity is assigned to the three features in the transfer function.

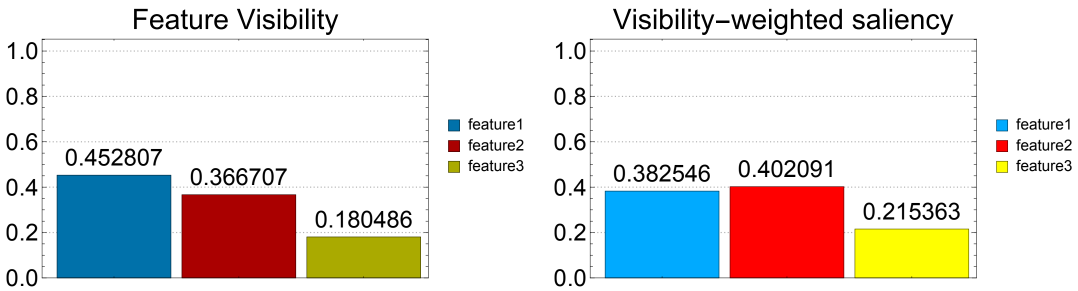


(a)

(b)

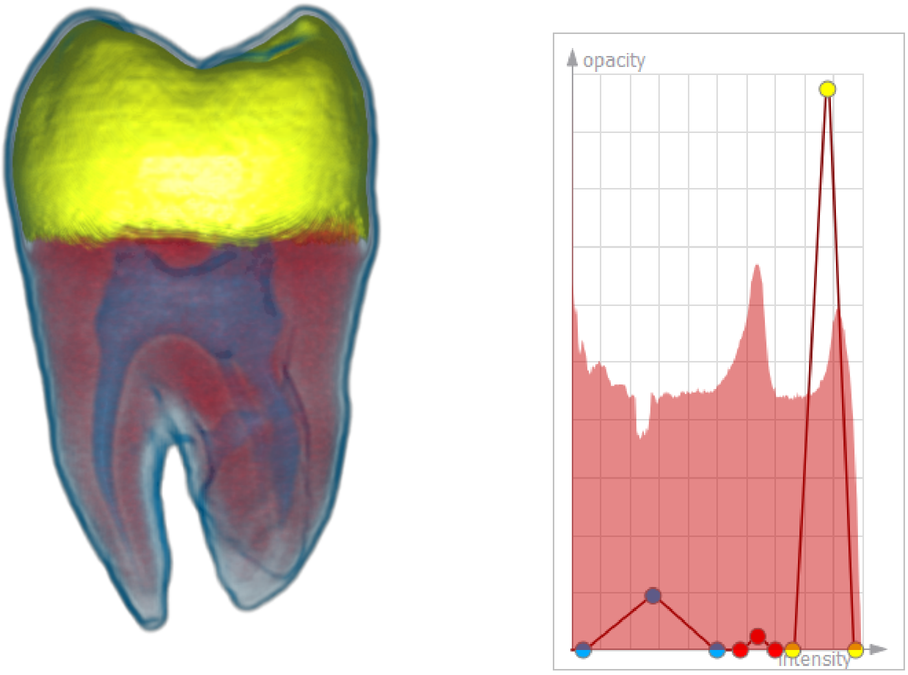
(c)

Figure 4.8: Visibility-weighted saliency fields of the three features, computed with the transfer function in Figure 4.7. From left to right, the features are cementum & pulp chamber (a), dentine (b) and enamel (c).



(a) (b)

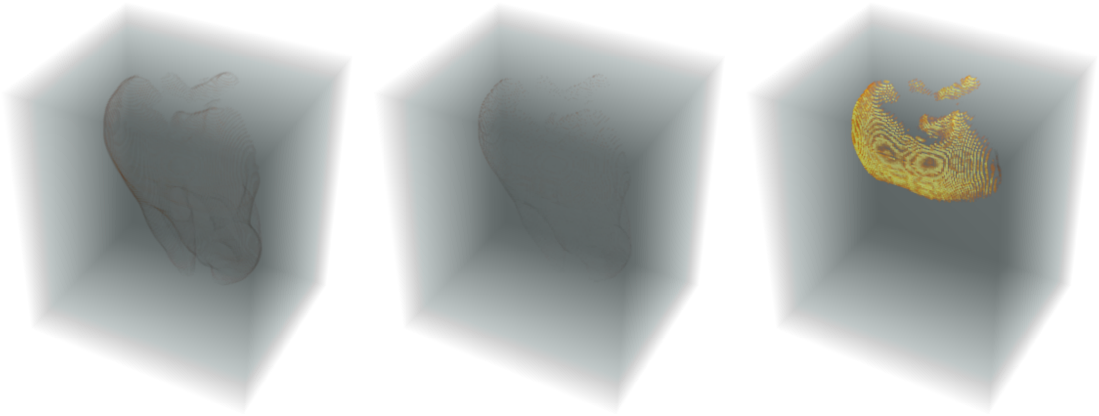
Figure 4.9: Feature visibility [6] (a) and visibility-weighted saliency (b) of the three features, computed with the transfer function in Figure 4.7.



(b)

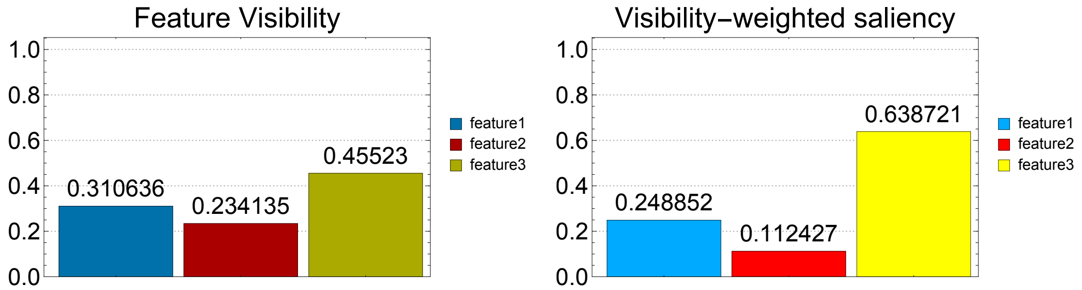
(a)

Figure 4.10: A tooth data set with a transfer function particularly highlighting the enamel (the yellow feature)



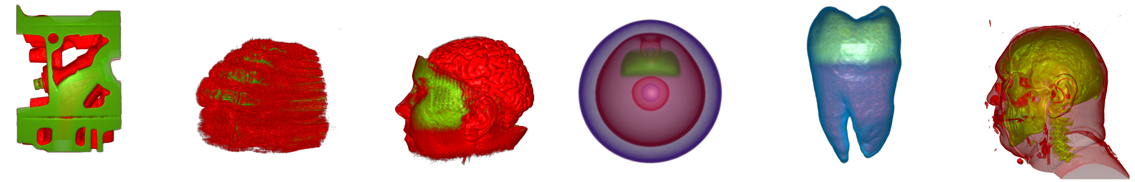
(a) (b) (c)

Figure 4.11: Visibility-weighted saliency field of the three features, computed with the transfer function in Figure 4.10. From left to right, the features are cementum & pulp chamber (a), dentine (b) and enamel (c).



(a) (b)

Figure 4.12: Feature visibility (a) and visibility-weighted saliency (b) of the three features, computed with the transfer function in Figure 4.10.



(a) engine block (b) foot (c) MRbrain (d) nucleon (e) tooth (f) VisMale

Figure 4.13: Volume data sets used in the experiment

4.5.1 Source Images and Participants

The images used for the study were rendered by Voreen [29] using a variety of transfer functions of tent-like shapes, e.g. the transfer function in Figure 5.1 (b), highlighting different features. Eight publicly available volume data sets were used to produce images for the experiment. Two of these data sets (CThead [1] and lobster [1]) were only used in a brief training session at the beginning of the experiment. The remaining six volume data sets comprised the following: nucleon [2], engine block [1], foot [1], tooth [1], VisMale [1] and MRbrain [3] as shown in Figure 4.13. For these, we empirically modulated the transfer function to achieve varying degrees of emphasis on individual features. It should be noted that relatively simple transfer functions were used in order to simplify the explanation of the task to participants; for instance, in order to associate a feature with an easily identifiable color.

30 participants (20 male and 10 female) took part in the experiment. All participants were aged between 22 and 39 years old and consisted of postgraduates, undergraduates and researchers. Over half of the participants had a computer science background. One of the male participants reported that he had slight red-green color blindness and his results were excluded from the study; thus the scores of 29 participants were used in the final data analysis.

[1] Volume Library: http://www9.informatik.uni-erlangen.de/External/vollib/

[2] Voreen data sets: http://www.uni-muenster.de/Voreen/

[3] Stanford volume data archive: https://graphics.stanford.edu/data/voldata/

4.5.2 Methods and Measurements

Participants were asked to sit in front of a computer display viewing the images generated from volume visualization. The participants' task was to score the images on a scale of 1 to 5 by keyboard input, based on how conspicuous a particular feature appeared. In each trial, a screen identifying the feature of interest using a textual task description e.g. *"How clear and distinct is the RED feature?"* was first shown to the participant for 3 seconds, followed by a screen displaying a volume rendered image for 17 seconds (See Appendix B for a sample questionnaire of the experiment). The duration of the full experiment was approximately 20 minutes.

Some images were shown more than once in the experiment in order to detect whether the participants's scores were consistent during the experiment. We analyzed the sum of differences of the repeated images's scores and noticed that the scores by the participant with slight red-green color blindness were significantly less consistent than other participants. Hence, the scores by this participant were excluded and the other 29 participants's scores were used in the data analysis.

Our intention was to validate whether the VWS is able to predict how users perceive relative conspicuity of individual features. If VWS is valid, the ordering of features by user scores should correspond to ordering with VWS. The efficacy of the metric was compared to two other approaches from existing literature:

4.5.3 2D Feature Saliency (2DFS)

A widely used saliency model is the saliency map by Itti et al. [16], based on properties of early primate vision. Three visual attributes, i.e. intensity contrast, color opponency and orientation, are considered in the model in order to determine whether an image pixel stands out from its surroundings. The output from this model is a 2D view-dependent map indicating the visual saliency of pixels in the full rendered image and thus can not directly be used to estimate the visual saliency of 3D voxels. However, using an inverse distance weighting [189] between pixels of the final image and individual 2D feature images (rendered from the same view but each isolating only the individual features), we can estimate the visual saliency of each feature as the weighted total saliency of each 2D feature saliency map (see Appendix A for details). We refer to this as 2D Feature Saliency (2DFS) in the rest of the thesis.

4.5.4 Feature Visibility (FV)

Similar in objective to VWS, Wang et al. [6] proposed a scheme that measures the influence of individual features on the volume rendered image. Their approach allows the user to directly specify the desired visibility for the features of interest, and subsequently the opacity transfer function is optimized using an active set algorithm towards this target. We refer to their measure as Feature Visibility (FV) and use it as an additional basis to compare against VWS.

4.5.5 Data Analysis

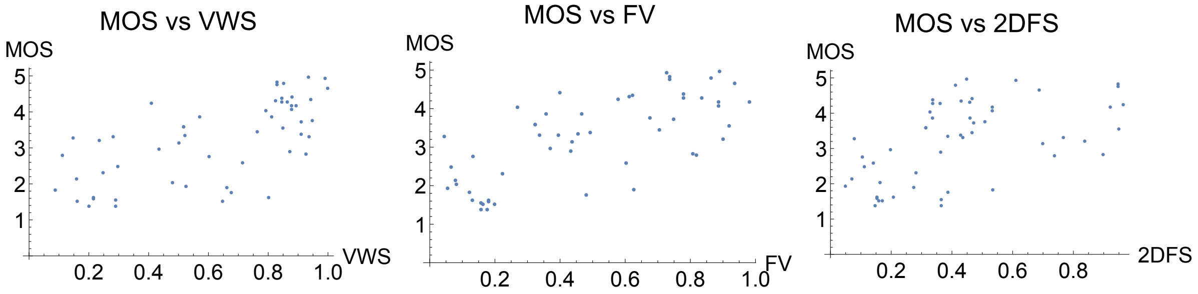
Because we are interested in the monotonic relationships between the variables, Spearman's rank correlation [190], which does not assume normality of the variables, would be suitable for evaluating the strength of monotonic associations of the observed scores (mean opinion scores) and scores calculated from computational metrics.

We applied the Spearman's rank correlation to analyze the monotonic association between the mean opinion scores (MOS) and three computational metrics, i.e. our visibility-weighted saliency (VWS) metric, feature visibility (FV) [6] and 2D feature saliency (2DFS) discussed in Section 4.5.3.

As shown in Figure 4.14, there are strong positive correlations between MOS and VWS (0.67508), and MOS and FV (0.678626) respectively. On the other hand, there is a moderate positive correlation between MOS and 2DFS (0.550472). Both VWS and FS are more monotonically correlated to MOS than 2DFS, and FV is slightly better than VWS by about 0.0036.

The p-values indicate statistical significance of the associations. In our results, the p-values for all the three cases are very small (below 2×10^{-5}). Therefore, we can reject the null hypothesis (there is no association between each pair of variables) and conclude that there are associations between MOS and VWS, FS and 2DFS respectively.

In addition, we group MOS by features to provide a more detailed view of the correlation between MOS and VWS across individual data sets. Figure 4.15 displays line plots of MOS versus VWS for each feature of the data sets separately, with the data points sorted by VWS on the x axis. Because some images were shown more than once in the experiment, the resulting data points have the same VWS and very similar MOS in the line plots. Although the line plots are not strictly monotonic (i.e. MOS



(a) Spearman's $\rho = 0.67508$, P-value$= 2.16005 \times 10^{-8}$

(b) Spearman's $\rho = 0.678626$, P-value$= 1.70738 \times 10^{-8}$

(c) Spearman's $\rho = 0.550472$, P-value$= 1.61418 \times 10^{-5}$

Figure 4.14: Spearman's rank correlation of 54 opinion scores against the corresponding visibility-weighted saliency (VWS), feature visibility (FV) and 2D feature saliency (2DFS) respectively. There are strong positive correlations between MOS and VWS (a) and between MOS and FV (b). There is a moderate positive correlation between MOS and 2DFS (c).

and VWS are not strictly monotonically correlated), we notice that they have loose monotonic associations.

Figure 4.16 and Figure 4.17 display two features of the VisMale data set represented by the data points in Figure 4.15 (g) and (h) respectively. In Figure 4.16 and Figure 4.17, (a), (c), (e) and (g) are the volume rendered images and (b), (d), (f) and (g) are the corresponding visibility-weighted saliency histograms respectively. The curve in Figure 4.16 is strictly monotonic, which indicates that VWS and MOS are well correlated for Feature 2 (the green feature) of the VisMale data set. Figure 4.17 shows a case where VWS and MOS are not completely correlated. Figure 4.17 (e) and (g) are represented by the third and the fourth (from left to right) data points respectively in Figure 4.15 (h). For Feature 1 (the red feature) of the VisMale data set, Figure 4.17 (g) received a higher score than (e) from VWS, but MOS gave the opposite ranking of the two volume rendered images.

The analysis of the experiment results indicates that our VWS is better than 2DFS and equivalent to FV in terms of predicting human perception of visual saliency of features in volume rendering images.

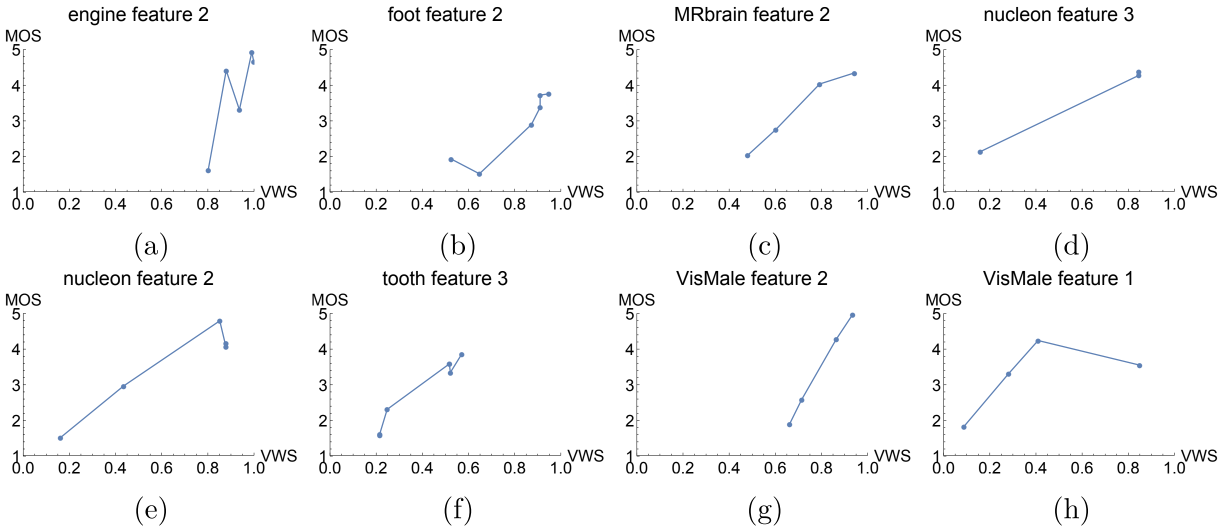


Figure 4.15: Line plots of MOS versus VWS for each feature of the data sets separately

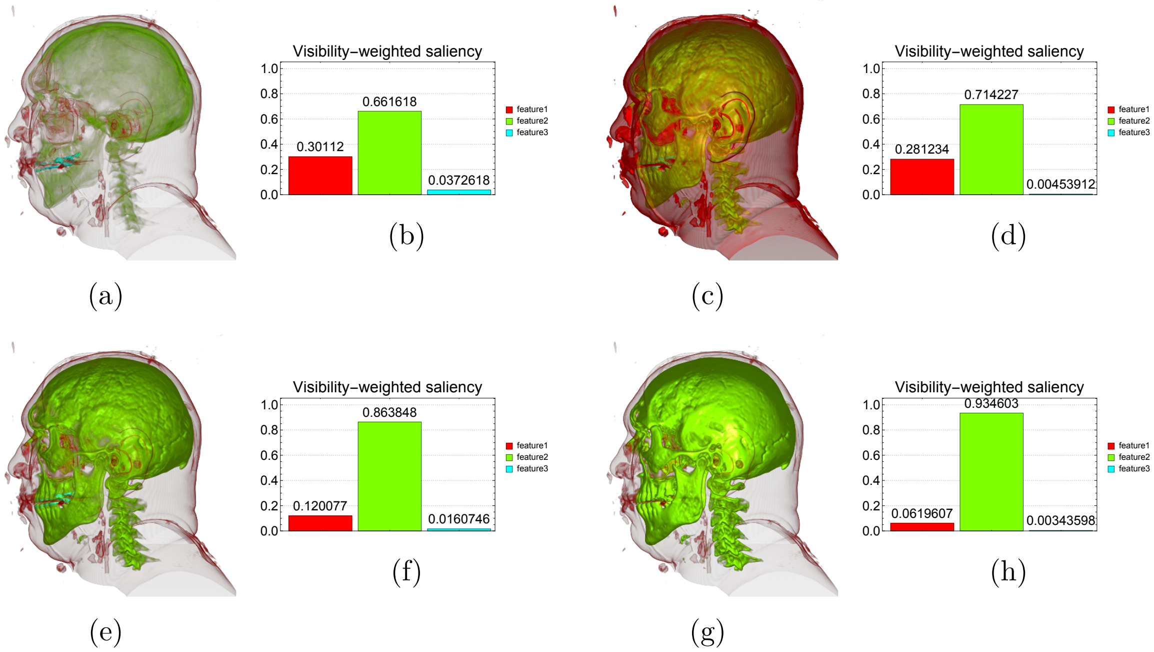


Figure 4.16: Visibility-weighted saliency values of the green feature are represented by the data points in Figure 4.15 (g)

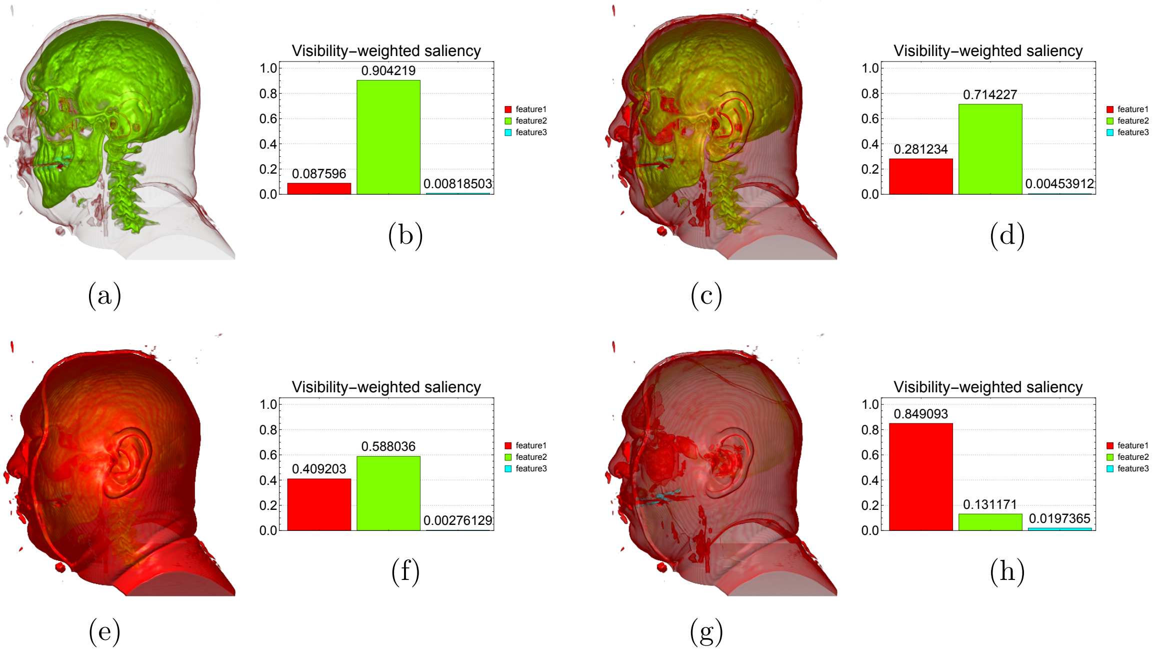


Figure 4.17: Visibility-weighted saliency values of the red feature are represented by the data points in Figure 4.15 (h)

4.5.6 Varying Saturation and Brightness in Transfer Functions

A major difference between our visibility-weighted saliency and the feature visibility is that our approach reflects two aspects of the resulting visualization, i.e. voxel visibility and visual saliency, while the feature visibility only reflects voxel visibility. Figure 4.18 displays an image of the tooth data set and its feature visibility and visibility-weighted saliency. In Figure 4.19, the saturation of the red feature is reduced. Our visibility-weighted saliency is aware of the saturation change, thus the red feature has lower VWS and the yellow feature has higher VWS in Figure 4.19 (c). On the other hand, the feature visibility is not aware of the saturation change, thus the feature visibility in Figure 4.19 (b) is the same as that in Figure 4.18 (b).

