Habilitationsschrift

# Visual Exploration and Analysis of Volumetric Data

ausgeführt in den Jahren 2008-2012
zum Zwecke der Erlangung der *venia docendi* (Lehrbefugnis)
im Habilitationsfach ”Praktische Informatik”

eingereicht im April 2012
an der Technischen Universität Wien
Fakultät für Informatik

von

Dipl.-Ing. Dr.techn. Stefan Bruckner
Flesichmanngasse 7/1B, A-1040 Wien
geboren am 2. Juni 1980 in Oberwart
`bruckner@cg.tuwien.ac.at`

Wien, im April 2012

# Kurzfassung

Der Einsatz von Informationstechnologie hat zu einem rapiden Anstieg der Datenmengen in Bereichen wie der Biologie, der Medizin, der Klimaforschung und in den Ingineurswissenschaften geführt. In vielen Fällen sind diese Daten volumetrisch, d.h. sie beschreiben die räumliche Verteilung von einer oder mehreren Quantitäten. Volumensvisualisierung ist jener Forschungsbereich, der sich mit der Transformation von solchen Daten zu Bildern beschäftigt, um so beispielsweise das Verstehen von Struktur oder die Identifikation von besonderen Merkmalen zu erleichtern. In der vorliegenden Arbeit werden Ansätze präsentiert, die diesen Transformationsprozess untersützten indem sie die interaktive Darstellung, Analyse und Exploration von volumentrischen Daten erleichtern.

# Abstract

Information technology has led to a rapid increase in the amount of data that arise in areas such as biology, medicine, climate science, and engineering. In many cases, these data are volumetric in nature, i.e., they describe the distribution of one or several quantities over a region in space. Volume visualization is the field of research which investigates the transformation of such data sets into images for purposes such as understanding structure or identifying features. This thesis presents work to aid this process by improving the interactive depiction, analysis, and exploration of volumetric data.

# Contents

**Preface** vii

**1 Introduction** **1**
- 1.1 Background 1
- 1.2 Selected Publications 7
- 1.3 Contributions 7

**2 Instant Volume Visualization using Maximum Intensity Difference Accumulation** **15**
- 2.1 Introduction 15
- 2.2 Background 16
- 2.3 Related Work 17
- 2.4 Maximum Intensity Difference Accumulation 19
- 2.5 Combining DVR, MIP, and MIDA 22
- 2.6 Results and Discussion 23
- 2.7 Conclusions 25

**3 Isosurface Similarity Maps** **27**
- 3.1 Introduction 27
- 3.2 Related Work 28
- 3.3 Isosurface Similarity 29
- 3.4 Applications 34
- 3.5 Implementation 39
- 3.6 Discussion 39
- 3.7 Conclusions 42

**4 Volume Analysis Using Multimodal Surface Similarity** **45**
- 4.1 Introduction 45
- 4.2 Related Work 46
- 4.3 Multimodal Volume Data 49
- 4.4 Multimodal Surface Similarity 51
- 4.5 Similarity-Based Multimodal Volume Visualization 55

4.6 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
4.7 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . 67
4.8 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
4.9 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

**5 BrainGazer – Visual Queries for Neurobiology Research 73**
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
5.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . 74
5.3 System Overview . . . . . . . . . . . . . . . . . . . . . . . . 76
5.4 Visualization . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
5.5 Visual Queries . . . . . . . . . . . . . . . . . . . . . . . . . . 83
5.6 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . 87
5.7 Results and Discussion . . . . . . . . . . . . . . . . . . . . . 88
5.8 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90

**6 Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design 93**
6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
6.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . 94
6.3 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
6.4 Sampling and Clustering . . . . . . . . . . . . . . . . . . . . 98
6.5 Interactive Exploration . . . . . . . . . . . . . . . . . . . . . 103
6.6 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . 109
6.7 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
6.8 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114

**Bibliography 117**

An expert is a man who has made all the mistakes, which can be made, in a very narrow field.

— *Niels Bohr*

# Preface

THIS thesis presents selected parts of my research carried out at the Vienna University of Technology, Austria and the Simon Fraser University, Canada in the years from 2008 to 2012.