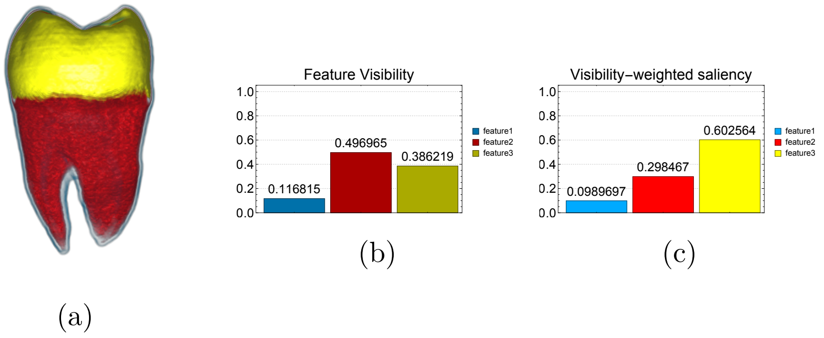


Figure 4.18: (a) The tooth data set; (b) Feature visibility; (c) Visibility-weighted saliency

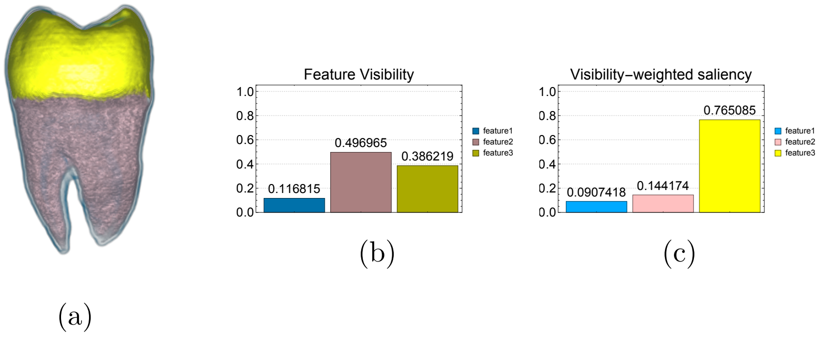


Figure 4.19: (a) The saturation of the red feature is reduced; (b) The feature visibility is the same as Figure 4.18 (b); (c) The visibility-weighted saliency of (a): the red feature has lower VWS and the yellow has higher VWS.

4.6 Conclusions

In this chapter, we propose visibility-weighted saliency as an improved measure of the visual saliency of features in volume rendered images, in order to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. According to the results of our experiment, the proposed visibility-weighted saliency metric is as effective as feature visibility and better than 2D feature saliency at predicting the visual saliency of features in volume data. Our visibility-weighted saliency also captures changes in saliency due to brightness/contrast, which is not taken into account by feature visibility. With visibility-weighted saliency, the perceptual importance of features rendered in different viewpoints with different transfer functions can be measured in a quantitative and fully automated way.

Chapter 5

Transfer Function Optimization Using Visibility-Weighted Saliency

In this chapter, we present a transfer function optimization approach using the visibility-weighted saliency metric discussed in Chapter 4. This is an automated approach that adjusts transfer functions to match the visibility-weighted saliency towards user-specified targets. In addition, a parallel line search strategy is presented for exploiting the computing power of multi-core processors to improve the performance of the transfer function optimization approach.

5.1 Introduction

Volume visualization is an effective means of discovering meaningful features in volume data sets. Both exterior and interior of structures can be revealed simultaneously in a semi-transparent manner by specifying opacity values for the features in transfer functions [6]. Features could include intensity intervals in 1D transfer functions, rectangular or other shapes in 2D or higher-dimensional transfer functions.

In the specification of transfer functions for volume visualization, users often have a rough idea of how clear and opaque each feature should be and then adjust the opacity value of the features accordingly. However, the relationship between the opacity of features and the saliency of the features in the final image is not linear. The saliency of a feature in the final image depends on the opacity value assigned to the feature as well as the neighborhood of the feature and view-dependent occlusion of the feature.

Therefore, it is desirable to have an automated method to assist the user in the design of transfer functions that match target saliency levels specified by the user. In this chapter, we propose an optimization approach that supports this requirement by automatically refining a user-defined transfer function towards any given saliency distribution. Moreover, we present a parallel line search strategy to improve the performance of the transfer function optimization.

5.2 Related Work

Transfer function specification is a non-trivial and unintuitive task in volume visualization. Compared to typical transfer function approaches, which are often subjective, it is desirable to have objective feedback regarding the clarity of features in volume visualization.

Correa and Ma [5] introduced visibility histograms to guide transfer function design for both manual and automatic adjustment. Visibility histograms (Figure 2.6), which summarize the distribution of visibility of voxels from a given viewpoint, are a powerful feedback mechanism for volume visualization [13]. Wang et al. [6] extended visibility histograms to feature visibility histograms, in order to measures the influence of each feature to the resulting images. They described a scheme that allows users to specify a desired visibility for features of interest and subsequently the opacity transfer function is optimized using an active set algorithm [99].

Researchers have developed a variety of parallel strategies to accelerate sequential optimization algorithms [191]. Phua et al. [192] proposed a parallel extension to quasi-Newton methods [193]. Their approach generates several search directions at each iteration and then applies different line search and scaling strategies in parallel along each search direction. Peachey et al. [194] presented another approach to parallelize the quasi-Newton methods. In their applications, the objective function evaluation typically requires minutes or hours of processing time. Therefore, they introduced an approach that evaluates the objective function in parallel over a cluster of computers and continues to the next iteration before all evaluations finish in order to accelerate convergence.

5.3 Method

In Chapter 4, visibility-weighted saliency was proposed as a measure of visual saliency of features in volume visualization. This metric indicates the perceptual importance of voxels and the visibility of features in volume rendered images and can be utilized to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. In this chapter, we describe a transfer function optimization approach based on the visibility-weighted saliency metric in order to automatically adjust the volume visualization to satisfy user-specified targets set on the visibility-weighted saliency for the features.

The approach described in Chapter 3 is an automated method of optimizing transfer functions, based on the intensity distribution of voxels in the volume data set. However, this approach does not take into account the spatial distribution of voxels and the viewpoint of the visualization. Visibility-weighted saliency, on the other hand, takes into account both of these two aspects. The visibility-weighted saliency consists of two component fields, i.e. saliency field and visibility fields. Saliency fields are essentially difference of Gaussians, which include the information of local neighborhoods of voxels. Visibility fields are computed from opacity contribution of voxels to volume rendered images, which indicate viewpoint dependent occlusions of the voxels.

Constraints are introduced in the search of the parameter space. Only the opacity of features are changed in the transfer function domain. The definition of features (e.g. intensity ranges on 1D transfer functions) and the colors of features remain the same. These constraints are based on the assumption that the user has explored the volume data and done the classification of features. Our approach aims to help the user adjust the saliency distribution and reduce occlusion while preserving the user's knowledge or judgments of the data set. While the approach discussed in Chapter 3 was designed for exploration and visual search of volume data, the approach in this chapter could aid in analysis, understanding or closer inspection of the data.

5.3.1 Objective Function

Users define target importance values for each feature in the transfer function domain. Our transfer function optimizer adjusts the transfer function to match the visibility-weighted saliency with the user-defined target saliency values. Multiple saliency fields computed from different appearance attributes can be combined together in order to represent different aspects of the visual saliency of voxels. In our implementation, brightness and saturation are used respectively to compute visibility-weighted saliency fields and define the weighted sum of the two sets of feature saliency as visibility-weighted feature saliency. The objective function F is defined as the root mean square of the differences of the visibility-weighted saliency and target importance of each feature.

$$F = \sqrt{\frac{\sum_{i=1}^{n}(W_i - t_i)^2}{n}} \tag{5.1}$$

where $W_F = u_1 W_F(O_b, i, \sigma) + u_2 W_F(O_s, i, \sigma)$ is the visibility-weighted saliency of feature i, and t_i is the user-defined importance of feature i. These user-defined saliency values are normalized and they add up to 1, in other words, $t_i \in [0.1]$ and $\sum_{i=1}^{n} t_i = 1$.

As previously described in Section 4.3.5, multiple saliency fields computed from different appearance attributes can be combined together in order to represent different aspects of the visual saliency of voxels. In our implementation, $W_F = u_1 W_F(O_b, i, \sigma) + u_2 W_F(O_s, i, \sigma)$ is a weighted sum of visibility-weighted saliency values computed using brightness and saturation of voxels respectively, and u_1 and u_2 are weights of the two appearance attributes.

However, the visibility-weighted saliency W_i is not a variable that can be directly modified. Instead, W_i is a complicated function of the color and opacity of voxels in feature i and is also influenced by the viewpoint of rendering. A visibility-weighted saliency field is a combination of a visibility field and a saliency field. The saliency field is a view-independent field based on the color of every voxel in the volume data set, while the visibility field is a view-dependent field computed from the opacity contribution of every voxel to the final image when rendered from a certain viewpoint.

The computation of visibility fields is non-trivial. In order to compute a visibility field, a slice-based rendering is performed on a series of quads which are parallel to the viewing plane, one for each slice. Subsequently, the visibility values are computed by subtracting the accumulated opacity of the previous slice from that of the current slice. After collecting the visibility values of all voxels, the visibility field can be constructed. The details of visibility fields were previously described in Section 4.3.2.

The evaluation of the objective function is computationally expensive. However, in an iterative optimization, the visibility field and visibility-weighted saliency have to be recomputed at each step after the feature opacity values are updated.

5.3.2 Parameter Space

We use a nucleon data set [195] to demonstrate how the visibility-weighted saliency of features change when the feature opacity values change. As displayed in Figure 5.1, three features are defined in the transfer function for the nucleon data set.

The dimension of the parameter space is the same as the number of features defined by the user. In this case, three features are defined for the nucleon data set. The opacity of each feature is mapped to an axis in the parameter space. Therefore, the opacity values of the 3 features are mapped to x, y, z axes of a 3D scalar field respectively. Figure 5.2 displays a visualization of three 3D scalar fields representing this parameter space. One field is presented for each feature to provide an intuitive overview of the relationship between the feature opacity values and the visibility-weighted saliency values. In order to avoid confusion, please note that this has no spatial relationship to the actual 3D volume data (which is itself a 3D scalar field).

Feature 1 (the purple structure in Figure 5.1) is the exterior of the nucleon data set, the visibility-weighted saliency of this feature is shown in the 3D scalar fields in the same color at the left in Figure 5.2. The visibility-weighted saliency of Feature 1 increases as its opacity increases, as demonstrated by the fact that both the brightness and opacity of the corresponding 3D field increases along the x-axis. Similar patterns also appear in the other 2 scalar fields, the visibility-weighted saliency of Feature 2 (the red structure in Figure 5.1) and Feature 3 (the green structure in Figure 5.1) also increase as their opacity values increase.

Moreover, Feature 1 is the exterior of the nucleon, its visibility-weighted saliency is almost not influenced by the opacity of other features. On the other hand, the visibility-weighted saliency of Feature 2 is influenced by both the opacity of Feature 1 and Feature 2. In addition, the visibility-weighted saliency of Feature 3 is drastically influenced by the opacity of Feature 1, Feature 2 and Feature 3, as Feature 3 is an interior structure and can be easily occluded by the other 2 features.

In order to demonstrate the distribution of the objective function in the parameter space $x \in [0, 1]$, $y \in [0, 1]$ and $z \in [0, 1]$, we sample the parameter space with sampling

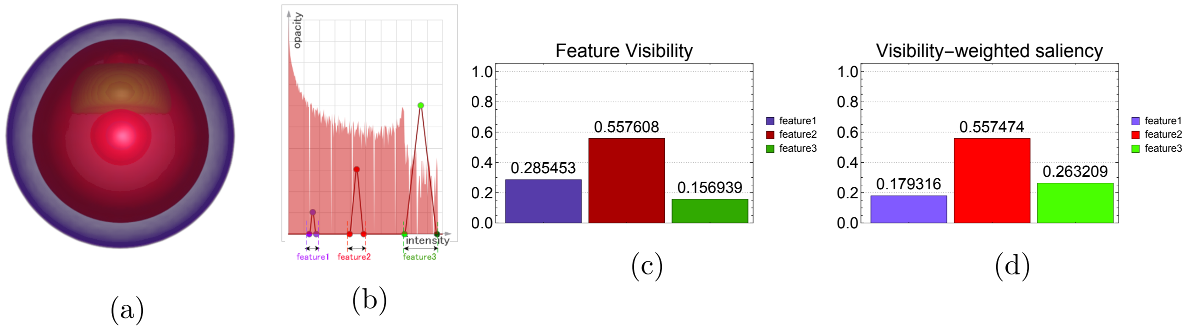


Figure 5.1: (a) A nucleon data set [195]; (b) A user-specified transfer function with 3 features (c) The feature visibility histogram [6]; (d) The visibility-weighted saliency histogram

interval 0.1, from 0 to 1 along each axis. There are 11 sampling points along each axis, which results in 1331 sampling points in the parameter space. In Figure 5.3, the parameter space is rendered as a density plot with a color function resembling a temperate map which gradually changes from orange to blue.

5.3.3 Optimization Algorithm

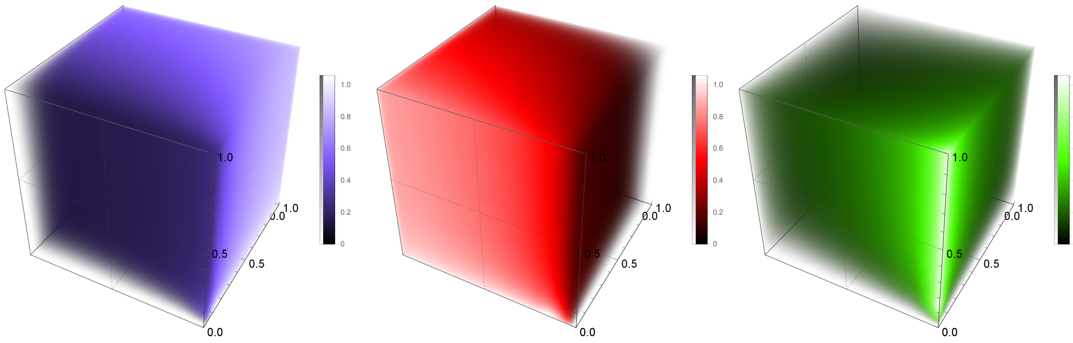
The gradient descent algorithm is employed in our transfer function optimizer. Gradient descent is a first-order optimization algorithm. It is based on the observation that if a function $f(x)$ is defined and differentiable in a neighborhood of a point x_1, then $f(x)$ decreases fastest in the direction of the negative gradient of the function [196].

Given a continuously differentiable function $f(x)$ with $x \in \mathbb{R}^n$, let x_k be the current iteration point and $g_k = g(x_k) = \nabla f(x_k)$ be the gradient of $f(x)$ at x_k. The gradient descent method defines the next iteration point by

$$x_{k+1} = x_k - \alpha_k g_k, k \geq 0$$

for α_k small enough, then $f(x_{k+1}) \leq f(x_k)$. The gradient varies as the iteration proceeds, tending to zero as it approaches a local minimum. When the gradient decreases, the iteration step sizes also decrease. So hopefully the sequence x_k converges to the desired local minimum after performing the iteration.

In gradient descent methods, we can either take very small step sizes and reevaluate



(a) VWS of feature 1 (b) VWS of feature 2 (c) VWS of feature 3

Figure 5.2: Visibility-weighted saliency of the 3 features are mapped to brightness and opacity of the 3D fields in (a), (b) and (c) respectively. The visibility-weighted saliency of Feature 2 (red) is affected by both the opacity of Feature 1 (purple) and Feature 2. The visibility-weighted saliency of Feature 3 (green) is affected by the opacity of Feature 1, Feature 2 and Feature 3.

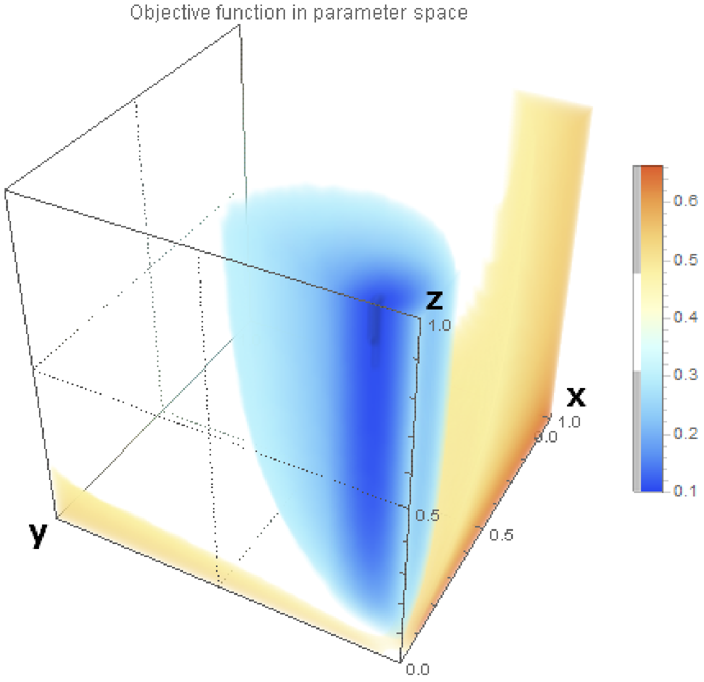


Figure 5.3: Each position (x, y, z) in the parameter space represents 3 features with opacity values (x, y, z). The value of the objective function (with {0.1, 0.3, 0.6} as target) is mapped to the color in the parameter space (with sampling interval 0.1). For clarity, only the high and low values are visible and the data range in the middle is set to transparent.

the gradient at every step, or take large steps each time. If the step size is too small, it may end up in a laborious situation that the objective function converges very slowly. If the step size is too large, it results in a more zigzag path and may have the risk of missing the local minimum and thus cannot converge.

5.3.4 Estimating Descent Directions

In each step of the optimization, the descent direction g_k has to be updated. As previously discussed, the visibility-weighted saliency of a feature increases as its feature opacity increases. However, the relationship between the visibility-weighted saliency and the opacity of a feature also depends on the viewpoint of rendering and the spatial distribution of voxels of every feature in the volume data set. An exact derivative of the visibility-weighted saliency with respect to the opacity of the feature cannot be determined in advance.

In the following subsections, two methods for estimating descent directions are described.

Gradients with Backward Difference

The partial derivative of the objective function F with respect to x_i is

$$\frac{\partial F}{\partial x_i} = \frac{\partial F}{\partial W_i}\frac{\partial W_i}{\partial x_i}$$

where x_i is the opacity and W_i is the visibility-weighted saliency of feature i. The partial derivative $\frac{\partial F}{\partial W_i}$ can be solved from the objective function F. However, $\frac{\partial W_i}{\partial x_i}$ cannot be determined without knowledge of the actual volume data set.

For a function $f(x)$, its first-order derivative can be estimated by a backward difference divided by a small step.

$$\frac{\nabla_d[f](x)}{d} = \frac{f(x) - f(x-d)}{d}$$

where d is a nonzero number. When d is small, the backward difference divided by d approximates the derivative. Assuming that f is differentiable, the error in this approximation can be derived from Taylor's theorem.

$$\frac{\nabla_d[f](x)}{d} - f'(x) = \mathcal{O}(d) \to 0,\ as\ d \to 0$$

The backward difference is used here to approximate $\frac{\partial W_i}{\partial x_i}$, hence we have

$$\frac{\partial W_i}{\partial x_i} \approx \frac{\nabla_d[W_i](x_i)}{d}$$

The evaluation of the objective function F is very computationally expensive. In our implementation, a small step size d is adopted, therefore the backward difference can be calculated from values of the objective function, visibility-weighted saliency and steps of the previous iteration. In this case, no extra evaluation of the visibility-weighted saliency and the objective function is required.

Furthermore, if the function W_i of x_i is approximately a linear function, the partial derivative $\frac{\partial W_i}{\partial x_i}$ becomes constant and could be replaced by an empirical constant b_i. In this case, the gradient of the objective function with respect to the visibility-weighted saliency is used instead, which should be more computationally efficient.

$$\frac{\partial F}{\partial x_i} \approx \frac{\partial F}{\partial W_i} b_i$$

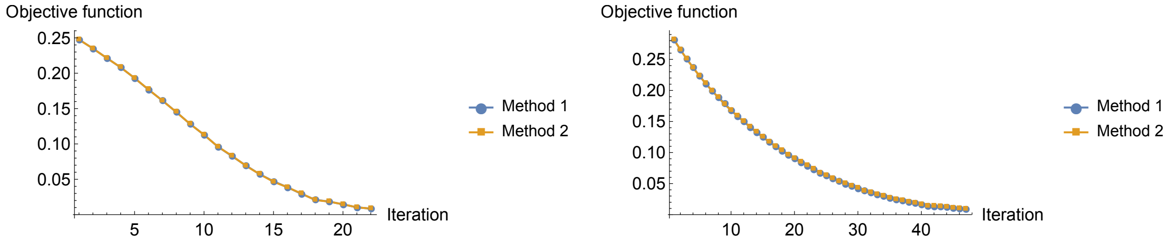
Descent Directions with Second-Order Derivatives

Newton's method, which is an iterative method for finding the roots of a differentiable function, can be used to find a minimum or maximum of a function. Because the derivative is zero at a minimum or maximum, minima and maxima can be found by applying Newton's method to the derivative.

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

This iteration equation gives a similar form of gradient descent, thus $\frac{\partial^2 F}{\partial x_i^2}$ can be adopted in our optimization algorithm as the descent direction.

The first-order derivative can be estimated by backward difference of the objective function F, and the second-order derivative F'' can be estimated by backward difference



(a) (b)

Figure 5.4: The two methods for estimating descent directions are applied to transfer function optimization on the nucleon data set (a) and a tooth data set (b) respectively. Note that the steps of the two methods overlap most of the time.

of the first-order derivative F'.

$$\frac{\partial^2 F}{\partial x_i^2} \approx \frac{\frac{\nabla_d[F](x_i)}{d}}{\frac{\nabla_d[F'](x_i)}{d}} = \frac{\nabla_d[F](x_i)}{\nabla_d[F'](x_i)}$$

Comparison of Descent Directions

We have tested the descent directions discussed above with several volume data sets with various step sizes. All the above described methods worked with small step sizes. As the step size increases, using $\frac{\partial F}{\partial W_i}\frac{\nabla_d[W_i](x_i)}{d}$ as descent directions would be unstable. While using $\frac{\partial F}{\partial W_i}b_i$ and $\frac{\nabla_d[F](x_i)}{\nabla_d[F'](x_i)}$ as descent directions would be still stable even when the step sizes are large. In our implementation, $b_i = 1$ is used and this yields desirable results.

Gradient descent was performed on several volume data sets with various transfer functions, using the two normalized descent directions computed from $[\frac{\partial F}{\partial W_1}b_1 ... \frac{\partial F}{\partial W_n}b_n]$ (Method 1) and $[\frac{\nabla_d[F](x_1)}{\nabla_d[F'](x_1)} ... \frac{\nabla_d[F](x_n)}{\nabla_d[F'](x_n)}]$ (Method 2). In the preliminary results, the convergence speeds of the two methods were very similar. Figure 5.4 shows that the steps made by the two methods overlap most of the time.

5.3.5 Line Search and Parallel Line Search

Performing gradient descent with a small step size may result in converging too slowly and require a lot of evaluations of the objective function, which is rather expensive to compute in our situation. Various approaches have been proposed regarding the

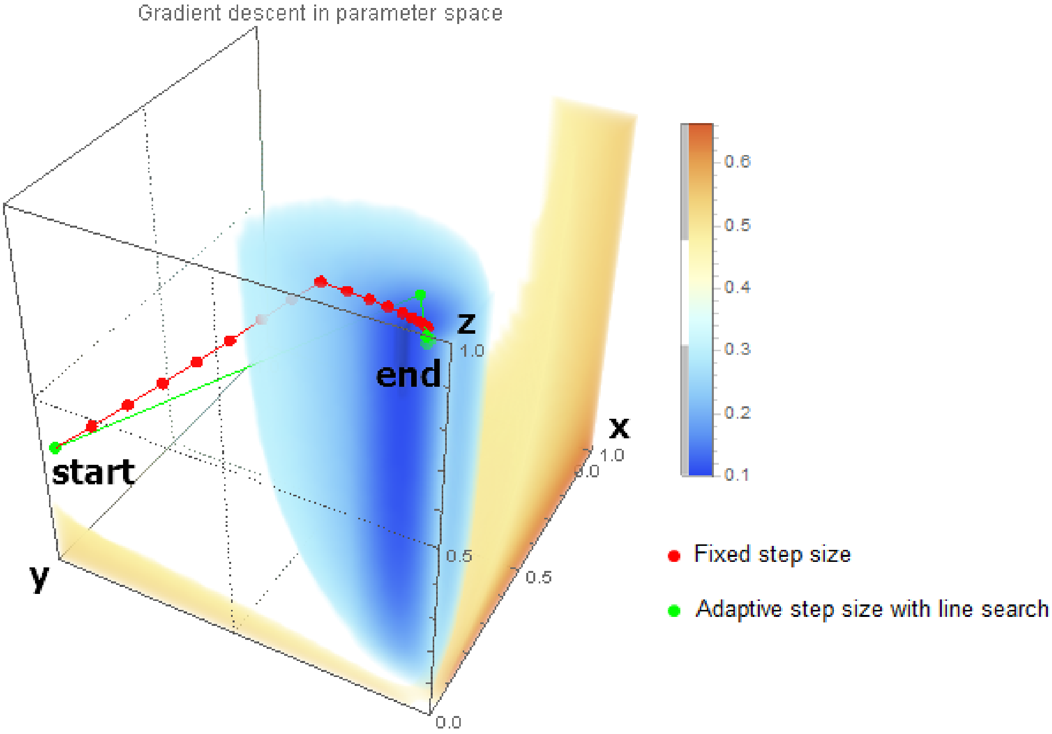


Figure 5.5: The steps of gradient descent methods with fixed step size and adaptive step size are shown in the parameter space in Figure 5.3 (the step size is 0.1)

choices of step sizes, which lead to various gradient algorithms [197].

Line Search

The line search strategy is an iterative approach that adapts the step size in gradient descent in order to achieve a reduction in the objective function while still making sufficiently fast progress.

$$h(\gamma) = f(x_k + \gamma g_k)$$

where g_k is the descent direction and x_k is the current point at the *k-th* iteration.

There are two type of approaches of line search, exact line search and inexact line search [198]. Exact line search chooses the next iteration point by achieving the least objective function value. However, despite the optimal properties, exact line search often behaves poorly and tends to zigzag in two orthogonal directions, which usually implies deteriorations in convergence [199]. In contrast, inexact line search only loosely finds a sufficient decrease of the objective function along the descent direction.

The inexact line search we used in our implementation is as follows.

1. Set initial iteration count $i = 0$ and set n to the maximum iteration count.

2. Check whether $f(x_k + \gamma_{i+1} g_k) < f(x_k + \gamma_i g_k)$ where $\gamma_i = 2^i$

3. If so and $i < n - 1$, $i = i + 1$ and repeat Line 2, otherwise terminate the line search.

The step size γ_n is chosen after the above line search procedure. This strategy does not find the exact minimum along the line direction, instead it yields reasonable results and descends much faster than using fixed step sizes.

With line search approaches, the optimization algorithm can converge much faster than using fixed step sizes. Figure 5.5 displays the paths of two gradient descent methods, one progresses regularly with fixed step sizes, the other progresses aggressively with line search. It takes fewer steps for the latter to reach a local minimum.

Parallel Line Search

The classical gradient descent is a sequential algorithm. In its iterative procedure, the next iteration takes the result from the previous iteration as input. However, the line search at each iteration can be computed in parallel to accelerate the optimization. This idea is particularly useful in our transfer function optimization, because the most expensive computation in our transfer function optimization is the evaluation of visibility-weighted saliency, which is required in the evaluation of the objective function at each iteration.

Thus, we propose a parallel line search strategy, which evaluates the objective function at different candidate points in parallel along the line search direction. Instead of sequentially searching for a desirable step size, parallel line search simultaneously evaluates the candidate step sizes and choose the best one for the current iteration. With this parallel approach, the computing power of modern multi-core processors can be better exploited to accelerate the transfer function optimization. Specifically, parallel line search launches multiple threads to perform the line search. Each thread computes the visibility-weighted saliency and the objective function at a candidate point. Subsequently, the results at all the candidate points are aggregated and the candidate point with the minimum objective function value is chosen as the next step.

The parallel line search is shown as follows.

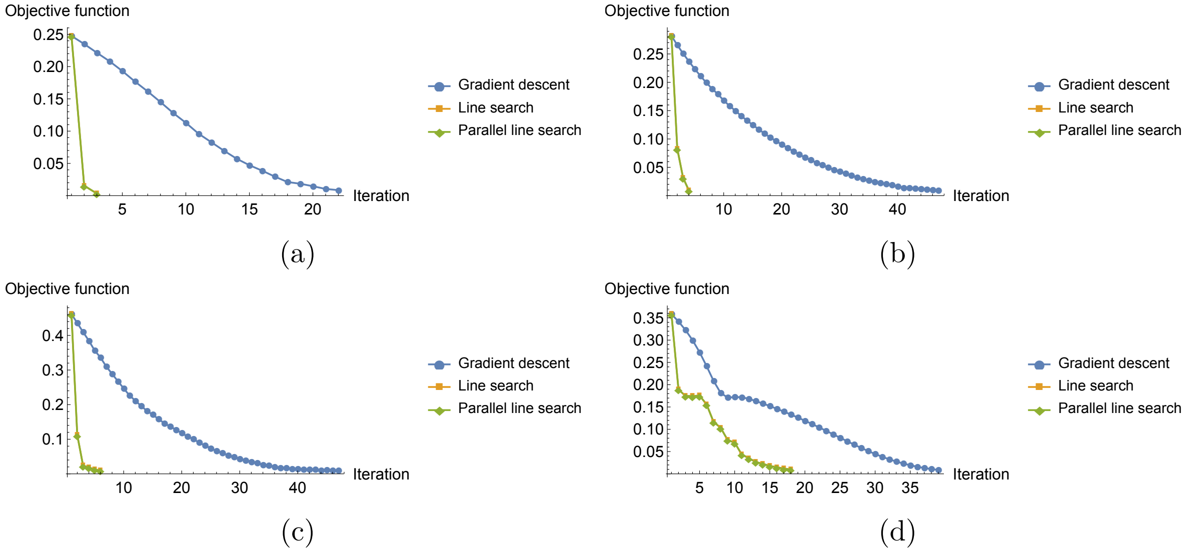
1. Generate a list of step sizes $S = \{\gamma_0, \gamma_1, ..., \gamma_{n-1}\}$ where $\gamma_i = 2^i$

2. Evaluate $f(x_k + \gamma_i g_k)$ in parallel for each γ_i in S

3. Find the index i of the minimum $f(x_k + \gamma_i g_k)$, then γ_i is the chosen step size.

The mechanism of the parallel line search is sightly different from the sequential line search. In the sequential line search, if the current candidate point does not meet the condition, the line search is terminated and the next candidate point would not be evaluated. By contrast, the parallel line search would always evaluate all the candidate points and pick the one with least value of the objective function. However, these two methods would have the same behavior if the objective function is a convex function.

The parallel line search strategy would introduce extra overhead of starting and terminating threads. Moreover, the number of threads should not exceed the number of cores of the processor, otherwise multiple threads have to share the same core and this would cause performance impact. The parallel line search is beneficial only when the evaluation of the objective function is more expensive than the parallel overhead. In our case, the evaluation of the objective function is very computationally expensive. It



(a)

(b)

(c)

(d)

Figure 5.6: The line search and parallel line search are applied to transfer function optimization on the nucleon data set (a), the tooth data set (b), the CT-Knee data set (c) and the first time step of the vortex data set (d) respectively. Note the curves of line search and parallel line search overlap most of the time, as they made the same step choices in most cases.

requires computing the visibility fields, which in turn requires that a pass of slice-based volume rendering is performed.

Figure 5.6 displays results of applying the two line search methods in optimizing transfer functions. The two curves of objective functions are mostly overlapping, which indicates the two methods acts almost the same in choosing step sizes.

5.4 Results and Discussions

In this section, we present some results to demonstrate the effectiveness of our approach on the nucleon data set (41 × 41 × 41), a tooth data set (140 × 120 × 161), a CT-knee (379 × 229 × 305) data set [3] and one time-step of a simulated turbulent vortex flow (128 × 128 × 128, 100 time-steps) [186]. Results were obtained in our experimental programs written in Wolfram Mathematica 11 on a computer equipped with an Intel Xeon E3-1246 v3 processor, 16GB of RAM and a NVIDIA Quadro K4200 graphics card.

In the results, we examine the transfer functions and volume rendered images before and after transfer function optimization, as well as the evolution of visibility-weighted saliency and opacity values of features due to the optimization. The feature visibility histograms [6] are displayed along with the visibility-weighted saliency histograms for comparison.

The two optimization methods shown in the results are gradient descent with fixed step size and gradient descent with parallel line search, both using the Method 1 discussed in Section 5.3.4 for estimating descent directions.

Figure 5.7 shows the optimization results of Figure 5.1. In Figure 5.7 (a), the three features from outside to inside appear in different transparency levels, from weak to strong. This reveals a clear perspective of the three structures. Figure 5.7 (b) to (d) are the optimized transfer function, the feature visibility histogram and the visibility-weighted saliency histogram respectively. Figure 5.7 (e) and (f) are the evolution of visibility-weighted saliency of each feature in gradient descents with fixed step sizes and parallel line search respectively. Figure 5.7 (g) and (h) show the evolution of the opacity of each feature (opacity of the peak control point) in gradient descents with fixed step sizes and parallel line search respectively.

Figure 5.8 (a) to (h) display the volume rendered image and the transfer function, the feature visibility histogram and the visibility-weighted saliency histogram of the tooth data set before and after optimization respectively. Figure 5.8 (i) and (j) are the evolution of visibility-weighted saliency of each feature in the gradient descent with fixed step sizes and parallel line search respectively. Figure 5.8 (k) and (l) illustrate the opacity of features in the gradient descent with fixed step sizes and parallel line search respectively.

Figure 5.9 ((a) to (h)) shows the volume rendered images of a CT-Knee data set, the transfer functions, and the feature visibility histograms and the visibility-weighted saliency histograms before and after optimization respectively.

Figure 5.10 ((a) to (h)) shows the volume rendered images of the first time-step of a vortex data set, the transfer functions, and the feature visibility histograms and the visibility-weighted saliency histograms before and after optimization respectively.

The evolution of visibility-weighted saliency and opacity of each feature, in gradient descents with fixed step sizes and parallel line search, are displayed in Figure 5.9 (i) to (l) and Figure 5.10 (i) to (l) for the CT-Knee data set and the vortex data set respectively.

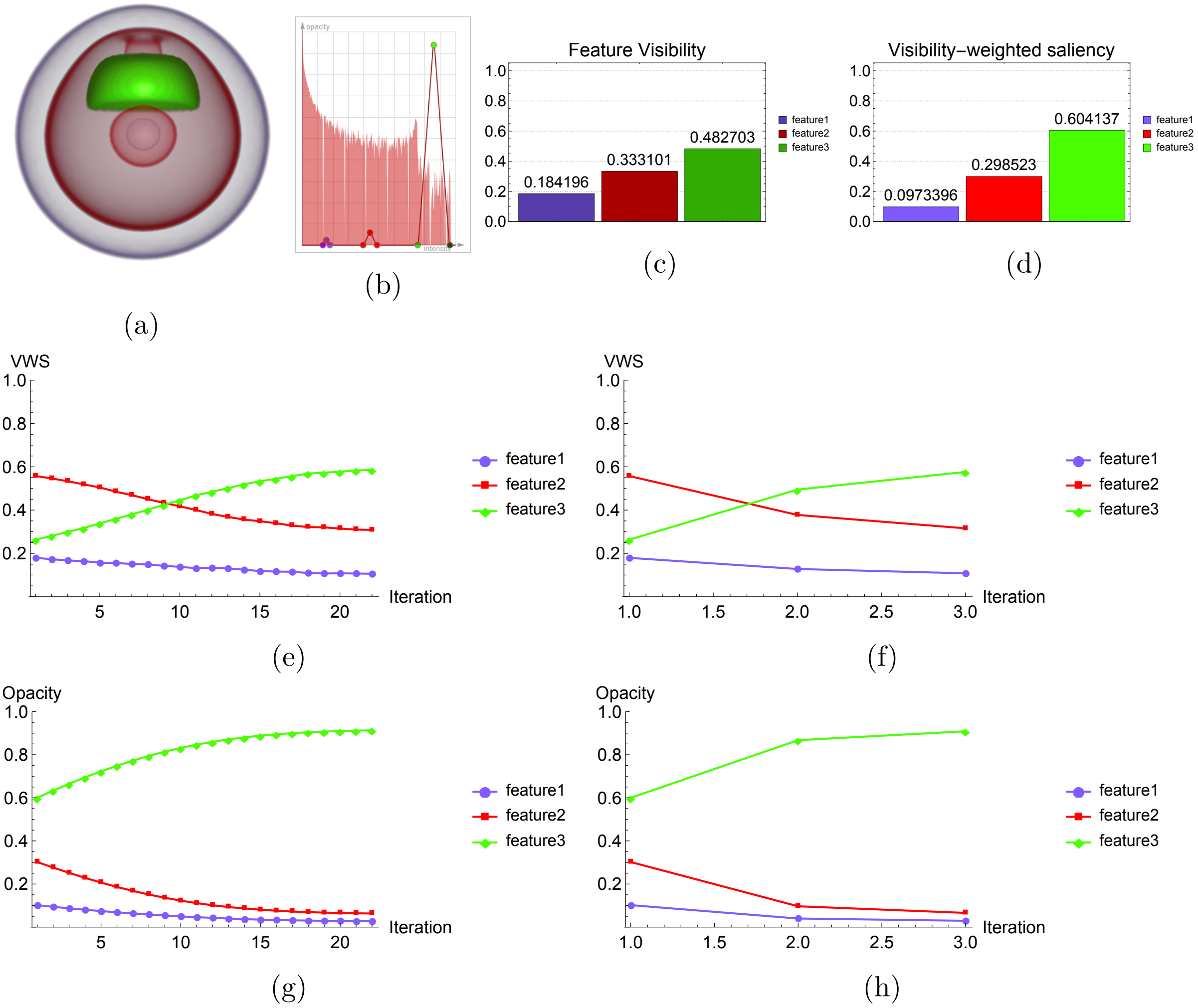
In the optimized results of the 4 data sets, the visibility-weighted saliency values of the 3 features are very close to the user-specified targets, as shown in Figure 5.7 (d), Figure 5.8 (h), and Figure 5.9 (h).

Moreover, performance tests of the optimization approaches were conducted on the 4 data sets. In the tests, 0.05 is used as the step size and the objective function F falling below 0.03 is regarded as convergence. For the 4 data sets we generated an "ideally optimized" image using a large number of iterations and compared this, using the SSIM metric [200], to images at different progressive stages of optimization. We noted, across the 4 data sets, that for values of the energy function below 0.03, the SSIM scores settled consistently at over 0.99, which was taken as an indicator that further iterations lead to an almost imperceptible change to the rendered image. In practice this threshold can be chosen as demanded by the application or determined using a perceptual study.

Table 5.1 displays the number of iterations and the time taken for the objective function to converge. We noticed the two line search methods require much fewer steps to converge than the fixed step-size method and they made the same choices of adaptive step sizes during the iterations (the two curves completely overlapped). In addition, although the two line search methods use the same numbers of steps, the parallel line search is significantly faster than the sequential line search.

Figure 5.11 shows the convergence time of the parallel line search approach on the 4 data sets over 1, 2, 4 and 8 CPU threads respectively. We observed that the computation time decreased as the number of threads increased. However, the speedup from 4 threads to 8 threads is minor, which may due to the fact that the CPU of the experiment computer only has 4 cores.

For the sake of clarity of presentation and convenience of the user study, the results in this section are includes cases of transfer functions with three features and three distinctive feature colors. See Section 5.4.2 for details of applying the transfer function optimization approach to other transfer functions and other color schemes.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.7: Optimization results of nucleon. After optimization to target $\{0.1, 0.3, 0.6\}$, all the 3 features are visible and the green feature inside is particularly emphasized. (a) The optimized volume rendered image of the nucleon data set; (b) The optimized transfer function; (c) The feature visibility histogram [6]; (d) The visibility-weighted saliency histogram; (e) & (f) VWS of features for gradient descent with fixed step sizes and parallel line search respectively; (g) & (h) Opacities of features (opacities of peak control points) for gradient descent with fixed step sizes and parallel line search respectively.