First and foremost, I want to thank Meister Eduard Gröller for his mentorship and guidance over many years. Not only has he provided me with complete freedom to pursue my research interests, but he has encouraged and supported me in the perusal of my academic career in every imaginable respect. His devotion to scientific integrity and hard work, coupled with his ingenious leadership skills, have made him a role model for me and his sense of humor has brightened up even stressful times. I further thank Werner Purgathofer, the head of the Institute of Computer Graphics and Algorithms, for his constant efforts in providing ideal working conditions and a genuinely enjoyable environment.

My gratitude also goes to all my former and present colleagues, collaborators, and students for their support. In particular, I would like to thank Martin Haidacher, Armin Kanitsar, Peter Kohlmann, Torsten Möller, Daniel Patel, Peter Rautek, and Ivan Viola for giving me the pleasure to work with them.

Finally, I want to express my deepest love and gratitude to my girlfriend Petra. Without her patience and support, I certainly would have not accomplished this.

Vienna, Austria, April 2012 *Stefan Bruckner*

A professor is one who can speak on any subject – for precisely fifty minutes.

— *Norbert Wiener*

# CHAPTER 1 Introduction

This chapter serves as a brief introduction into the area of volume visualization, its application fields, and its challenges with respect to the topic of this thesis. Furthermore, it gives an overview of the author's scientific contributions to this area and provides additional background information on the selection of articles contained in this thesis.

## 1.1 Background

OUR society is confronted with rapidly growing amounts of scientific data that arise in areas such as biology, medicine, climate science, and engineering. The process of scientific knowledge discovery can be characterized by the interaction between three distinct spaces: the parameter space, the model space, and the data space.

The data space constitutes the observable characteristics of the phenomenon under investigation, while the parameter space represents the set of input conditions. The model space is the set of possible relationships between parameter space and data space. In science, the goal is to uncover the physical laws, biological processes, or mathematical equations which transform a point in parameter space to the observable characteristics in data space, i.e., to identify a particular model which describes the relationship between the two spaces. Very generally speaking, the knowledge discovery processes can be seen as a continuous interaction of the following basic steps:

**Exploration.** When little knowledge about the model space is available, it is frequently desirable to first characterize the data space, i.e., investigate the possible variations achievable by varying a set of input parameters.

**Analysis.** Having acquired a sizeable set of data points, researchers can now proceed to seek the underlying relationships between parameter space and data space by careful analysis of the available data.

**Modeling.** Based on the results of the analysis, a model of the phenomena involved can be developed. The development of a model is guided by the insights gained during analysis of the acquired data as well as external factors such as known laws or relationships.

**Validation.** Given the availability of a model, the characteristics of this model are investigated. Predictions of the model can now be compared with the characteristics of newly acquired data points. This phase guides the further refinement or extension of the model.

To support these tasks it is essential to provide effective and efficient means for forming a mental model of data space and parameter space in order to gain a better understanding of their relationships. Interactive visualization acts as a high-throughput channel for gaining insight into these spaces by transforming information into a visual form. Effective means for performing this transformation, however, are particularly challenging to obtain for volumetric data which is important in many disciplines.

A volumetric data set represents the distribution of one or more quantities over a certain region of space. Most commonly, volume data are given as samples on a regular grid, but many other types of lattices are employed as well. Depending on the application, these samples can represent actual measured data or be the result of a computational simulation. While volume data are generated and used in many diverse fields, they play a particularly important role in the following areas:

**Medicine.** In modern medicine huge amounts of volumetric data are generated on a daily basis. Medical imaging data acquired using X-Ray Computed Tomography (CT), Magnetic Resonance Imaging (MRI), Position Emission Tomography (PET), 3D Ultrasonography, and many other methods play an important role in diagnosis and treatment planning. Initiatives towards the creation of an electronic health record, a systematic collection of electronic health information about patients or populations that transcends the boundaries of individual institutions, are ongoing. However, there is a widening gap between data collection and data comprehension. As more and more medical procedures employ imaging as a preferred diagnostic tool, advanced methods to quickly identify and analyze potential pathologies are necessary.

**Biology.** A main source of volume data in biology are microscopy techniques such as Laser-Scanning Confocal Microscopy (LSCM) or Transmission Electron Microscopy (TEM). These methods enable imaging of organisms at a very high level of detail. In neurobiology, for example, such techniques are used to define the cellular architecture of the brain. Mapping out the fine anatomy of complex neuronal circuits is an essential first step in investigating the neural mechanisms of information processing. Advances in imaging technology have triggered efforts of building large databases which collect information of anatomy and physiology over large populations of individuals. The complexity and sheer amount of these data, however, necessitate effective visualization and interaction techniques for generating, refining, and analyzing such atlases.