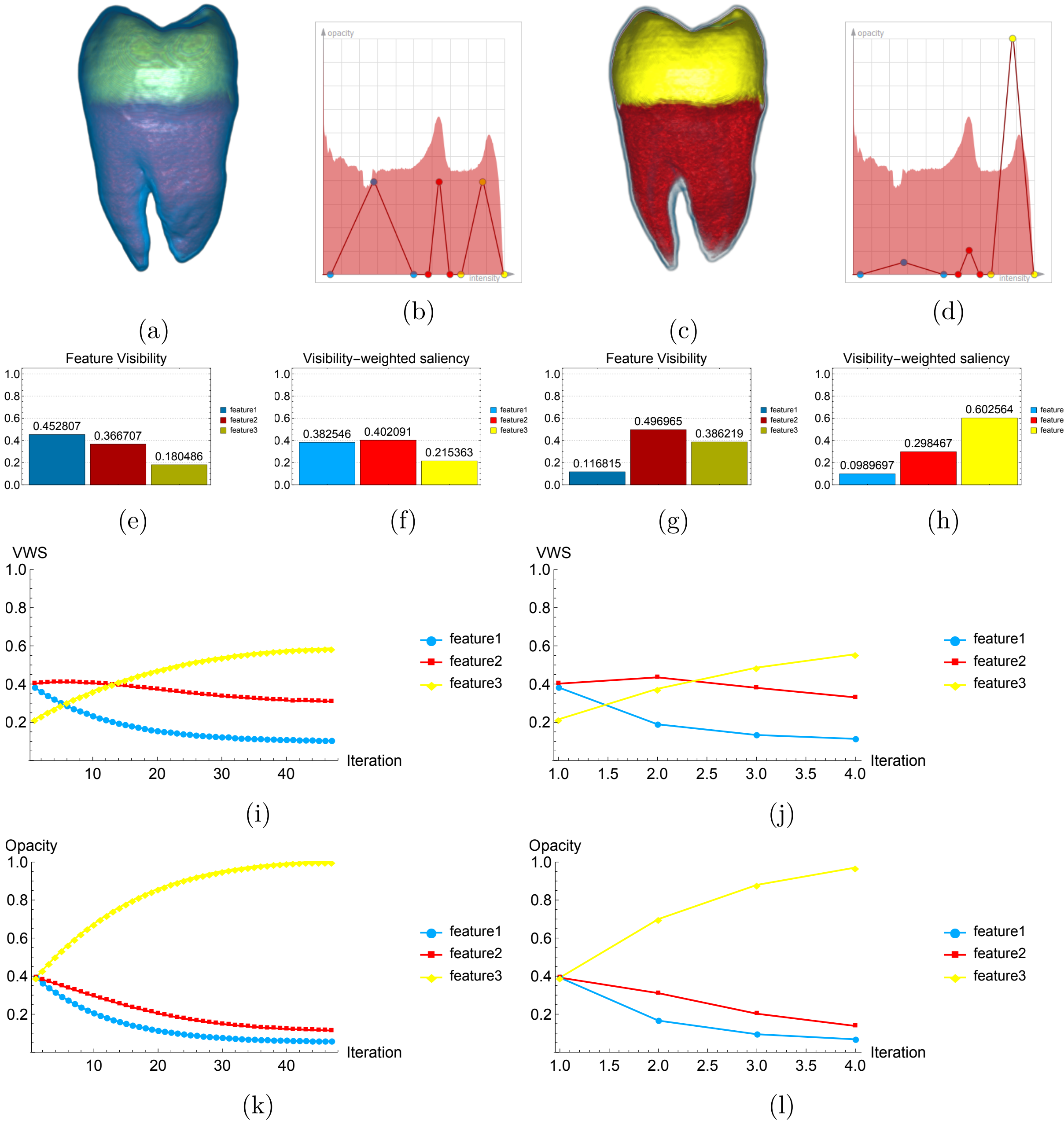
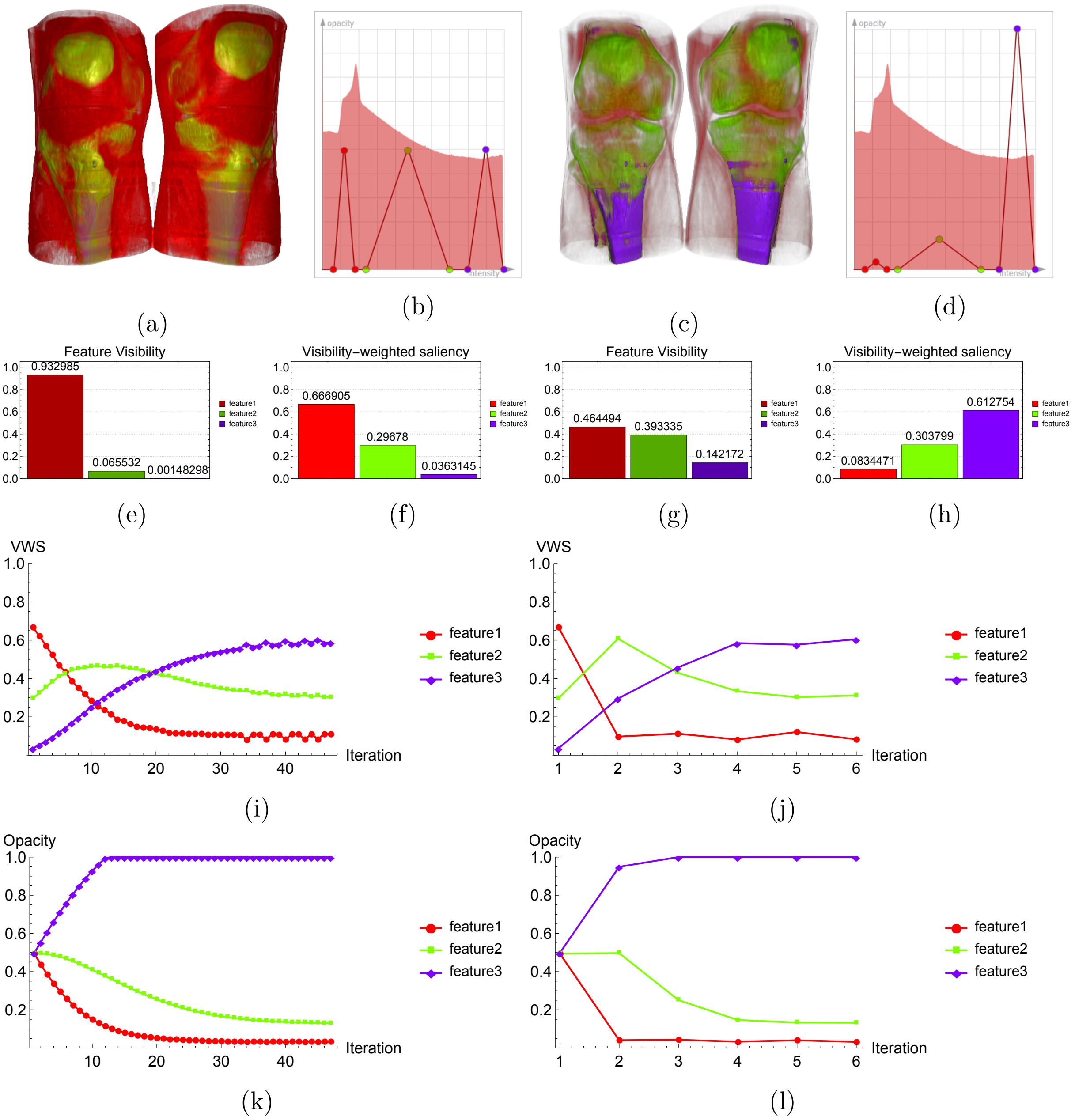


Figure 5.8: Optimization results of tooth. After optimization to target {0.1, 0.3, 0.6}, all the 3 features are visible and the yellow feature inside is particularly emphasized. (a) & (b) Initial volume rendered image and transfer function; (c) & (d) Optimized volume rendered image and transfer function; (e) & (f) Feature visibility and VWS of features before optimization; (g) & (h) Feature visibility and VWS of features after optimization; (i) & (j) VWS of features for gradient descent with fixed step sizes and parallel line search respectively; (k) & (l) Opacities of features for gradient descent with fixed step sizes and parallel line search respectively.



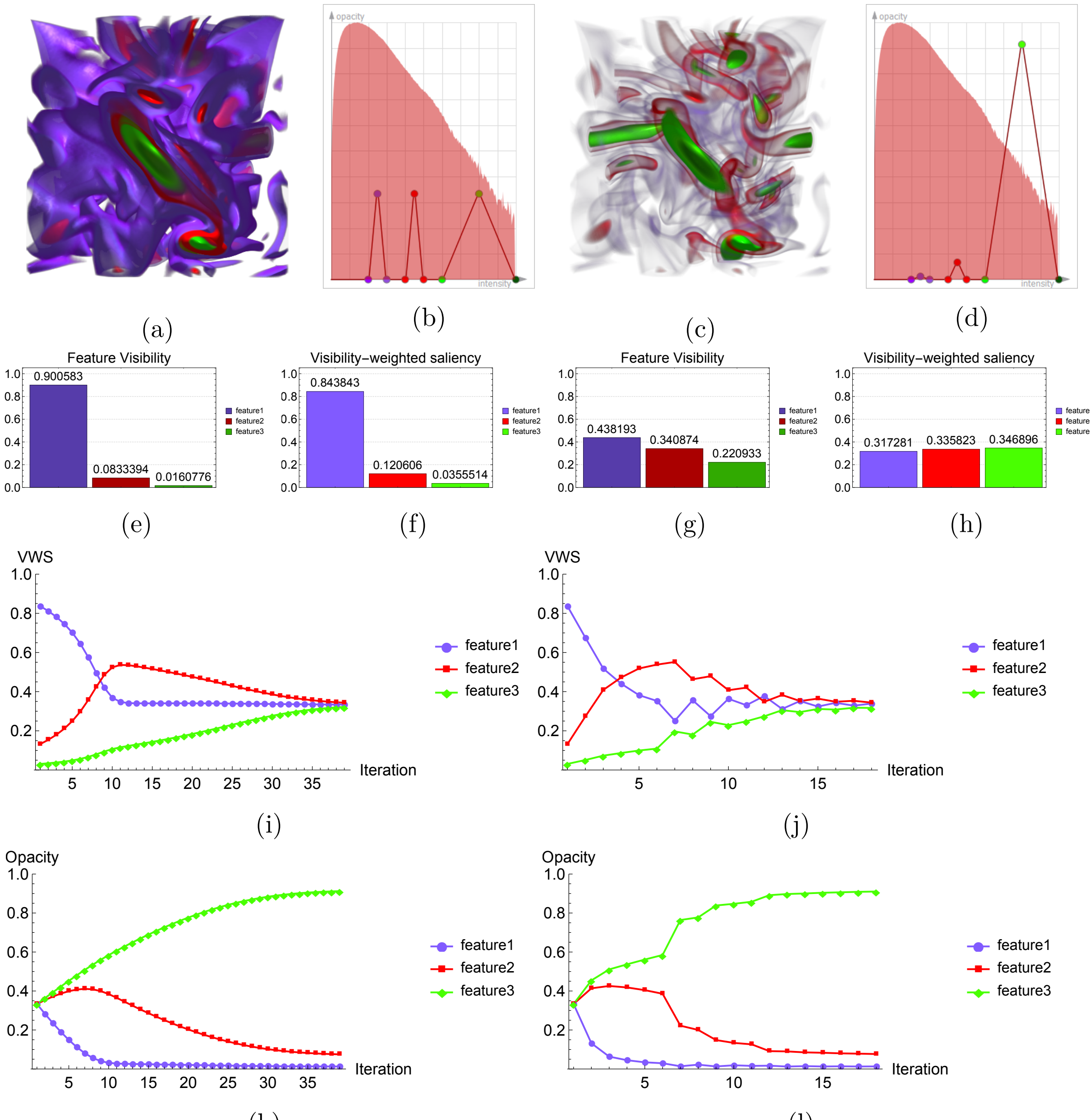
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

(k) (l)

Figure 5.9: Optimization results of CT-Knee. After optimization to target {0.1, 0.3, 0.6}, all the 3 features are visible, and the green and the purple features inside become clearer. (a) & (b) Initial volume rendered image and transfer function; (c) & (d) Optimized volume rendered image and transfer function; (e) & (f) Feature visibility and VWS of features before optimization; (g) & (h) Feature visibility and VWS of features after optimization; (i) & (j) VWS of features for gradient descent with fixed step sizes and parallel line search respectively; (k) & (l) Opacities of features for gradient descent with fixed step sizes and parallel line search respectively.



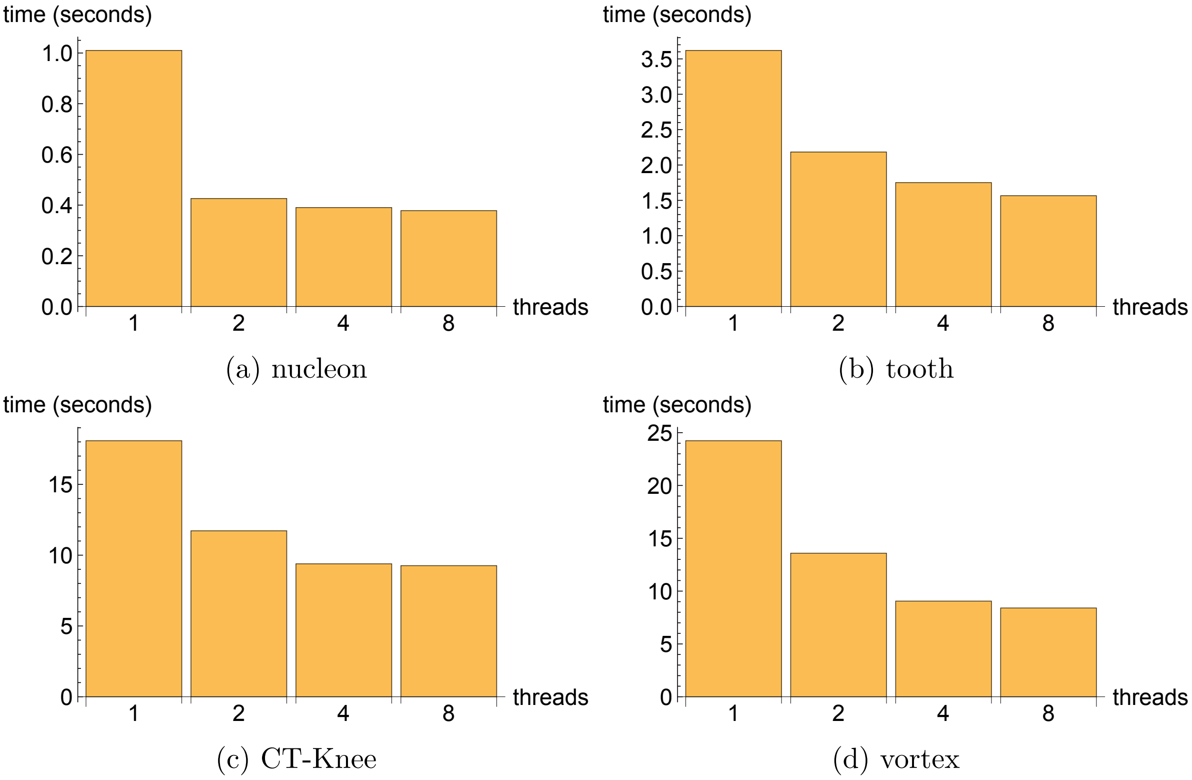
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

(k) (l)

Figure 5.10: Optimization results of vortex. After optimization to target {1/3, 1/3, 1/3}, all the 3 features are visible and the green feature inside is particularly more emphasized in comparison to the unoptimized result. (a) & (b) Initial volume rendered image and transfer function; (c) & (d) Optimized volume rendered image and transfer function; (e) & (f) Feature visibility and VWS of features before optimization; (g) & (h) Feature visibility and VWS of features after optimization; (i) & (j) VWS of features for gradient descent with fixed step sizes and parallel line search respectively; (k) & (l) Opacities of features for gradient descent with fixed step sizes and parallel line search respectively.



(a) nucleon

(b) tooth

(c) CT-Knee

(d) vortex

Figure 5.11: Performance of parallel line search (seconds taken to converge) over different number of CPU threads

		Fixed step size	Line search	Parallel line search
nucleon	Steps to converge	17	2	2
	Time (seconds)	1.07	0.59	0.38
tooth	Steps to converge	21	2	2
	Time (seconds)	7.56	3.25	1.57
CT-Knee	Steps to converge	17	2	2
	Time (seconds)	33.84	17.81	9.26
vortex	Steps to converge	33	13	13
	Time (seconds)	14.00	15.22	8.40

Table 5.1: Performance of the 3 optimization approaches showing steps and time (seconds) taken to converge ($F < 0.03$ is regarded as convergence.)

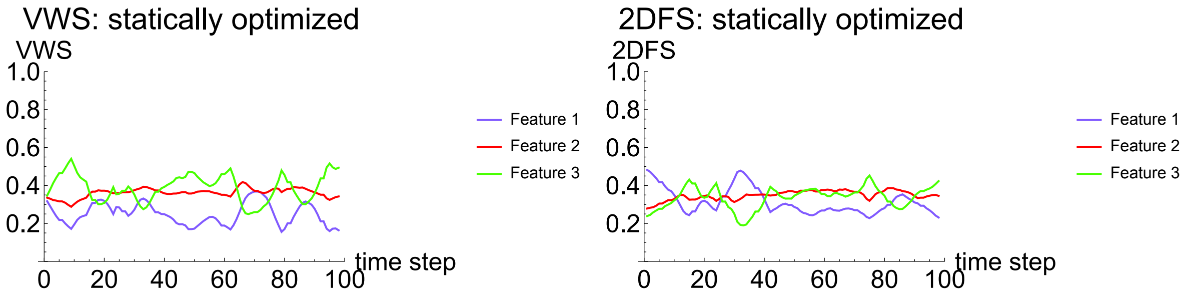
5.4.1 Transfer Function Optimization For Time-Varying Data Sets

Similar to the approach discussed in Section 3.5.4, we apply our transfer function optimization on all the time steps of the vortex data set. Our optimizer dynamically optimizes the transfer function to the same user-specified target (equal weights i.e. (1/3, 1/3, 1/3) were set as target in this test) for each time step of the time-varying data set.

Figure 5.12 displays the temporal curves of the visibility-weighted saliency (VWS) and 2D feature saliency (2DFS, discussed in Section 4.5.3) of the visualization with a static transfer function (optimized for the first time step) respectively. Similarly, Figure 5.13 displays the temporal curves of the VWS and 2DFS of the visualization with a dynamic transfer function (optimized for each time step) respectively. The VWS curves in Figure 5.13 (a) are more converged than the VWS curves in Figure 5.12 (a) because the dynamic transfer function is constantly optimized towards the target (1/3, 1/3, 1/3).

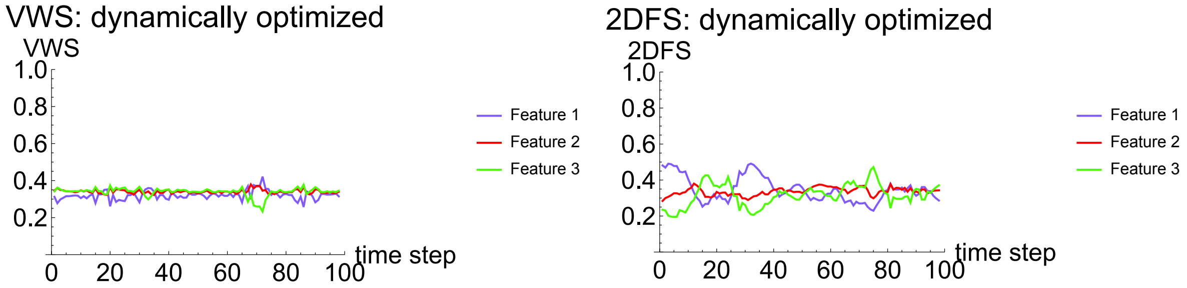
The 2DFS curves in Figure 5.12 (b) and Figure 5.13 (b) are provided as comparison for the fact that 2DFS is an image space technique independent from VWS. We notice that there are more small changes in the 2DFS curves in Figure 5.12 (b) and the three curves (representing the 2DFS of the three features) cross each other more often than the 2DFS curves in Figure 5.12 (b).

Time step 30 and 80 rendered with the static transfer function and the dynamic



(a) (b)

Figure 5.12: (a) VWS and (b) 2DFS of the vortex data set with a static transfer function only optimized for the first time step



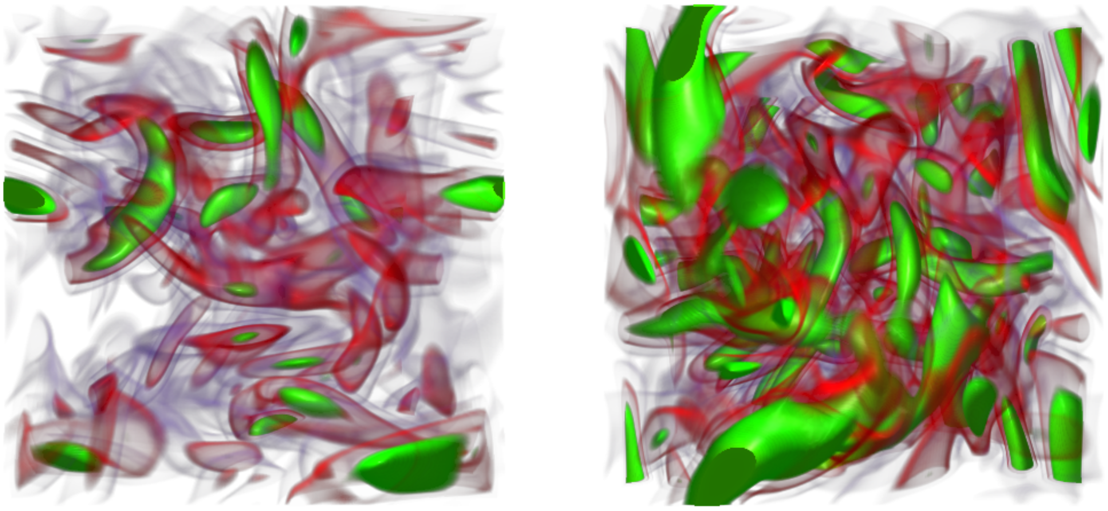
(a) (b)

Figure 5.13: (a) VWS and (b) 2DFS of the vortex data set with a dynamic transfer function optimized for each time step

transfer function are displayed in Figure 5.14 and Figure 5.15 respectively. As coherence can be an important factor in time-varying visualization, we notice the dynamic transfer function can maintain similar level of visual saliency for the purple feature when the sizes and proportions of the features change over time.

5.4.2 Generality of Transfer Functions

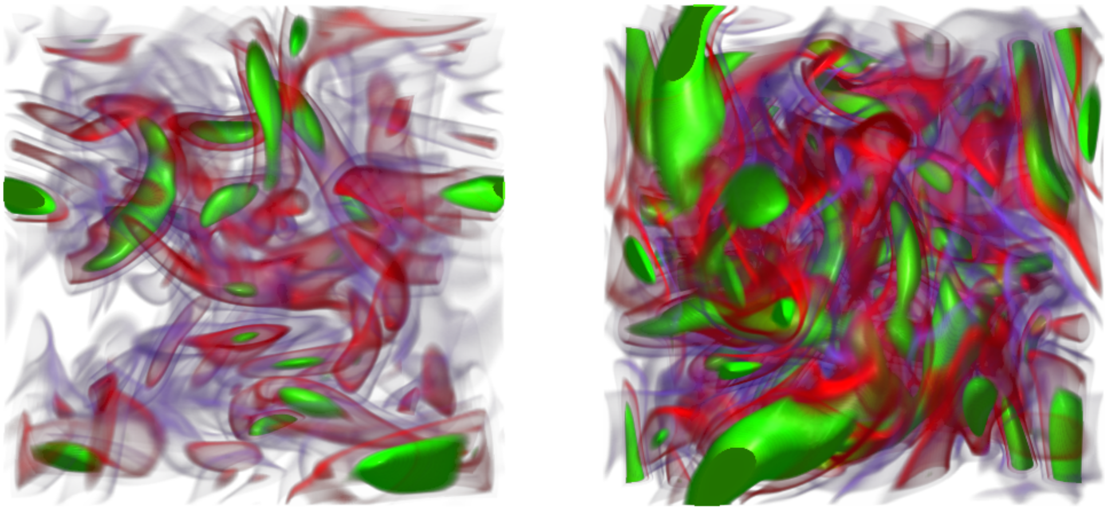
We have provided examples of transfer functions with three features and three distinctive feature colors. However, this was purely for the sake of clarity of presentation as well as to be able to specify, for the user study, tasks that could be easily explained to the user. In this section, we provide some examples to demonstrate that the approach can, in fact, be applied equally well to other transfer functions and other color schemes. Furthermore, we wish to examine if the iterative optimization is reasonably robust to



(a)

(b)

Figure 5.14: Time step 30 (a) and time step 80 (b) rendered with a static transfer function only optimized for the first time step



(a)

(b)

Figure 5.15: Time step 30 (a) and time step 80 (b) rendered with a dynamic transfer function optimized for each time step

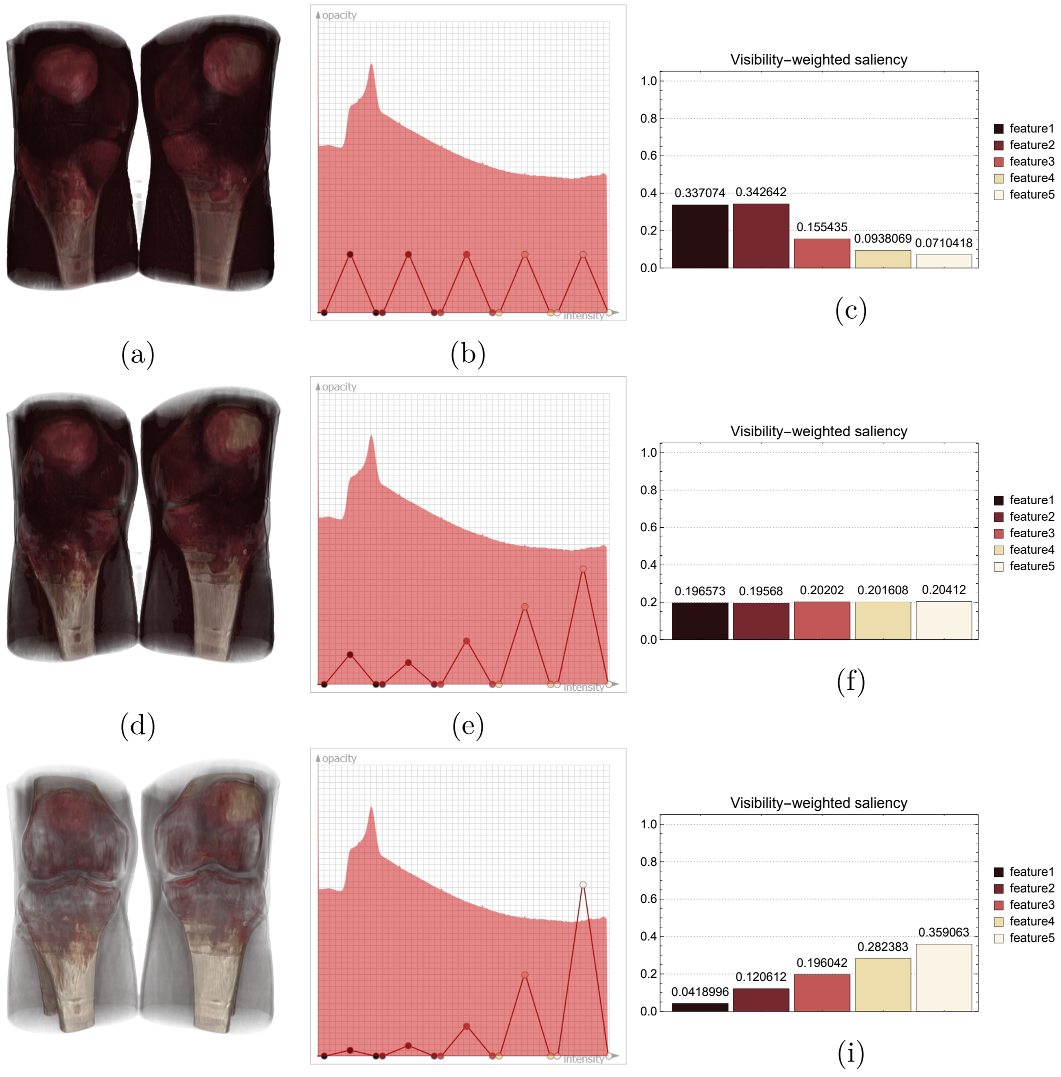
changes in initial conditions. We present results of the CT-Knee data set rendered using transfer functions with a color mapping that is more likely to be used in a real-world application. Specifically, we chose the *CT-Bone* transfer function provided with the medical imaging tool, 3D Slicer [1]. For the opacity channel, two types of initial transfer functions are used in this section, namely a function with peaks of equal opacities and another with peaks of linearly increasing opacities. The transfer functions are optimized towards two different VWS targets, i.e. equal VWS and linearly increasing VWS.

Figure 5.16 displays a visualization of the CT-Knee data set before and after optimization to the two VWS targets. In this example, the transfer function is initially set to have equal peak opacity values for all five features. The top row (Figure 5.16 (a) - (c)) shows the volume rendered image, the initial transfer function, and the VWS graph respectively. The second row (Figure 5.16 (d) - (f)) shows the rendered image, transfer function and the VWS graph after optimizing towards a target with equal conspicuity for all features. The bottom row (Figure 5.16 (g) - (h)) shows the results after optimizing towards a VWS target with linearly increasing values for each subsequent feature.

Similarly, Figure 5.17 displays a similar set of examples for the same data set, however the initial transfer function in this case is set to linearly increasing peak opacities, in order to test how the initial conditions impact the resulting optimization.

Although, we observe, by comparing Figure 5.16(e), (h) and Figure 5.17 (e), (h) respectively, that the optimized transfer function for the corresponding targets are slightly different, the final volume rendered images turn out to be very similar in appearance. We also ran tests with the four data sets and opacity transfer functions with differing number of features, ranging from 3 to 9 tent-shaped peaks. We observed similar behavior to the CT-Knee examples above, indicating that the process is not significantly sensitive to initial conditions in terms of both output quality and performance.

[1] www.slicer.org

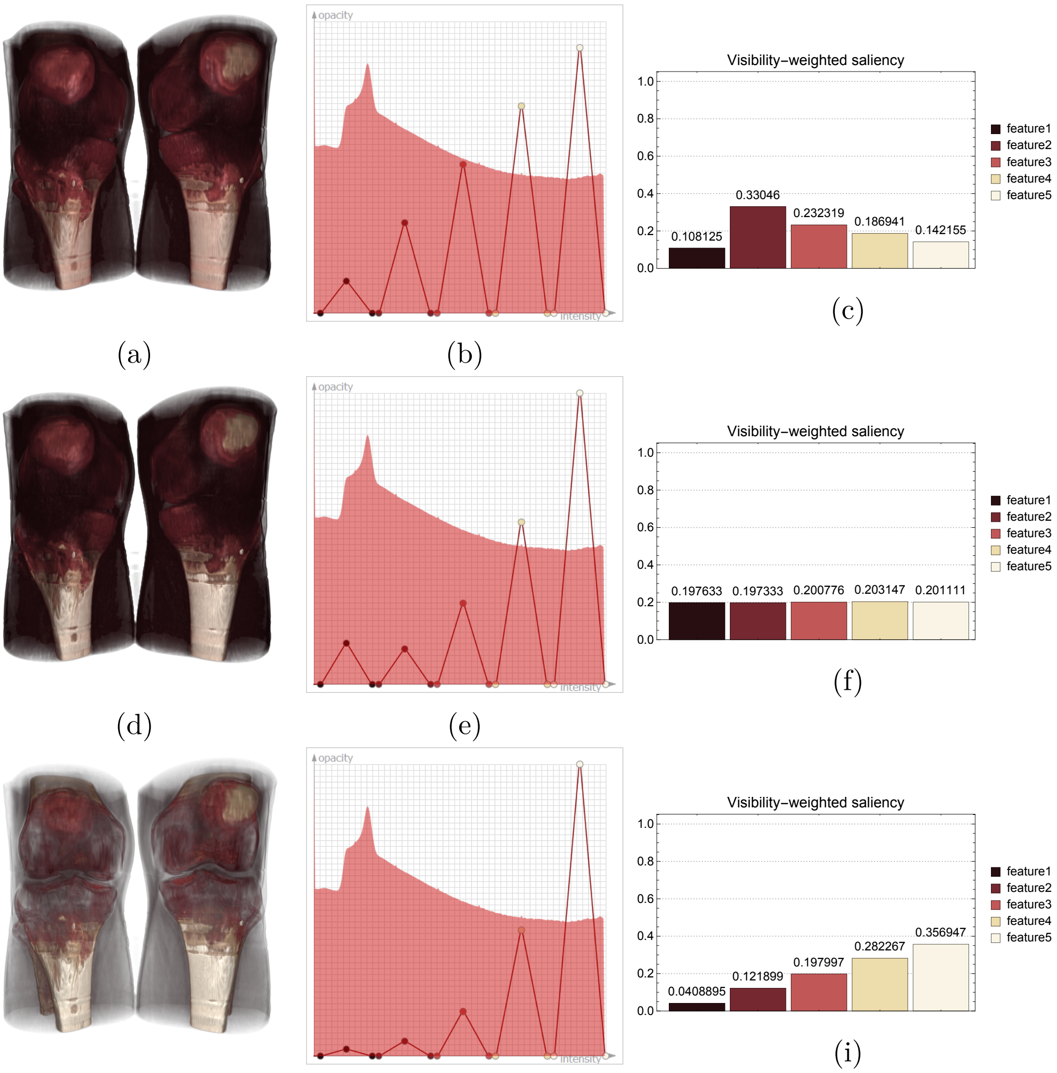


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.16: CT-knee: (a), (b) and (c) are the volume rendered images, transfer functions and VWS graphs respectively; (d), (e) and (f) are the images after VWS-optimization to a even VWS target, where all features have similar VWS; (g), (h) and (i) are the images after VWS-optimization to a diagonal VWS target, where the internal features are clearer.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.17: CT-knee: (a), (b) and (c) are the volume rendered images, transfer functions and VWS graphs respectively; (d), (e) and (f) are the images after VWS-optimization to a even VWS target, where all features have similar VWS; (g), (h) and (i) are the images after VWS-optimization to a diagonal VWS target, where the internal features are clearer.

5.5 Conclusions

This chapter proposes a novel transfer function optimization approach using the visibility-weighted saliency metric. With this approach, the design of transfer functions becomes more intuitive. This approach allows the user to directly set target visibility-weighted saliency for features of interest and then the transfer function is automatically refined to match the visibility-weighted saliency of the features with user-defined targets. In addition, a parallel line search strategy is presented for exploiting the computing power of multi-core processors to improve the performance of the transfer function optimization approach. This approach has proven to be effective over several volume data sets.

Chapter 6

Selective Saturation and Brightness for Visualizing Time-Varying Volume Data

6.1 Introduction

Time-varying volume data is used in many areas of science and engineering, however, visualizations of such data are not easy for users to visually process due to the amount of information that can be presented simultaneously. In this chapter, we propose a novel visualization approach which modulates focus, emphasizing important information, by adjusting saturation and brightness of voxels based on an importance measure derived from temporal and multivariate information. By conducting a voxel-wise analysis of a number of consecutive frames, we acquire a volatility measure of each voxel. Subsequently, the opacity, saturation and brightness are used to represent intensity, volatility and additional multivariate information in the data set. The method was tested in visualizing a multivariate hurricane data set and a smoke data set. The results suggest that our method can give the user a more detailed understanding of the data by presenting multivariate information variables, including time-variant characteristics, in one self-contained visualization.

6.2 Related Work

The visualization of time-varying data is an important and active topic in the visualization community. Transfer function specification for static volume data has been widely studied over the years [19]. However, much less work has been done for transfer function design of time-varying data.

Jankun-Kelly and Ma first studied transfer function specification for time-varying data [121]. Kniss and Hansen applied the techniques from multidimensional transfer function based volume rendering to the visualization of multivariate data from weather simulations [109]. Akiba et al. [117] described three approaches for the data-fusion problem in multivariate data visualization. One approach, which is to use one variable for each color channel in RGB space, is popular because of its simplicity but is limiting due to the difficulty for viewers to interpret the resulting color. The second approach, is to use one of the values based on some criterion e.g. [201] use alternating sampling for rendering two volumes and this has been shown to work well for medical imaging but not for fluid flow visualization. The third approach is to compute a weighted sum of all the values. This approach is more flexible however this may not be guaranteed to lead to an effective visualization as blending different colors might lead to ambiguous mixing of different hues.

6.3 Method

In this section, we propose an intuitive approach of using color mappings in the HSB color space to effectively represent multivariate time-variant data. We use opacity to represent the main variable and saturation to represent the volatility (a temporal characteristic) of the main variable. In addition, brightness can be used to represent other information such as an additional variable from a multivariate or multidimensional data set. This allows constraining hues, which are sometimes chosen based on some domain knowledge and changing hues across different time steps could be unintuitive.

For a multivariate data set with two variables X, Y, we use variable X of voxels for the alpha channel and modulate the saliency of voxels by adjusting saturation and brightness based on the volatility of variable X and the values of variable Y respectively. The volatility is measured by temporal standard deviation std(i), which is the standard deviation of the i-th voxel in the recent n consecutive frames, where n is a user specified number. The hue, saturation, brightness and alpha of the i-th voxel (an element in a volume data set) in the HSB color space are defined as follows:

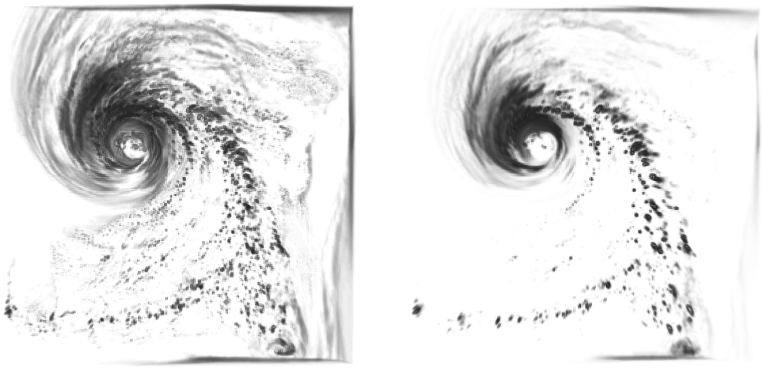
$$\begin{aligned} Hue(i) &= C(i) \\ Saturation(i) &= Clip(a \times std(i)) \\ Brightness(i) &= 1 - Clip(b \times Y(i)) \\ Alpha(i) &= X(i) \end{aligned} \tag{6.1}$$

where std(i) is the volatility and C(i) can be either a user-specified constant hue or mapped to a variable such as X(i). Clip is a function that clips the value to the range [0, 1]. *a* and *b* are scale factors for saturation and brightness respectively, which are determined by the user based on the distribution of variables X and Y in the data set. Variables X(i) and Y(i) are normalized to the range [0, 1].

6.4 Results

We tested our method on the hurricane data set from the National Center for Atmospheric Research provided for the IEEE Visualization 2004 Contest. The hurricane model is Hurricane Isabel, which was a very strong hurricane in the west Atlantic region in September of 2003. The resolution of the data set is 500 (longitude) × 500 (latitude) × 100 (height) × 48 (time). One time-step is one hour in the simulation. The variables we used in the tests are cloud moisture mixing ratio (kg water/kg dry air) and total precipitation mixing ratio (sum of mixing ratios of graupel, rain and snow). The ground in the hurricane data set, which is a height field of the surface topography, is rendered with relief mapping and presented as background in the visualization results.

The cloud, precipitation and volatility of cloud of frame 40 are displayed in Figure 6.1 and Figure 6.2 for reference. Figure 6.1 shows the cloud and precipitation, and Figure 6.2 shows the volatility of cloud over 5 frames, 10 frames and 20 frames respectively. Figure 6.3 shows images of 3 frames of the hurricane rendered with cloud, volatility of cloud and precipitation. Cloud is used as variable X and precipitation as variable Y. The temporal standard deviation is calculated using the recent 10 frames.



(a) (b)

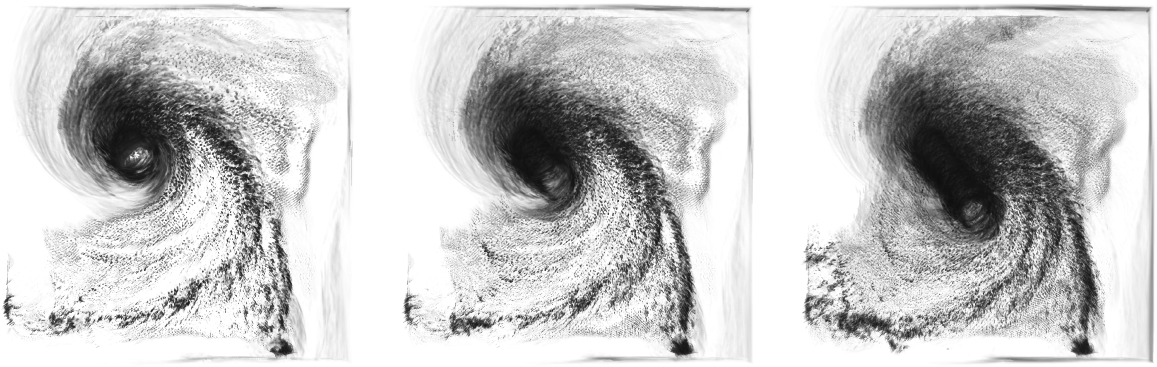
Figure 6.1: Cloud (a) and precipitation (b) at frame 40

Figure 6.4 (a) displays a 3D perspective view of the visualization in Figure 6.3 (b). The strong red color near the hurricane eye indicates where the hurricane was in previous several frames and the clouds with more precipitation are darker in the images.

Figure 6.4 (b) displays a 3D perspective view of the frame 40 of the hurricane in a cyan hue with both saturation and brightness representing the precipitation. Note how parts of the hurricane are desaturated and darkened based on the precipitation.

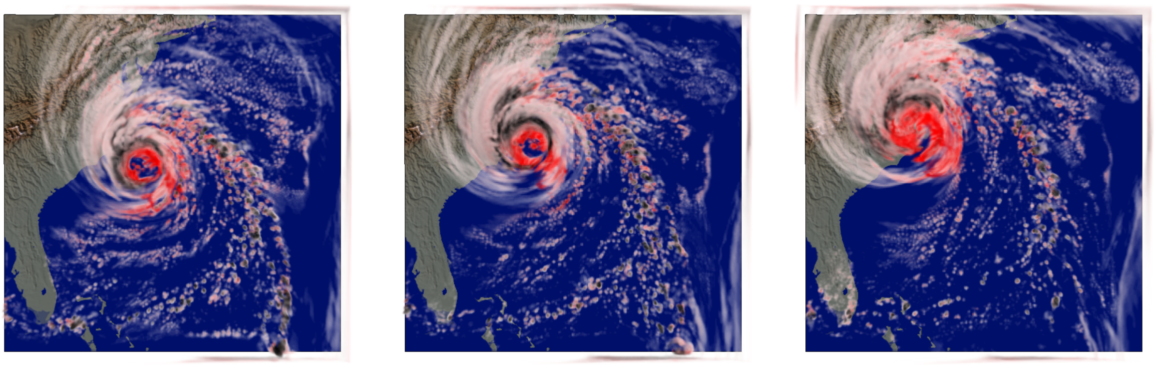
In addition, Figure 6.5 displays an alternative visualization, where the variables X and Y are swapped, i.e. precipitation is used as variable X and cloud as variable Y. Note the precipitation is desaturated based on its volatility over 10 frames and darkened based on the cloud.

We also tested our method on a time-varying smoke simulation based on an implementation of the approach by Fedkiw et al. [202]. Figure 6.6 shows the smoke density, volatility of smoke density and gradient magnitude of smoke density rendered in grayscale for comparison. The gradient magnitude is the Euclidean norm of the gradient at a voxel position, and the temporal standard deviation is calculated using the recent 10 frames. In Figure 6.7, the smoke density was used as variable X and gradient magnitude as variable Y and the parameters were $a = 10$ and $b = 5$. The hue was mapped to the smoke density in Figure 6.7 (a) and a constant red hue was used in Figure 6.7 (b).