**Earth Sciences.** Exploratory data analysis, as advocated by Tukey [153], has led to a virtual explosion in scientific data in the earth sciences. The goal is to rapidly identify promising hypotheses that are afterwards checked in an analytical, confirmative process (e.g., through statistical models). Large-scale measurements of dynamic processes as well as numerical modeling and computational simulation result in huge amounts of spatial data that need to be analyzed. Simulation data from global climate models, for example, consist of multiple time series of volumetric data which represent different variables of interest. A major goal in the visualization of such data is to enable scientists to make sense of complex phenomena, both qualitatively and quantitatively.

**Engineering.** Simulations based on computational fluid dynamics and finite element methods are a mainstay in many engineering disciplines and an important application for volume visualization. Furthermore, non-destructive testing benefits from visualization in conjunction with new imaging technology. Novel ways of testing provide unique insight into complex components, which allow precise, fast, and inexpensive characterizations. Thus, especially in the preproduction phase of a new component these new technologies significantly reduce the design costs, development time, as well as time to market. In consequence, the fast return on investment of new developments stimulates research activity in the field of analysis and visualization.

The research presented in this thesis covers general approaches for improving the visual exploration and analysis of volume data as well as their application to specific domains. The subsequent sections provide a brief overview of the state-of-the-art and the challenges involved in the visualization of volumetric data.

### 1.1.1 Volume Rendering

Over the years many techniques have been developed to visualize volumetric data. Since methods for displaying geometric primitives were already well-established, most of the early methods involve approximating a surface contained within the data using geometric primitives. Such algorithms fit geometric primitives such as polygons to constant-value contour surfaces in volumetric data sets. After extracting this intermediate representation, hardware-accelerated rendering can be used to display the surface primitives. Popular surface extraction algorithms such as marching cubes [98] are commonly used to generate models based on specific features in volumetric data.

However, in general, these methods need to make a decision for every data sample whether or not the surface passes through it. This can produce false positives (spurious surfaces) or false negatives (erroneous holes in surfaces), particularly in the presence of small or poorly defined features. As information about the interior of objects is generally not retained, a fundamental drawback of these methods is that one dimension of information is essentially lost.

In response to this, volume rendering techniques were developed that attempt to capture the entire 3D data in a single 2D image. These methods convey more information than surface rendering methods, but at the cost of increased algorithmic complexity, and consequently increased rendering times. To improve interactivity in volume rendering, many optimization methods as well as special-purpose volume rendering hardware have been developed. Instead of extracting an intermediate representation, volume rendering provides a method for directly displaying the volumetric data. Samples of the volumetric function are mapped to optical properties along viewing rays passing through the volume. Volume rendering comprises more information in a single image than traditional surface representations and is thus a valuable tool for the exploration and analysis of data. Over the last two decades, many researchers have worked on various related problems in direct volume rendering. Also, tremendous efforts have been put into improving efficiency of volume rendering and current graphics hardware enables the interactive visualization of even large data sets.

Two of the most commonly used volume rendering techniques are Maximum Intensity Projection (MIP) and Direct Volume Rendering (DVR). MIP [161] is based on the simple assumption that the highest data values represent important features and consequently only depicts the maximum along each viewing ray. DVR [41, 96], on the other hand, accumulates the contributions of multiple samples along a ray. Both approaches have distinct advantages and most systems for the visualization of volume data allow users to switch between these techniques. More advanced rendering techniques simulate physical light transport within the volume to various degrees of accuracy and include effects such as shadows [160] and scattering [87]. Furthermore, substantial work on the simulation of artistic techniques and styles has been performed. However, as this is not the main topic of the thesis, it will not be explored further.

### 1.1.2 Visual Analysis

In the visual analysis of volume data, the fact that a three-dimensional data set needs to be represented in a 2D view means that it is essential to provide means for quickly identifying important features, and, conversely, that non-essential information should be discarded.