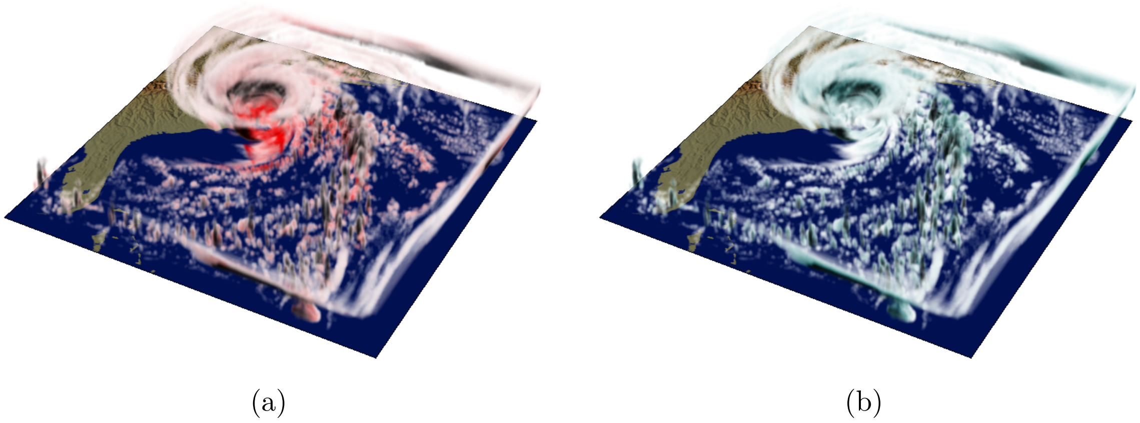
(a) (b) (c)

Figure 6.2: Volatility of cloud over 5 frames (a), 10 frames (b) and 20 frames (c) at frame 40



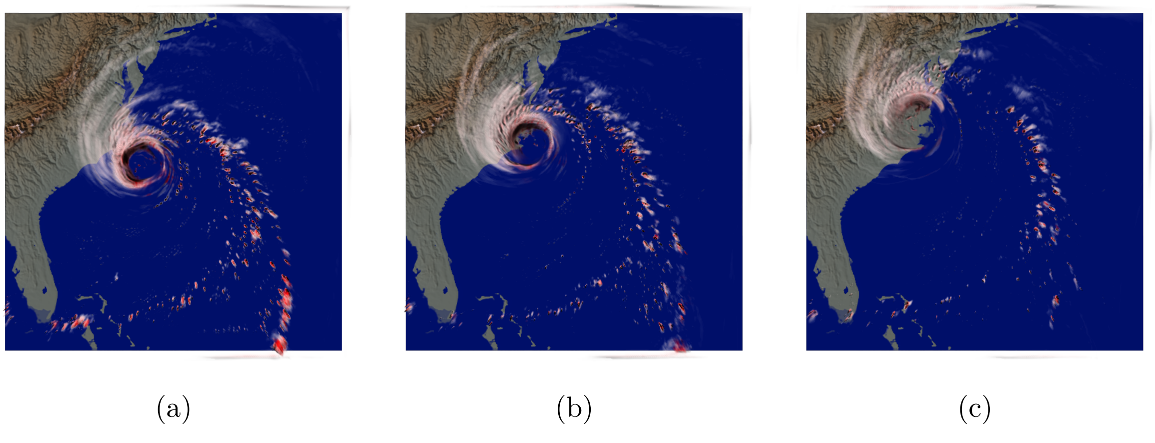
(a) (b) (c)

Figure 6.3: Adjusting saturation and brightness of frame 35 (a), frame 40 (b) and frame 45 (c)



(a) (b)

Figure 6.4: (a) A 3D perspective view of Figure 6.3 (b); (b) A 3D perspective view of the hurricane at frame 40 rendered with both saturation and brightness representing the precipitation, the hue is cyan, a=10 and b=10



(a) (b) (c)

Figure 6.5: An alternative visualization of the hurricane at frame 35 (a), frame 40 (b) and frame 45 (c)

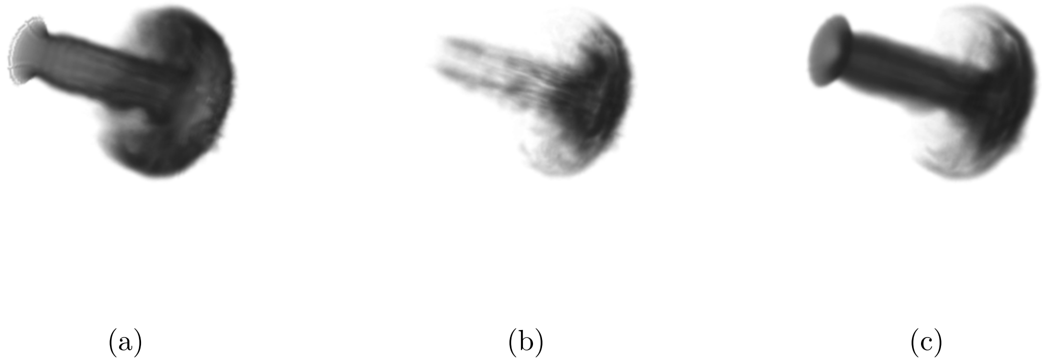


Figure 6.6: Smoke density (a), volatility of density (b) and gradient magnitude of density (c) at frame 160

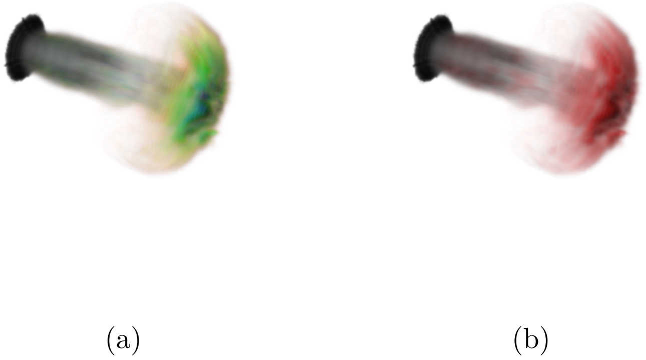


Figure 6.7: (a) Smoke density as variable X and gradient magnitude as variable Y at frame 160. Hue is X(i), a=10 and b=5; (b) Hue is red. Note how parts of the smoke are desaturated and darkened based on the volatility and gradient magnitude

6.5 Conclusions

Our main contribution is a mechanism for exploiting saturation and brightness to modulate focus in time-variant volume visualization using an importance measure that is based on volatility. In addition, we demonstrate how additional variables in a multivariate data set could be communicated simultaneously through the brightness channel. Preliminary results indicate that the approach can provide more visual information for the test data sets.

Clearly, more research is warranted in this area including a generalization of the approach to different multivariate characteristics, perceptual evaluation with user studies and integrating this form of optimization with the other results reported in this thesis. However, this was felt to be too far outside of the core scope of this thesis thus this is left for future work. Nevertheless, we present the preliminary results here as we feel they represent noteworthy results in a closely related area of research.

Chapter 7

Conclusions

This chapter provides an overview of the contributions of this thesis and directions for future work on visualizing volume data with automated techniques.

7.1 Summary of Contributions

Volume visualization is a broad field covering many subtopics. The goal of this thesis was to provide a framework of automated transfer function techniques for obtaining clear visualization of features of interest in volume data. This thesis proposed and investigated novel automated optimization techniques for emphasizing features of interest in volume visualization using entropy, visual saliency and visibility of voxels.

We have presented a transfer function refinement approach, which exploits the entropy of voxels to equalize the opacity transfer function, in order to reduce general occlusion and improve the clarity of features of interest in the resulting visualization. Furthermore, this approach assists the user in exploring and enhancing features of interest by interactively specifying either priority intensity ranges in the transfer function domain or regions of interests in the resulting visualization. Our approach is different from Ruiz et al. [66], where the transfer function is adjusted towards a user-defined target distribution by minimizing the informational divergence between the transfer function and the user-defined target distribution.

In addition to view-independent information, we have proposed visibility-weighted saliency for measuring the view-dependent saliency of features of interest for volume visualization. This metric aims to assist users in choosing suitable viewpoints and designing effective transfer functions to visualize the features of interest. Compared to existing approaches such as visibility histogram [63] and feature visibility [6] which only measure the visibility of voxels, our approach reflects two aspects of the resulting visualization, i.e. voxel visibility and visual saliency. Our approach is aware of the change of other appearance attributes, i.e. saturation and brightness, while existing approaches do not take into account this information. A user study was conducted to evaluate the efficiency of our metric in comparison to feature visibility and 2D feature saliency.

Subsequently, we have described an automated transfer function optimization method based on the visibility-weighted saliency metric. This method takes into account the perceptual importance of voxels and the visibility of features, and automatically adjusts the transfer function to match the target saliency levels specified by the user. In addition, a parallel line search strategy is presented to improve the performance of the optimization algorithm.

Finally, we have developed a novel visualization approach which modulates focus, emphasizing important information, by adjusting saturation and brightness of voxels based on an importance measure derived from temporal and multivariate information. By conducting a voxel-wise analysis of a number of consecutive frames, we acquire a volatility measure of each voxel. We then use intensity, volatility and additional multivariate information to determine opacity, saturation and brightness of the voxels.

7.2 Limitations and Future Work

7.2.1 Automated Transfer Function Approaches

The main limitation of our automated transfer function approaches (Chapter 3 and Chapter 5) is that the variations to transfer functions are limited to color and opacity, and an initial setup of control points over the intensity ranges has to be defined by the user. Therefore, prior knowledge of the data sets may be necessary in choosing the most ideal intensity ranges for placing the control points.

In future work we plan to develop transfer function generation methods which identify important features in volume data sets and combine them with our automated transfer function optimization approaches. A promising direction would be the rule-enhanced transfer function generation method [153], which defines features with rules based on the frequency distribution of data attributes (e.g. intensity and gradient magnitude) and employs machine learning algorithms to select a set of rules that are most effective in distinguishing the target tissue from other tissues. We would also like to examine the feasibility of optimizing other color components such as saturation, brightness and even hue.

The initial choice of intensity ranges, number of control points and color mapping across the histograms can affect the quality of the final output and some prior knowledge of the data sets may be of benefit for optimal results. On the other hand the simple and straightforward techniques presented in this paper should be fully compatible with independent mechanisms for choosing optimal combinations of other visual parameters or indeed if the user wishes to combine these with more manual choices of parameters such as the color map. In addition, the transfer functions in our proposed system are intuitive and easy to use. Users may benefit from the flexibility of being able to further tweak the intensity or opacity of the control points after the application of the automated optimization techniques discussed in this thesis.

7.2.2 Visibility-Weighted Saliency

The major limitation of the visibility-weighted saliency metric presented in Chapter 4 is that it cannot currently detect changes in hue and orientation in volume visualization. The saliency field used in the metric is computed using the center surround mechanism based on saturation and brightness of voxels, but other factors such as hue of the voxels and orientation of the local neighborhoods around the voxels are not taken into account.

The center-surround operator [18] we used in computing saliency fields of volume data is essentially a Laplace differential operator. This operator can handle scalar fields of data attributes such as opacity, saturation and brightness. However, computing the difference between colors and orientations is more complicated, we have not yet integrated it in our model.

In the future, we would like to examine the feasibility of using other appearance attributes such as hue and orientation, as well as the weighting between these attributes.

The difference between colors could be computed for each pair of voxels in a color space, e.g. the perceptually uniform LAB color space. The orientations of voxels could be estimated using the Gabor filter, which would be similar to the construction of the orientation conspicuity map [16].

In addition to transfer function optimization, we would like to implement an automatic viewpoint optimization technique based on the proposed metric.

7.2.3 Selective Saturation and Brightness

We have not applied the visibility-weighted saliency metric to the work in Chapter 6. One reason is that the work in Chapter 6 was done before Chapter 4 and Chapter 5. On the other hand, it is difficult to apply the proposed metric to measure the features in Chapter 6's work, because the feature definition and transfer functions are very different from those in the previous chapters. Instead of defining features on intensity ranges, volatility (temporal standard deviation) is used in the feature definition in Chapter 6.

In the future, we would like to investigate how to apply the visibility-weighted saliency metric to other types of feature definitions and how automated optimization techniques can be exploited to better visualize time-varying and multivariate volume data sets. Automatic color scheme generation [93] for transfer functions would also be studied in the future work. We would also like to conduct perceptual experiments to quantitatively evaluate the mechanism and determine the optimal parameters for the approach.

Appendix A

Estimating Feature Saliency Using 2D Saliency Maps

A saliency map is a model of visual attention using bottom-up features such as intensity, color and orientation of an image. However, the traditional map provides an indication of saliency for a points on an image and not the saliency of a particular object or 3D feature. In order to use 2D saliency maps [16] to estimate visual saliency of 3D features in volume visualization, an inverse distance weighting [189] can be applied to divide a 2D saliency map into several feature saliency maps, one for each feature. Subsequently, the visual saliency of each feature can be estimated with the total intensity of each feature saliency map.

The distance between a pixel of each feature and the pixel in the final image is necessary in computing the inverse distance weighting. Hence, we perform volume rendering of each feature separately, i.e. other intensity ranges in the transfer function are set to zero except for the feature. These feature images $P_i (i \in \{1, ..., n\})$ are rendered with the same settings (viewpoint, screen size etc.) as the final image. In addition, a 2D saliency map S of the final image P is computed using the model by Itti et al. [16].

Let w_i be the weight of a pixel p in the *i-th* feature

$$w_i = \frac{\frac{1}{d_i^m}}{\sum_{j=1}^{n} \frac{1}{d_j^m}}$$

where d_i is the color distance between the pixel p in the final image and the corresponding pixel p_i in the *i-th* feature image, n is the number of features, and m is a user-defined coefficient for controlling the bias of the weighting. Pixels with small distances would have larger weights when m increases. $m = 1$ is used and the color distance d_i is computed in the LAB color space in our implementation

Then the corresponding pixel s_i in the *i-th* feature saliency map S_i is

$$s_i = w_i s$$

where s is the pixel in the 2D saliency map S of the final image.

Therefore, we can obtain n feature saliency maps by performing the above a pixel-wise operation using the 2D saliency map S and the final image P along with each feature images P_i respectively.

Figure A.1 shows an engine block (P) and its two features (P_1, P_2). Figure A.2 shows the 2D saliency map S and the two feature saliency maps(S_1, S_2) obtained using the above operation. The saliency maps in Figure A.2 are enhanced (multiplied by 8) for better contrast in illustrations. However, the original (i.e. not enhanced) saliency maps are used in the actual computation.

In practice, saliency resulting from a visual feature is not sharply delimited by the boundary of the feature, instead strong feature edges tend to attract attention increasing saliency in a small distribution around the edge. After distributing the 2D saliency map S into feature saliency maps $S_i(i \in \{1, ..., n\})$ using the inverse distance weighting, a small amount of bright pixels around the boundary of the engine block remain in the residual saliency image S', as shown in Figure A.3 (a). Let the residual saliency image be S'.

$$S' = S - \sum_{j=1}^{n} S_j$$

We distribute this residual saliency image S' to the features according to their influence in the region. The influence of the features are approximated by Gaussians of the feature saliency images, as shown in (b) and (c) of Figure A.3. Firstly, we apply a Gaussian filter with kernel size k to each feature saliency map S_i and get a Gaussian image G_i. In practice, the kernel size k should be large enough in order to allow the resulting Gaussian images to have non-zero pixels cover most of the bright pixels in the residual saliency image S'. Let g_i be a pixel in the Gaussian image G_i. Secondly, we distribute the residual saliency image S' into n images $(S'_1, ..., S'_n)$.

$$s'_i = \frac{g_i}{\sum_{j=1}^{n} g_j} s'$$

where s' is a pixel in S'. Figure A.4 displays the residual saliency images (S'_1, S'_2) of the two features on the engine block.

Thirdly, we pixel-wisely add the image S'_i to the feature saliency map S_i and obtain the total feature saliency map T_i of the *i-th* feature.

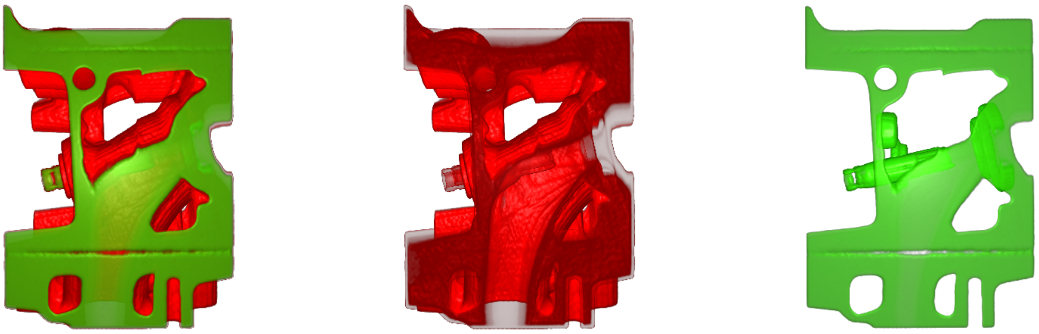
$$T_i = S_i + S'_i$$

Figure A.5 (a) and (b) display the total feature saliency maps of the red feature and the green feature respectively.

Finally, as shown in Figure A.5 (c), we compute the 2D feature saliency using the sum of intensity values of the total feature saliency maps, i.e. T_i for $i \in \{1, ..., n\}$. Hence, the 2D feature saliency of the *i-th* feature is

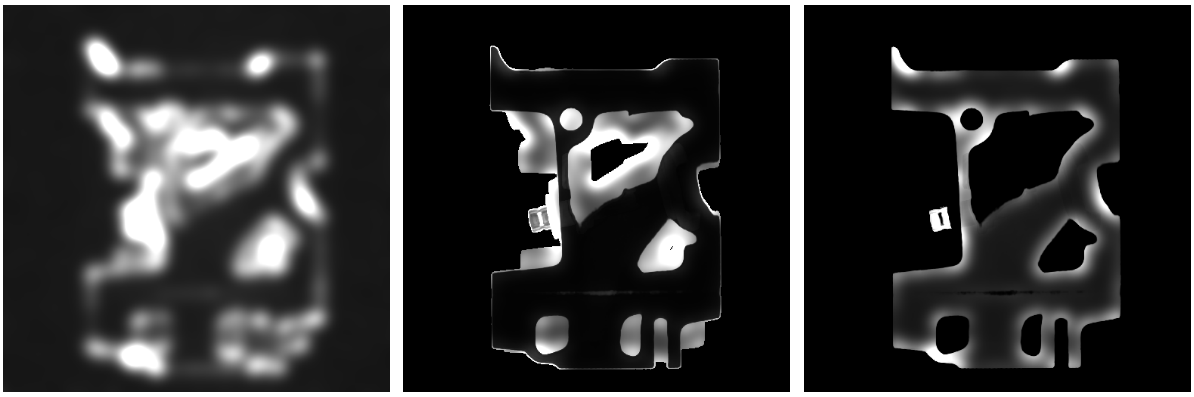
$$FS_i = \frac{Intensity(T_i)}{\sum_{j=1}^{n} Intensity(T_j)}$$

A histogram of the 2D feature saliency of the two features of the engine block is shown in Figure A.5 (c).



(a) P (b) P_1 (c) P_2

Figure A.1: (a) An engine block; (b) and (c) isolated volume rendering images of the red feature and the green feature

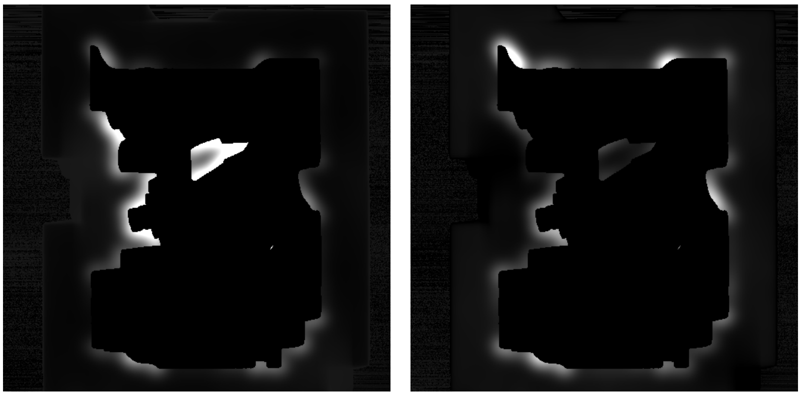


(a) S (b) S_1 (c) S_2

Figure A.2: (a) The 2D saliency map; (b) and (c) the feature saliency maps of the two features. The saliency maps are enhanced (multiplied by 8) for better contrast in illustrations.

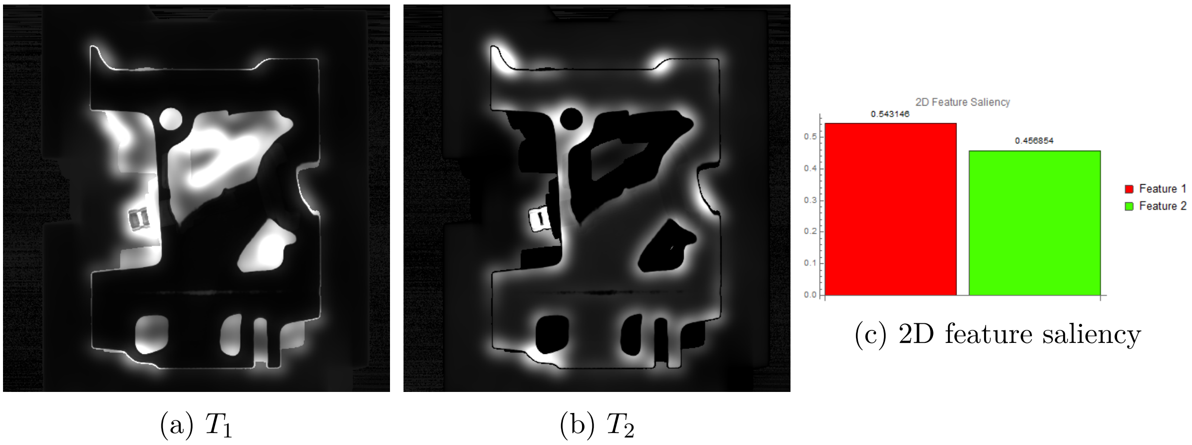
(a) S' (b) G_1 (c) G_2

Figure A.3: (a) The residual saliency image; (b) and (c) the Gaussians of the two feature saliency maps with a kernel size of one eighth of the image width



(a) S'_1 (b) S'_2

Figure A.4: The residual saliency images of the two features



(a) T_1

(b) T_2

(c) 2D feature saliency

Figure A.5: (a) and (b) The total feature saliency maps of the two features; (c) 2D feature saliency of the two features

Appendix B

Experiment Questionnaire

This is the questionnaire given to the participants in the experiment described in Section 4.5.

Example Questionnaire

The following parts are entered on paper before and after the experiment.

To be Completed Before Experiment

Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

Before you begin, you will be shown some example videos. Please do not hesitate to ask the experimenter if you have any questions before the experiment begins. It will last approximately 20 minutes.

Your results will be kept strictly confidential, stored using a unique ID number and the experimenter will not be able to identify your data or link them with your personal details. You are also free to withdraw from this study at any time without any penalty. If you have any further questions, please do not hesitate to ask the experimenter.

Please help us by filling in the below information:

Sex: Male □ Female □ **Age:** ……….

Familiarity with scientific visualization:

None □

Medium (e.g. student) □

High (e.g. expert) □

Do you have any type of colour blindness (colour vision deficiency)?

No □

Yes □

If yes, please specify the type of colour blindness (e.g. red-green, blue-yellow):

To be Completed After Experiment

Please note any comments you have below, making sure not to identify or name any individuals. (Filling this form is optional and you may skip it without any penalty.):

……

……

……

……

……

……

……

……

……

……

……

During the Experiment

Below are examples of the question the user is presented with on a computer screen at the end of each trial which consists of, being shown for 15 seconds, an image created using a computer visualisation technique.

Questionnaire for each image shown (sample)

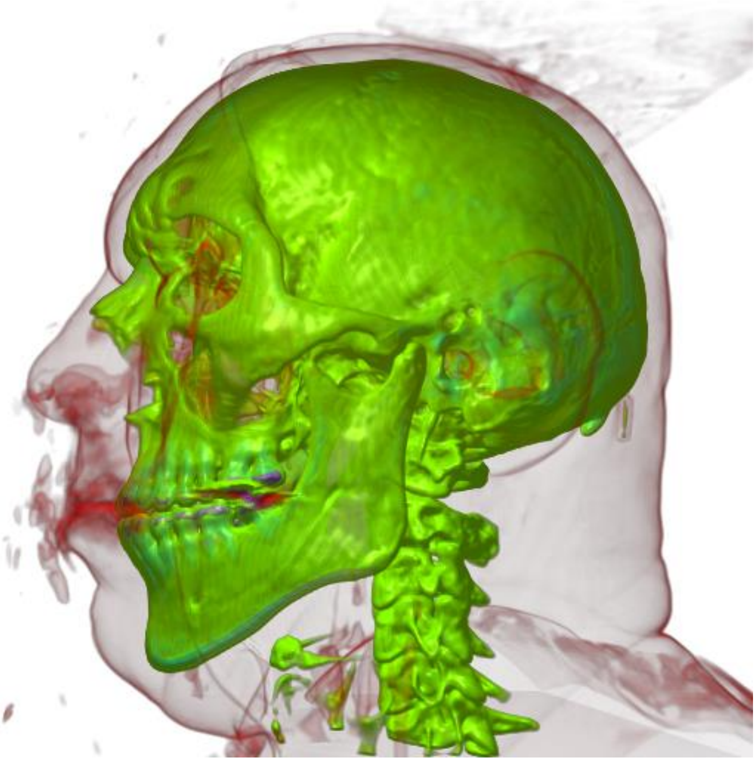
Each question is optional. Feel free to omit a response to any question; however the researcher would be grateful if all questions are responded to.

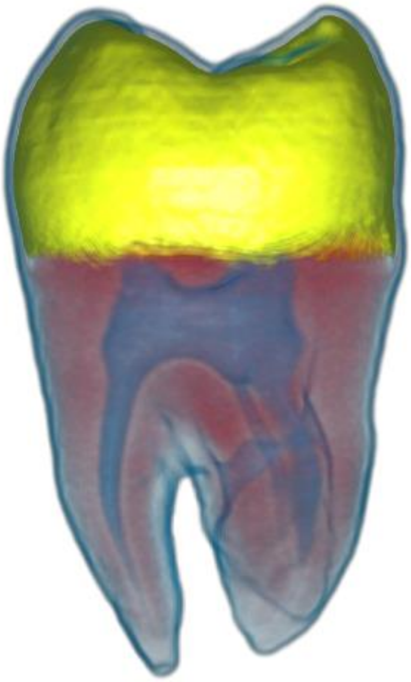
How clear and distinct is the green object in relation to the rest of the image?
(Please enter a score on a scale of 1-5 using the keyboard.)

1	2	3	4	5
Not at all		Neutral		Very clear

Sample Images from the Experiment

(Each image will be shown full screen for approximately 15 seconds after the above questions are presented.)





Bibliography

[1] T. G. Feeman, *The Mathematics of Medical Imaging: A Beginner's Guide*. Springer, Dec. 2009.

[2] M. Hadwiger, P. Ljung, C. R. Salama, and T. Ropinski, "Advanced illumination techniques for GPU-based volume raycasting," in *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, (New York, NY, USA), pp. 2:1–2:166, ACM, 2009.

[3] S. Roettger, "The volume library." `http://lgdv.cs.fau.de/External/vollib/`, January 2006. Accessed: 2013-01-20.

[4] G. Kindlmann, "Transfer functions in direct volume rendering: Design interface interaction," *SIGGRAPH Course Notes*, 2002.

[5] C. Correa and K.-L. Ma, "Visibility-driven transfer functions," in *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pp. 177–184, 2009.

[6] 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*, vol. 30, no. 7, pp. 2117–2126, 2011.

[7] A. König and M. E. Gröller, "Mastering transfer function specification by using VolumePro technology," Technical Report TR-186-2-00-07, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, Mar. 2000. human contact: technical-report@cg.tuwien.ac.at.

[8] A. E. Kaufman, "Volume visualization: Principles and advances," in *ACM SIGGRAPH Course Notes*, pp. 23–35, 1997.

[9] M. Levoy, "Display of surfaces from volume data," *IEEE Computer Graphics and Applications*, vol. 8, no. 3, pp. 29–37, 1988.

[10] J. Beyer, M. Hadwiger, and H. Pfister, "State-of-the-Art in GPU-Based Large-Scale Volume Visualization," *Computer Graphics Forum*, vol. 34, pp. 13–37, Dec. 2015.

[11] J. Kniss, G. Kindlmann, and C. Hansen, "Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets," in *Proceedings of the conference on Visualization '01*, VIS '01, (Washington, DC, USA), pp. 255–262, IEEE Computer Society, 2001.

[12] U. Bordoloi and H.-W. Shen, "View selection for volume rendering," in *IEEE Visualization*, pp. 487–494, 2005.

[13] G. Emsenhuber, *Visibility Histograms in Direct Volume Rendering.* Master's Thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, Nov. 2008.

[14] C. Wang and H.-W. Shen, "Information theory in scientific visualization," *Entropy*, vol. 13, pp. 254–273, Jan. 2011.

[15] M. Emami and L. L. Hoberock, "Selection of a best metric and evaluation of bottom-up visual saliency models," *Image and Vision Computing*, vol. 31, pp. 796–808, Oct. 2013.

[16] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, Nov. 1998.

[17] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency," in *Proceedings of Neural Information Processing Systems (NIPS)*, pp. 545–552, 2006.

[18] Y. Kim and A. Varshney, "Saliency-guided Enhancement for Volume Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 925–932, Sept. 2006.

[19] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Avila, K. Raghu, R. Machiraju, and J. Lee, "The transfer function bake-off," *IEEE Computer Graphics and Applications*, vol. 21, no. 3, pp. 16–22, 2001.

[20] S. Luo and J. Dingliana, "Information-guided transfer function refinement," in *Eurographics 2014 - Short Papers* (E. Galin and M. Wand, eds.), (Strasbourg, France), pp. 61–64, Eurographics Association, Apr. 2014.

[21] Shengzhou Luo and John Dingliana, "Transfer Function Refinement for Exploring Volume Data," in *Proceedings of Eurasia Graphics 2014: International Conference on Computer Graphics, Animation and Gaming Technologies*, (Ankara, Turkey), 2014.

[22] S. Luo and J. Dingliana, "Visibility-Weighted Saliency for Volume Visualization," in *Computer Graphics and Visual Computing (CGVC)* (R. Borgo and C. Turkay, eds.), (London, UK), The Eurographics Association, 2015.

[23] S. Luo and J. Dingliana, "Transfer Function Optimization Based on a Combined Model of Visibility and Saliency," in *EuroVis 2016 - Posters* (T. Isenberg and F. Sadlo, eds.), The Eurographics Association, 2016.

[24] S. Luo and J. Dingliana, "Selective Saturation and Brightness for Visualizing Time-Varying Volume Data," in *EuroVis 2015 - Posters*, (Cagliari, Italy), The Eurographics Association, 2015.

[25] A. Garcia, *Parallel time varying volume rendering on tile displays.* PhD thesis, Ohio State University, Columbus, OH, USA, 2006. AAI3197874.

[26] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3d Surface Construction Algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, (New York, NY, USA), pp. 163–169, ACM, 1987.

[27] L. Rosenblum, ed., *Scientific visualization: advances and challenges.* London: Academic Press, 1994.

[28] A. Corcoran, *Enhancing real-time focus and context direct volume rendering.* PhD Thesis, Trinity College Dublin, Dublin, Ireland, 2013.

[29] J. Meyer-Spradow, T. Ropinski, J. Mensmann, and K. Hinrichs, “Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations,” *IEEE Computer Graphics and Applications*, vol. 29, no. 6, pp. 6–13, 2009.

[30] K. Engel, “CERA-TVR: A framework for interactive high-quality teravoxel volume visualization on standard PCs,” in *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 123–124, Oct. 2011.

[31] T. Fogal, A. Schiewe, and J. Krüger, “An analysis of scalable GPU-based ray-guided volume rendering,” in *2013 IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, pp. 43–51, Oct. 2013.

[32] L. A. Westover, *Splatting: A Parallel, Feed-forward Volume Rendering Algorithm.* PhD thesis, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1991. UMI Order No. GAX92-08005.

[33] K. Mueller, T. Möller, and R. Crawfis, “Splatting Without the Blur,” in *Proceedings of the Conference on Visualization ’99: Celebrating Ten Years*, VIS ’99, (Los Alamitos, CA, USA), pp. 363–370, IEEE Computer Society Press, 1999.

[34] P. Lacroute and M. Levoy, “Fast Volume Rendering Using a Shear-warp Factorization of the Viewing Transformation,” in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’94, (New York, NY, USA), pp. 451–458, ACM, 1994.

[35] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl, “Interactive Volume on Standard PC Graphics Hardware Using Multi-textures and Multi-stage Rasterization,” in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware*, HWWS ’00, (New York, NY, USA), pp. 109–118, ACM, 2000.

[36] W. Hibbard and D. Santek, “Interactivity is the Key,” in *Proceedings of the 1989 Chapel Hill Workshop on Volume Visualization*, VVS ’89, (New York, NY, USA), pp. 39–43, ACM, 1989.

[37] R. Westermann and B. Sevenich, “Accelerated volume ray-casting using texture mapping,” in *Visualization, 2001. VIS ’01. Proceedings*, pp. 271–278, Oct. 2001.

[38] J. Krüger, "A New Sampling Scheme for Slice Based Volume Rendering," in *Proceedings of the 8th IEEE/EG International Conference on Volume Graphics*, VG'10, (Aire-la-Ville, Switzerland, Switzerland), pp. 1–4, Eurographics Association, 2010.

[39] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-time Volume Graphics*. Natick, MA, USA: A. K. Peters, Ltd., 2006.

[40] N. Svakhine, Y. Jang, D. Ebert, and K. Gaither, "Illustration and photography inspired visualization of flows and volumes," in *IEEE Visualization, 2005. VIS 05*, pp. 687–694, Oct. 2005.

[41] I. Viola, A. Kanitsar, and M. E. Gröller, "Importance-driven feature enhancement in volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, pp. 408–418, July 2005.

[42] N. A. Svakhine, D. S. Ebert, and W. M. Andrews, "Illustration-inspired depth enhanced volumetric medical visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 1, pp. 77–86, 2009.

[43] A. Stompel, E. B. Lum, and K.-L. Ma, "Visualization of multidimensional, multivariate volume data using hardware-accelerated non-photorealistic rendering techniques," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, PG '02, (Washington, DC, USA), p. 394, IEEE Computer Society, 2002.

[44] H. Hauser, L. Mroz, G. Italo Bischi, and E. Gröller, "Two-level volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, pp. 242–252, Sept. 2001.

[45] M. Hadwiger, C. Berger, and H. Hauser, "High-quality two-level volume rendering of segmented data sets on consumer graphics hardware," in *IEEE Visualization, 2003. VIS 2003*, pp. 301–308, Oct. 2003.

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

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

[48] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Groller, “Illustrative Context-Preserving Exploration of Volume Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 1559–1569, Nov. 2006.

[49] C.-K. Chen, R. Thomason, and K.-L. Ma, “Intelligent Focus+Context Volume Visualization,” in *Eighth International Conference on Intelligent Systems Design and Applications, 2008. ISDA ’08*, vol. 1, pp. 368–374, Nov. 2008.

[50] M. Burns, M. Haidacher, W. Wein, I. Viola, and M. E. Gröller, “Feature Emphasis and Contextual Cutaways for Multimodal Medical Visualization,” in *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS’07, (Aire-la-Ville, Switzerland, Switzerland), pp. 275–282, Eurographics Association, 2007.

[51] S. Sigg, R. Fuchs, R. Carnecky, and R. Peikert, “Intelligent cutaway illustrations,” in *Visualization Symposium (PacificVis), 2012 IEEE Pacific*, pp. 185–192, Feb. 2012.

[52] S. Bruckner and M. Groller, “Exploded Views for Volume Data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 1077–1084, Sept. 2006.

[53] J. Díaz, E. Monclús, I. Navazo, and P. Vázquez, “Adaptive Cross-sections of Anatomical Models,” *Computer Graphics Forum*, vol. 31, pp. 2155–2164, Sept. 2012.

[54] J. Kniss, G. Kindlmann, and C. Hansen, “Multidimensional transfer functions for interactive volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 270–285, July 2002.

[55] I. W. O. Serlie, F. Vos, R. Truyen, F. Post, and L. van Vliet, “Classifying CT image data into material fractions by a scale and rotation invariant edge model,” *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2891–2904, 2007.

[56] I. Serlie, R. Truyen, J. Florie, F. Post, L. v. Vliet, and F. Vos, "Computed Cleansing for Virtual Colonoscopy Using a Three-Material Transition Model," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003* (R. E. Ellis and T. M. Peters, eds.), no. 2879 in Lecture Notes in Computer Science, pp. 175–183, Springer Berlin Heidelberg, Jan. 2003.

[57] G. Kindlmann and J. Durkin, "Semi-automatic generation of transfer functions for direct volume rendering," in *IEEE Symposium on Volume Visualization, 1998*, pp. 79–86, 1998.

[58] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma, "A Novel Interface for Higher-Dimensional Classification of Volume Data," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, (Washington, DC, USA), pp. 66–, IEEE Computer Society, 2003.

[59] F.-Y. Tzeng and K.-L. Ma, "A cluster-space visual interface for arbitrary dimensional classification of volume data," in *Proceedings of the Sixth Joint Eurographics - IEEE TCVG conference on Visualization*, VISSYM'04, (Aire-la-Ville, Switzerland, Switzerland), pp. 17–24, Eurographics Association, 2004.

[60] S. Arens and G. Domik, "A survey of transfer functions suitable for volume rendering," in *Proceedings of the 8th IEEE/EG international conference on Volume Graphics*, VG'10, (Aire-la-Ville, Switzerland, Switzerland), pp. 77–83, Eurographics Association, 2010.

[61] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber, "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*, SIGGRAPH '97, (New York, NY, USA), pp. 389–400, ACM Press/Addison-Wesley Publishing Co., 1997.

[62] Y. Wu and H. Qu, "Interactive transfer function design based on editing direct volume rendered images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 1027–1040, 2007.

[63] C. D. Correa and K.-L. Ma, “Visibility histograms and visibility-driven transfer functions,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 2, pp. 192–204, 2011.

[64] M. Haidacher, S. Bruckner, A. Kanitsar, and M. E. Gröller, “Information-based transfer functions for multimodal visualization,” in *Proceedings of the First Eurographics conference on Visual Computing for Biomedicine*, EG VCBM’08, (Aire-la-Ville, Switzerland, Switzerland), pp. 101–108, Eurographics Association, 2008.

[65] S. Bruckner and T. Möller, “Isosurface Similarity Maps,” *Computer Graphics Forum*, vol. 29, pp. 773–782, June 2010.

[66] M. Ruiz, A. Bardera, I. Boada, I. Viola, M. Feixas, and M. Sbert, “Automatic transfer functions based on informational divergence,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1932–1941, 2011.

[67] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert, “Information theory-based automatic multimodal transfer function design,” *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 870–880, 2013.

[68] L. Zhou, M. Schott, and C. Hansen, “Transfer function combinations,” *Computers & Graphics*, vol. 36, pp. 596–606, Oct. 2012.

[69] S. Bruckner and M. E. Gröller, “Style transfer functions for illustrative volume rendering,” *Computer Graphics Forum*, vol. 26, no. 3, pp. 715–724, 2007.

[70] S. Lindholm, P. Ljung, C. Lundstrom, A. Persson, and A. Ynnerman, “Spatial Conditioning of Transfer Functions Using Local Material Distributions,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1301–1310, 2010.

[71] C. Rezk-Salama and A. Kolb, “Opacity peeling for direct volume rendering,” *Computer Graphics Forum*, vol. 25, no. 3, pp. 597–606, 2006.

[72] R. Maciejewski, I. Woo, W. Chen, and D. Ebert, “Structuring feature space: A non-parametric method for volumetric transfer function generation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1473–1480, 2009.

[73] C. Bajaj, P.-T. Bremer, and D. Schikore, "The contour spectrum," in *Visualization '97., Proceedings*, pp. 167–173, 1997.

[74] J. Kniss, S. premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun, "Gaussian transfer functions for multi-field volume visualization," in *IEEE Visualization, 2003. VIS 2003*, pp. 497–504, Oct. 2003.

[75] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, (Washington, DC, USA), pp. 513–520, IEEE Computer Society, 2003.

[76] P. Šereda, A. Bartrolí, I. Serlie, and F. Gerritsen, "Visualization of boundaries in volumetric data sets using LH histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 208–218, Apr. 2006.

[77] P. Šereda, A. Vilanova, and F. A. Gerritsen, "Automating Transfer Function Design for Volume Rendering Using Hierarchical Clustering of Material Boundaries," in *Proceedings of the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization*, EUROVIS'06, (Aire-la-Ville, Switzerland, Switzerland), pp. 243–250, Eurographics Association, 2006.

[78] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller, "Volume Visualization based on Statistical Transfer-Function Spaces," in *Proceedings of the IEEE Pacific Visualization 2010*, pp. 17–24, Mar. 2010.

[79] Y. Wang, J. Zhang, D. J. Lehmann, H. Theisel, and X. Chi, "Automating Transfer Function Design with Valley Cell-Based Clustering of 2d Density Plots," *Computer Graphics Forum*, vol. 31, pp. 1295–1304, June 2012.

[80] C. Y. Ip, A. Varshney, and J. Jaja, "Hierarchical exploration of volumes using multilevel segmentation of the intensity-gradient histograms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2355–2363, 2012.

[81] S. Roettger, M. Bauer, and M. Stamminger, "Spatialized transfer functions," in *Proceedings of the Seventh Joint Eurographics / IEEE VGTC conference on*

Visualization, EUROVIS'05, (Aire-la-Ville, Switzerland, Switzerland), pp. 271–278, Eurographics Association, 2005.

[82] A. Tappenbeck, B. Preim, and V. Dicken, "Distance-based transfer function design: Specification methods and applications," in *SimVis*, pp. 259–274, 2006.

[83] C. Correa and K.-L. Ma, "Size-based transfer functions: A new volume exploration technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1380–1387, Dec. 2008.

[84] J. Caban and P. Rheingans, "Texture-based transfer functions for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1364–1371, Dec. 2008.

[85] M. Alper Selver, "Exploring Brushlet Based 3d Textures in Transfer Function Specification for Direct Volume Rendering of Abdominal Organs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, pp. 174–187, Feb. 2015.

[86] X. Zhao and A. Kaufman, "Multi-dimensional reduction and transfer function design using parallel coordinates," in *Proceedings of the 8th IEEE/EG international conference on Volume Graphics*, VG'10, (Aire-la-Ville, Switzerland, Switzerland), pp. 69–76, Eurographics Association, 2010.

[87] H. Guo, H. Xiao, and X. Yuan, "Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates," in *Visualization Symposium (PacificVis), 2011 IEEE Pacific*, pp. 19–26, Mar. 2011.

[88] H. S. Kim, J. P. Schulze, A. C. Cone, G. E. Sosinsky, and M. E. Martone, "Dimensionality Reduction on Multi-Dimensional Transfer Functions for Multi-Channel Volume Data Sets," *Information visualization*, vol. 9, pp. 167–180, Sept. 2010.

[89] E. Zudilova-Seinstra, T. Adriaansen, and R. v. Liere, *Trends in Interactive Visualization: State-of-the-Art Survey.* Springer Publishing Company, Incorporated, 1 ed., 2008.

[90] T. He, L. Hong, A. Kaufman, and H. Pfister, “Generation of transfer functions with stochastic search techniques,” in *Visualization ’96. Proceedings.*, pp. 227–234, Oct. 1996.

[91] C. Rezk-Salama, P. Hastreiter, J. Scherer, and G. Greiner, “Automatic adjustment of transfer functions for 3d volume visualization,” in *In Proc. Workshop Vision, Modeling, and Visualization (VMV*, pp. 357–364, a, 2000.