A common approach in the design of visualization systems is to provide a set of parameters that control the visualization algorithm specifically tailored to a given type of data and task. The data attributes are then mapped to these visualization parameters. One of the most commonly used visualization mapping methods in volume visualization are transfer functions. A transfer function maps data attributes of a 3D volume to visual properties such as color and opacity. A mapping from more than one attribute is called a multidimensional transfer function [86]. Multidimensional transfer functions can be useful to discriminate between different features. Many different types of derived attributes have been investigated, for instance gradient magnitude [83], curvature [69, 84], and statistical properties [58].

Transfer functions are a powerful tool to achieve various visualizations, but their specification is a complex task, particularly in the multidimensional case. Typically, one- or multidimensional histograms are employed to provide additional, albeit limited, guidance. The user frequently needs expert knowledge about the underlying rendering technique to achieve the desired results. A further problem of transfer functions is that they typically only take into account local properties such as the original data values and their derivatives, but do not make use of global information. For this reason, topology-based methods [27] which exploit additional structural information have been investigated. However, they frequently suffer from noise and acquisition artifacts present in many real-world data sets. One approach to simplify the visualization mapping process is to loosen the coupling between data attributes and visual representation. By assigning semantics to both data characteristics and visualization parameters, meaningful rules for the mapping between them can be defined [129, 130].

Systems tailored towards specific applications frequently employ a combination of multiple spatial and abstract views [133]. By linking selections in these views, users can infer the relationships between spatial features and data attributes. A significant challenge in this area of research is to find computational techniques which can provide a concise overview of the data and can assist the users in identifying characteristics and features according to the goals of their investigation.

### 1.1.3 Parameter Space Exploration

The availability of high-throughput techniques in imaging and scientific computing enables experiments on a large scale, generating vast sets of data. Even though the results of each experiment may be visualized separately, this is no longer feasible for databases consisting of thousands of such experiments. This development means that it is no longer sufficient to provide tools for analyzing a single data set. Instead, many thousands of data points, each consisting of a volumetric representation, need to be investigated. Such volumetric data spaces not only require new approaches to data management and transfer, but also necessitate novel navigation, interaction, and visualization techniques. In particular, it is crucial to provide intuitive and efficient facilities to visually explore, query, and retrieve data items, as well as methods to categorize and abstract the space.

Areas such as climate research and engineering, for instance, increasingly make use of multi-run simulations which are performed to study the variability of a simulation model and to understand the model's sensitivity to certain control parameters [79]. The goals of such a sensitivity analysis include the identification of model parameters that require additional research, the determination of control parameters that are strongly correlated with the simulation output, or finding insignificant parameters that can be eliminated from the model. In medicine and biology, atlases are a common way to spatially organize data [162]. The atlas serves as reference frame for comparing and integrating image data from different experiments by spatially relating collections of drawings, microscopic images, or text. Such atlases are an invaluable reference in efforts to compile a comprehensive set of anatomical and functional data. Many existing tools, however, only provide very simple navigation tools such as text-based search and hierarchical lists.

The visual exploration of such parameter spaces is a major challenge in visualization research. Existing approaches mainly focus on abstract data spaces where each single point consists of only few attribute values. More than ten numerical attributes per data point, for example, are considered challenging high-dimensional data. In such scenarios, visualization methods such as scatterplot matrices or parallel coordinates can be very effective, as the primary goal is to identify patterns - a single data point alone generally does not provide significant information [175]. In spatial data spaces, on the other hand, a single point may consist of millions of data values and their spatial arrangement is essential for their interpretation.

### 1.1.4 Summary

The research presented in this thesis covers aspects of all areas mentioned in the previous subsections. However, many important topics in scientific visualization were not included in the brief overview. Instead, the focus was put on providing background information which helps to interpret the remainder of the thesis in a wider context.

Chapter 2 presents work which aims to unify the design space of volume rendering techniques. By fusing two of the most common methods in volume visualization, their distinct advantages can be combined. Chapters 3 and 4 are devoted to improving the visual analysis of volumetric data sets by deriving global structural information. They introduce the notion of isosurface similarity and show how this new concept can be used as a means of providing guidance in feature selection. The exploration of parameter spaces is covered in Chapters 5 and 6. Chapter 5 presents an approach for navigating in neuroanatomical atlases which comprise volumetric imaging data, geometrical models, and semantic information. An intuitive visual query interface integrates these different types of data. A different application scenario is discussed in Chapter 6 which investigates the interactive exploration of variations in multi-run simulation data based on spatiotemporal cluster analysis.