[92] M.-Y. Chan, Y. Wu, W.-H. Mak, W. Chen, and H. Qu, “Perception-based transparency optimization for direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1283 –1290, Dec. 2009.

[93] J. Zhou and M. Takatsuka, “Automatic transfer function generation using contour tree controlled residue flow model and color harmonics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1481–1488, 2009.

[94] M. Alper Selver and C. Guzelis, “Semiautomatic Transfer Function Initialization for Abdominal Visualization Using Self-Generating Hierarchical Radial Basis Function Networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 395–409, 2009.

[95] G. Lathen, S. Lindholm, R. Lenz, A. Persson, and M. Borga, “Automatic Tuning of Spatially Varying Transfer Functions for Blood Vessel Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 2345–2354, Dec. 2012.

[96] R. Maciejewski, Y. Jang, I. Woo, H. Jänicke, K. Gaither, and D. Ebert, “Abstracting attribute space for transfer function exploration and design,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 94–107, Jan. 2013.

[97] I. Viola, A. Kanitsar, and M. E. Groller, “Importance-Driven Volume Rendering,” in *Proceedings of the conference on Visualization ’04*, VIS ’04, (Washington, DC, USA), pp. 139–146, IEEE Computer Society, 2004.

[98] B. Preim and C. P. Botha, *Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2nd ed., 2013.

[99] B. T. Polyak, “The conjugate gradient method in extremal problems,” *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[100] L. Cai, W.-L. Tay, B. P. Nguyen, C.-K. Chui, and S.-H. Ong, “Automatic transfer function design for medical visualization using visibility distributions and projective color mapping,” *Computerized Medical Imaging and Graphics*, vol. 37, pp. 450–458, Oct. 2013.

[101] S. Marchesin, J.-M. Dischler, and C. Mongenet, “Per-Pixel Opacity Modulation for Feature Enhancement in Volume Rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 560–570, July 2010.

[102] E. Bronstad, J. Asen, H. Torp, and G. Kiss, “Visibility driven visualization of 3d cardiac ultrasound data on the GPU,” in *Ultrasonics Symposium (IUS), 2012 IEEE International*, pp. 2651–2654, Oct. 2012.

[103] Y. Jung, J. Kim, and D. Feng, “Dual-modal visibility metrics for interactive PET-CT visualization,” in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2696–2699, Aug. 2012.

[104] Y. Jung, J. Kim, S. Eberl, M. Fulham, and D. D. Feng, “Visibility-driven PET-CT visualisation with region of interest (ROI) segmentation,” *The Visual Computer*, vol. 29, pp. 805–815, June 2013.

[105] L. Zheng, C. Correa, and K.-L. Ma, “Visibility guided multimodal volume visualization,” in *2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 297–304, Dec. 2013.

[106] P. Schlegel and R. Pajarola, “Visibility-difference entropy for automatic transfer function generation,” vol. 8654, pp. 865406–865406–15, 2013.

[107] H. Qin, B. Ye, and R. He, “The voxel visibility model: An efficient framework for transfer function design,” *Computerized Medical Imaging and Graphics*, vol. 40, pp. 138–146, Mar. 2015.

[108] B. Wilson, E. B. Lum, and K.-L. Ma, “Interactive Multi-volume Visualization,” in *Computational Science âĂŤ ICCS 2002* (P. M. A. Sloot, A. G. Hoekstra, C. J. K. Tan, and J. J. Dongarra, eds.), no. 2330 in Lecture Notes in Computer Science, pp. 102–110, Springer Berlin Heidelberg, Apr. 2002. DOI: 10.1007/3-540-46080-2_11.

[109] J. Kniss, C. Hansen, M. Grenier, and T. Robinson, “Volume rendering multivariate data to visualize meteorological simulations: A case study,” in *Proceedings of the Symposium on Data Visualisation 2002*, VISSYM ’02, (Aire-la-Ville, Switzerland, Switzerland), pp. 189–ff, Eurographics Association, 2002.

[110] H. Akiba, N. Fout, and K.-L. Ma, “Simultaneous classification of time-varying volume data based on the time histogram,” in *Proceedings of the Eighth Joint Eurographics / IEEE VGTC conference on Visualization*, EUROVIS’06, (Aire-la-Ville, Switzerland, Switzerland), pp. 171–178, Eurographics Association, 2006.

[111] J. Woodring and H.-W. Shen, “Multi-variate, Time Varying, and Comparative Visualization with Contextual Cues,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 909–916, 2006.

[112] C. Wang, H. Yu, and K.-L. Ma, “Importance-driven time-varying data visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1547 –1554, Dec. 2008.

[113] T.-Y. Lee and H.-W. Shen, “Visualizing time-varying features with TAC-based distance fields,” in *Visualization Symposium, 2009. PacificVis ’09. IEEE Pacific*, pp. 1–8, Apr. 2009.

[114] T.-Y. Lee and H.-W. Shen, “Visualization and exploration of temporal trend relationships in multivariate time-varying data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1359–1366, Dec. 2009.

[115] R. Khlebnikov, B. Kainz, M. Steinberger, and D. Schmalstieg, "Noise-Based Volume Rendering for the Visualization of Multivariate Volumetric Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2926–2935, 2013.

[116] R. Khlebnikov, B. Kainz, M. Steinberger, M. Streit, and D. Schmalstieg, "Procedural Texture Synthesis for Zoom-Independent Visualization of Multivariate Data," *Computer Graphics Forum*, vol. 31, pp. 1355–1364, June 2012.

[117] H. Akiba, K.-L. Ma, J. H. Chen, and E. R. Hawkes, "Visualizing Multivariate Volume Data from Turbulent Combustion Simulations," *Computing in Science Engineering*, vol. 9, pp. 76 –83, Apr. 2007.

[118] H. Guo, H. Xiao, and X. Yuan, "Scalable Multivariate Volume Visualization and Analysis Based on Dimension Projection and Parallel Coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 1397–1410, Sept. 2012.

[119] S. Liu, B. Wang, J. Thiagarajan, P.-T. Bremer, and V. Pascucci, "Multivariate volume visualization through dynamic projections," in *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 35–42, Nov. 2014.

[120] J. Kehrer and H. Hauser, "Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 495–513, Mar. 2013.

[121] T. J. Jankun-Kelly and K.-L. Ma, "A study of transfer function generation for time-varying volume data," in *Proceedings of the 2001 Eurographics conference on Volume Graphics*, VG'01, (Aire-la-Ville, Switzerland, Switzerland), pp. 51–66, Eurographics Association, 2001.

[122] J. Woodring, C. Wang, and H.-W. Shen, "High dimensional direct rendering of time-varying volumetric data," in *IEEE Visualization, 2003. VIS 2003*, pp. 417 –424, Oct. 2003.

[123] J. Woodring and H.-W. Shen, "Chronovolumes: a direct rendering technique for visualizing time-varying data," in *Proceedings of the 2003 Eurographics/IEEE*

TVCG Workshop on Volume graphics, VG '03, (New York, NY, USA), pp. 27–34, ACM, 2003.

[124] A. Tikhonova, C. D. Correa, and K.-L. Ma, "An Exploratory Technique for Coherent Visualization of Time-varying Volume Data," *Computer Graphics Forum*, vol. 29, pp. 783–792, June 2010.

[125] Z. Fang, T. Möller, G. Hamarneh, and A. Celler, "Visualization and exploration of time-varying medical image data sets," in *Proceedings of Graphics Interface 2007*, GI '07, (New York, NY, USA), pp. 281–288, ACM, 2007.

[126] J. Woodring and H.-W. Shen, "Multiscale time activity data exploration via temporal clustering visualization spreadsheet," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 123–137, Feb. 2009.

[127] J. Woodring and H.-W. Shen, "Semi-automatic time-series transfer functions via temporal clustering and sequencing," *Computer Graphics Forum*, vol. 28, no. 3, pp. 791–798, 2009.

[128] M. O. Ward and Z. Guo, "Visual Exploration of Time-Series Data with Shape Space Projections," *Computer Graphics Forum*, vol. 30, pp. 701–710, June 2011.

[129] Y. Gu and C. Wang, "TransGraph: Hierarchical Exploration of Transition Relationships in Time-Varying Volumetric Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2015–2024, Dec. 2011.

[130] Y. Peng, J. Dong, L. Chen, H. Chu, and J. Yong, "An Optimal Color Mapping Strategy Based on Energy Minimization for Time-Varying Data," in *2011 12th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*, pp. 411–417, 2011.

[131] P. Haeberli, "Paint by numbers: abstract image representations," in *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '90, (New York, NY, USA), pp. 207–214, ACM, 1990.

[132] C. G. Healey, L. Tateosian, J. T. Enns, and M. Remple, "Perceptually based brush strokes for nonphotorealistic visualization," *ACM Trans. Graph.*, vol. 23, pp. 64–96, Jan. 2004.

[133] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser, “Illustrative flow visualization: State of the art, trends and challenges,” in *EuroGraphics 2012 State of the Art Reports (STARs)*, pp. 75–94, 2012.

[134] P. Rheingans and D. Ebert, “Volume illustration: nonphotorealistic rendering of volume models,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 3, pp. 253–264, 2001.

[135] A. Joshi and P. Rheingans, “Illustration-inspired techniques for visualizing time-varying data,” in *IEEE Visualization, 2005. VIS 05*, pp. 679 – 686, Oct. 2005.

[136] A. Joshi and P. Rheingans, “Evaluation of illustration-inspired techniques for time-varying data visualization,” *Computer Graphics Forum*, vol. 27, no. 3, pp. 999–1006, 2008.

[137] A. Joshi, J. Caban, P. Rheingans, and L. Sparling, “Case study on visualizing hurricanes using illustration-inspired techniques,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 709 –718, Oct. 2009.

[138] A. Kuhn, D. J. Lehmann, R. Gaststeiger, M. Neugebauer, B. Preim, and H. Theisel, “A clustering-based visualization technique to emphasize meaningful regions of vector fields,” in *Proc. of Vision, Modeling, and Visualization (VMV 2011)*, pp. 191–198, Eurographics Assosciation, 2011.

[139] J. Ma, C. Wang, and C.-K. Shene, “Coherent view-dependent streamline selection for importance-driven flow visualization,” vol. 8654, pp. 865407–865407–15, 2013.

[140] P. Bénard, A. Bousseau, and J. Thollot, “State-of-the-Art Report on Temporal Coherence for Stylized Animations,” *Computer Graphics Forum*, vol. 30, pp. 2367–2386, Dec. 2011.

[141] C.-K. Chen, S. Yan, H. Yu, N. Max, and K.-L. Ma, “An Illustrative Visualization Framework for 3d Vector Fields,” *Computer Graphics Forum*, vol. 30, pp. 1941–1951, Sept. 2011.

[142] T. Annen, H. Theisel, C. Rössl, G. Ziegler, and H.-P. Seidel, “Vector Field Contours,” in *Proceedings of Graphics Interface 2008*, GI ’08, (Toronto, Ont., Canada, Canada), pp. 97–105, Canadian Information Processing Society, 2008.

[143] C. Rezk-Salama, P. Hastreiter, C. Teitzel, and T. Ertl, “Interactive Exploration of Volume Line Integral Convolution Based on 3d-texture Mapping,” in *Proceedings of the Conference on Visualization ’99: Celebrating Ten Years*, VIS ’99, (Los Alamitos, CA, USA), pp. 233–240, IEEE Computer Society Press, 1999.

[144] C. Muelder and K.-L. Ma, “Interactive feature extraction and tracking by utilizing region coherency,” in *Visualization Symposium, 2009. PacificVis ’09. IEEE Pacific*, pp. 17–24, 2009.

[145] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, “The State of the Art in Flow Visualisation: Feature Extraction and Tracking,” *Computer Graphics Forum*, vol. 22, pp. 775–792, Dec. 2003.

[146] C. Wang and K.-L. Ma, “Information and knowledge assisted analysis and visualization of large-scale data,” in *Workshop on Ultrascale Visualization, 2008. UltraVis 2008*, pp. 1–8, 2008.

[147] K.-L. Ma, “Machine learning to boost the next generation of visualization technology,” *IEEE Computer Graphics and Applications*, vol. 27, no. 5, pp. 6–9, 2007.

[148] F.-Y. Tzeng and K.-L. Ma, “Intelligent Feature Extraction and Tracking for Visualizing Large-Scale 4d Flow Simulations,” in *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, SC ’05, (Washington, DC, USA), pp. 6–, IEEE Computer Society, 2005.

[149] J. Caban, A. Joshi, and P. Rheingans, “Texture-based feature tracking for effective time-varying data visualization,” *IEEE transactions on visualization and computer graphics*, vol. 13, pp. 1472–1479, Dec. 2007. PMID: 17968099.

[150] Y. Gu and C. Wang, “TransGraph: hierarchical exploration of transition relationships in time-varying volumetric data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 2015 –2024, Dec. 2011.

[151] T.-J. Hsieh, Y.-S. Yang, J.-H. Wang, and W.-J. Shen, “Feature extraction using bionic particle swarm tracing for transfer function design in direct volume rendering,” *The Visual Computer*, vol. 30, pp. 33–44, Feb. 2013.

[152] C. Johnson and J. Huang, "Distribution-Driven Visualization of Volume Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 734 –746, Oct. 2009.

[153] L.-L. Cai, B. P. Nguyen, C.-K. Chui, and S.-H. Ong, "Rule-Enhanced Transfer Function Generation for Medical Volume Visualization," *Computer Graphics Forum*, vol. 34, pp. 121–130, June 2015.

[154] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, Oct. 1948.

[155] M. Sbert, M. Feixas, J. Rigau, M. Chover, and I. Viola, "Information Theory Tools for Computer Graphics," *Synthesis Lectures on Computer Graphics and Animation*, vol. 4, pp. 1–153, Jan. 2009.

[156] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, "A feature-driven approach to locating optimal viewpoints for volume visualization," in *IEEE Visualization, 2005. VIS 05*, pp. 495–502, 2005.

[157] M. Feixas, M. Sbert, and F. González, "A Unified Information-theoretic Framework for Viewpoint Selection and Mesh Saliency," *ACM Trans. Appl. Percept.*, vol. 6, pp. 1:1–1:23, Feb. 2009.

[158] L. Xu, T.-Y. Lee, and H.-W. Shen, "An Information-Theoretic Framework for Flow Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1216–1224, Nov. 2010.

[159] T.-Y. Lee, O. Mishchenko, H.-W. Shen, and R. Crawfis, "View point evaluation and streamline filtering for flow visualization," in *2011 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 83–90, Mar. 2011.

[160] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, and M. Sbert, "Multimodal Data Fusion Based on Mutual Information," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1574–1587, 2012.

[161] C. Wang and H.-W. Shen, "LOD Map - A Visual Interface for Navigating Multiresolution Volume Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 1029 –1036, Oct. 2006.

[162] M. Ruiz, I. Boada, M. Feixas, and M. Sbert, "Viewpoint information channel for illustrative volume rendering," *Computers & Graphics*, vol. 34, pp. 351–360, Aug. 2010.

[163] M. Chen and H. Jäenicke, "An Information-theoretic Framework for Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, pp. 1206–1215, Nov. 2010.

[164] Q. Zhao and C. Koch, "Learning saliency-based visual attention: A review," *Signal Processing*, vol. 93, pp. 1401–1407, June 2013.

[165] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vision Research*, vol. 42, pp. 107–123, Jan. 2002.

[166] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: A Bayesian inference theory of attention," *Vision Research*, vol. 50, pp. 2233–2247, Oct. 2010.

[167] V. Mahadevan and N. Vasconcelos, "Spatiotemporal Saliency in Dynamic Scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 171–177, Jan. 2010.

[168] L. Duan, C. Wu, J. Miao, and A. Bovik, "Visual conspicuity index: Spatial dissimilarity, distance, and central bias," *Signal Processing Letters, IEEE*, vol. 18, pp. 690–693, Nov 2011.

[169] H. Jänicke and M. Chen, "A Salience-based Quality Metric for Visualization," *Computer Graphics Forum*, vol. 29, pp. 1183–1192, June 2010.

[170] W. Kim, C. Jung, and C. Kim, "Spatiotemporal Saliency Detection and Its Applications in Static and Dynamic Scenes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 446–456, Apr. 2011.

[171] W. Kim, C. Jung, and C. Kim, "Saliency detection: A self-ordinal resemblance approach," in *2010 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1260–1265, July 2010.

[172] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh Saliency," in *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, (New York, NY, USA), pp. 659–666, ACM, 2005.

[173] E. Shen, S. Li, X. Cai, L. Zeng, and W. Wang, "SAVE: saliency-assisted volume exploration," *Journal of Visualization*, pp. 1–11, Oct. 2014.

[174] E. Shen, Y. Wang, and S. Li, "Spatiotemporal volume saliency," *Journal of Visualization*, pp. 1–12, Apr. 2015.

[175] J. Giesen, K. Mueller, E. Schuberth, L. Wang, and P. Zolliker, "Conjoint Analysis to Measure the Perceived Quality in Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1664–1671, Nov. 2007.

[176] E. W. Anderson, "Evaluating visualization using cognitive measures," in *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, (New York, NY, USA), pp. 5:1–5:4, ACM, 2012.

[177] R. K. Christopher, C. G. Healey, V. Interrante, D. H. Laidlaw, and C. Ware, "Thoughts on user studies: Why, how, and when," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 20–25, 2003.

[178] N. Redmond, *Influencing user perception using real-time adaptive abstraction.* PhD Thesis, Trinity College Dublin, Dublin, Ireland, 2010.

[179] D. Laidlaw, R. Kirby, J. Davidson, T. Miller, M. da Silva, W. Warren, and M. Tarr, "Quantitative comparative evaluation of 2D vector field visualization methods," in *Visualization, 2001. VIS '01. Proceedings*, pp. 143–150, 2001.

[180] A. Lu, R. Maciejewski, and D. S. Ebert, "Volume Composition and Evaluation Using Eye-tracking Data," *ACM Trans. Appl. Percept.*, vol. 7, pp. 4:1–4:20, Jan. 2010.

[181] M. Kersten-Oertel, S.-S. Chen, and D. Collins, "An Evaluation of Depth Enhancing Perceptual Cues for Vascular Volume Visualization in Neurosurgery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 391–403, Mar. 2014.

[182] J. Díaz, T. Ropinski, I. Navazo, E. Gobbetti, and P. Vazquez, "Perceptual effects of volumetric shading models in stereoscopic desktop-based environments," in *Computer Graphics International*, pp. 1–10, 2015.

[183] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson, "A Model for Volume Lighting and Modeling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, pp. 150–162, Apr. 2003.

[184] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert, "An Information-Theoretic Observation Channel for Volume Visualization," *Computer Graphics Forum*, vol. 32, no. 3pt4, pp. 411–420, 2013.

[185] K.-L. Ma and D. Camp, "High performance visualization of time-varying volume data over a wide-area network," in *Supercomputing, ACM/IEEE 2000 Conference*, pp. 29–29, Nov 2000.

[186] K.-L. Ma, "Time-varying data repository." `http://www.cs.ucdavis.edu/~ma/ITR/`, October 2003. Accessed: 2013-02-10.

[187] S. E. Palmer, *Vision Science: Photons to Phenomenology.* MIT Press, 1999.

[188] M. D. Fairchild, *Color Appearance Models.* John Wiley & Sons, June 2013.

[189] D. Shepard, "A Two-dimensional Interpolation Function for Irregularly-spaced Data," in *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, (New York, NY, USA), pp. 517–524, ACM, 1968.

[190] D. Cunningham and C. Wallraven, *Experimental Design: From User Studies to Psychophysics.* Natick, MA, USA: A. K. Peters, Ltd., 1st ed., 2011.

[191] E. G. Spedicato, *Algorithms for Continuous Optimization: The State of the Art.* Springer Science & Business Media, Dec. 2012.

[192] P.-H. Phua, W. Fan, and Y. Zeng, "Parallel Algorithms for Large-scale Nonlinear Optimization," *International Transactions in Operational Research*, vol. 5, pp. 67–77, Jan. 1998.

[193] X.-q. Yang, *Optimization Methods and Applications.* Springer Science & Business Media, Apr. 2001.

[194] T. C. Peachey, D. Abramson, and A. Lewis, "Parallel line search," in *Optimization* (C. Pearce and E. Hunt, eds.), no. 32 in Springer Optimization and Its Applications, pp. 369–381, Springer New York, 2009. DOI: 10.1007/978-0-387-98096-6_20.

[195] J.-S. Praßni, "Voreen data sets." `http://www.uni-muenster.de/Voreen/download/workspaces_and_data_sets.html`, December 2013. Accessed: 2013-12-20.

[196] E. K. P. Chong and S. H. Zak, *An introduction to optimization.* Wiley series in discrete mathematics and optimization, Hoboken, New Jersey: Wiley, fourth edition ed., 2013.

[197] Y.-x. Yuan, "Step-sizes for the gradient method," *AMS IP Studies in Advanced Mathematics*, vol. 42, no. 2, p. 785, 2008.

[198] M. N. Vrahatis, G. S. Androulakis, J. N. Lambrinos, and G. D. Magoulas, "A class of gradient unconstrained minimization algorithms with adaptive stepsize," *Journal of Computational and Applied Mathematics*, vol. 114, pp. 367–386, Feb. 2000.

[199] B. Zhou, L. Gao, and Y.-H. Dai, "Gradient Methods with Adaptive Step-Sizes," *Computational Optimization and Applications*, vol. 35, pp. 69–86, Mar. 2006.

[200] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, April 2004.

[201] P. Hastreiter and T. Ertl, "Integrated registration and visualization of medical image data," in *Computer Graphics International, 1998. Proceedings*, pp. 78–85, June 1998.

[202] R. Fedkiw, J. Stam, and H. W. Jensen, "Visual Simulation of Smoke," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, (New York, NY, USA), pp. 15–22, ACM, 2001.