## 1.2 Selected Publications

This thesis contains the following papers [19–21, 23, 59]:

1. S. Bruckner and M. E. Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum*, 28(3):775–782, 2009.

2. S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010.

3. M. Haidacher, S. Bruckner, and M. E. Gröller. Volume analysis using multimodal surface similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, 2011.

4. S. Bruckner, V. Šoltészová, M.E. Gröller, J. Hladůvka, K. Bühler, J. Y. Yu, and B. J. Dickson. BrainGazer – visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1497–1504, 2009.

5. S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.

The papers included in this thesis appear unmodified in their original, published form, except for the typesetting, which has been adapted to conform to the style of this thesis. No textual changes were performed. The bibliography sections were joined into a single bibliography at the end of this thesis.

## 1.3 Contributions

This section gives a short overview on each of the publications contained in this thesis. The papers represent a sample of the research work the author carried out in the years from 2008 to 2012. For this time, he also lists 17 additional refereed scientific articles in his publication list. In total, the author has co-authored 39 papers at the time of writing.

The publications selected for inclusion in this thesis focus on the author's work on developing insight into complex spatial phenomena represented by volumetric data. Other contributions of the author to related areas such as illustrative visualization, realistic volume illumination, transfer functions, and interaction, would have been beyond the topical scope of this thesis.

In addition to publishing individual articles at the world's leading forums for visualization research, the author's research contribution have also received recognition by international organizations. In 2011, he was awarded the *Eurographics Young Researcher Award* by the *European Association for Computer Graphics* for his work in scientific visualization. This award is annually given to young researchers in the field who have already made a significant contribution.

As visualization research is a highly collaborative discipline, where cooperations between diploma- and PhD students, senior researchers, and domain experts are required to solve complex problems, none of the papers in this thesis is a single-author paper. Such papers are the exception, not the rule, in the field of visualization. However, the author substantially contributed to each of the included publications, as evidenced by his first-author position on all but one of them. In all cases the author's contributions include development of the initial idea, significant parts of the implementation, as well as writing of the article, and generation of results.

### 1.3.1 Chapter 2: Instant Volume Visualization using Maximum Intensity Difference Accumulation

Two of the most commonly employed techniques for investigating volumetric data are Direct Volume Rendering (DVR) and Maximum Intensity Projection (MIP). DVR employs a physically-motivated absorption-plus-emission optical model and frequently utilizes gradient-based shading to emphasize surface structures. The basis of MIP, on the other hand, is the assumption that the most relevant structures for the investigation at hand have higher intensity values. In practice, this is achieved through special scanning protocols or the administration of contrast agents.

Common examples include CT and MRI angiography as well as Positron Emission Tomography (PET), but also different microscopic imaging modalities where structures of interest are frequently highlighted using fluorescent marker proteins. One of the biggest advantages of MIP over DVR is that it does not require the specification of complex transfer functions to generate good visualization results. A major disadvantage is, however, that due to the order-independency of the maximum operator, spatial context is lost.

This chapter presents a novel approach for volume visualization which combines the complementary advantages of DVR and MIP. A key contribution of this work is that it enables a seamless transition between two previously distinct techniques which are both in wide-spread practical use – the result is a surprisingly simple superset of both methods which can be readily integrated into existing systems and therefore has considerable practical value.

The author developed the idea and implemented the method. He produced all the results and wrote the article with feedback by Eduard Gröller. The paper was presented at the *Eurographics/IEEE Symposium on Visualization (EuroVis) 2009*, a leading venue for international visualization research. The EuroVis proceedings are published as a special issue of the journal *Computer Graphics Forum*.

### 1.3.2 Chapter 3: Isosurface Similarity Maps

Volumetric data enables physicians, scientists, and engineers to investigate the interior of complex objects. However, providing clear visualizations of the structures contained in a volume data set is a major challenge. One of the issues is the lack of explicit geometric information and limited semantics. A volume data set contains a large number of isosurfaces at different target scalar field values, while its structure is typically characterized by a finite number of feature isosurfaces that segment the data set into several important components.

Irrespective of the chosen visualization method, providing guidance in the identification of salient isovalues plays an important role in improving the exploration process. Histogram-based visualizations are a common tool for providing such guidance. However, they only depict information about the frequency of data values while disregarding the spatial relationships between their corresponding isosurfaces.

The chapter describes a new way for analyzing volumetric data sets based on an information-theoretic measure of similarity between the level sets, or isosurfaces, of a scalar field. Instead of the frequency of individual data values and/or derived attributes, this new approach is based on the global notion of isosurface similarity. By employing mutual information, a measure of statistical dependence, between all combinations of isosurfaces their degree of similarity can be characterized. This process results in an isosurface similarity map which not only provides a compact overview of the similarities and differences within a data set, but also serves as the basis for several additional applications such as the automatic identification of relevant isovalues in a scalar field.

The author developed the initial idea, the theoretical formulation, and the complete implementation of the described approach. He wrote the article with feedback by Torsten Möller and generated all results. The paper was presented at the *Eurographics/IEEE Symposium on Visualization (EuroVis) 2010*, where it received the *Best Paper Award*. It was published in the journal *Computer Graphics Forum*.

### 1.3.3 Chapter 4: Volume Analysis Using Multimodal Surface Similarity

Imaging modalities have different advantages and disadvantages typically related to the physical principles they use to scan a specimen. They may suffer from different kinds of artifacts, can be differently affected by noise, may be able to distinguish different materials or tissues, and can have differences with respect to contrast and resolution. In order to gain insight into the phenomenon under investigation, it is essential to integrate this information effectively.

For this purpose, this chapter presents an extension of the concept of isosurface similarity to multimodal data. Instead of analyzing a single data set, the joint information provided by multiple modalities is considered. This enables the identification of similarities and differences between imaging techniques and allows the exploitation of their complementary advantages for improved visualization. The work shows that multimodal surface similarity can be used to guide the visual exploration process in an intuitive manner. By performing classification directly based on similarity, complex multi-dimensional transfer functions can be avoided.

The author contributed the initial idea and concept. He supervised the parts of the implementation performed by Martin Haidacher and contributed significant portions, including the similarity-space approach for classification, himself. The author also wrote the core of the paper. Results were generated together with Martin Haidacher and feedback was provided by Eduard Gröller. The paper was presented at the *IEEE Visualization 2011* conference, the world's premier venue for visualization research. The proceedings of this conference are published as a special issue of the journal *IEEE Transactions on Visualization and Computer Graphics*.

### 1.3.4 Chapter 5: BrainGazer – Visual Queries for Neurobiology Research

In neuroscience, mapping the fine anatomy of complex neuronal circuits is an essential first step in investigating the neural mechanisms of information processing. For this purpose, researchers compile atlases, i.e., databases that combine imaging data with additional information in order to ultimately uncover the inner workings of the brain.

Visualization is an important part of such projects. Proper visualization tools are indispensable for quality control (e.g., identification of acquisition artifacts and misclassifications), the sharing of generated resources among a network of collaborators, or the setup and validation of an automated analysis pipeline.

This chapter is devoted to the development of a system for exploring neuroanatomical atlases consisting of large amounts of volume and geometric data. Traditional database interfaces do not provide sufficient means to retrieve and interact with data items based on spatial relationships. The main focus of the presented work was to enable visual queries which allow domain experts to interact with such atlases based on their actual research questions. A further challenge was to develop scalable techniques, as the amount of data in such application scenarios is constantly growing.

In this inter-disciplinary collaboration, the author developed the initial concept and architecture of the described system. It is based on domain-expert requirements formulated by Barry Dickson, scientfic director of the Institute of Molecular Pathology, Vienna, and Jai Yu, a PhD student of his at the time of publication. The system design was refined together with Katja Bühler, senior researcher at the VRVis research center. The author contributed substantial parts of the implementation, including the visual query mechanism and the employed visualization techniques. He guided and supervised the development of the remaining components, which were implemented by the diploma student Veronika Šoltészová and Jiří Hladůvka, a collaborating researcher at the VRVis research center. All results were generated by the author, and the article was written by him with feedback by Katja Bühler and Eduard Gröller. The paper was presented at the *IEEE Visualization 2009* conference and published in the journal *IEEE Transactions on Visualization and Computer Graphics*.

### 1.3.5 Chapter 6: Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design

Computational models of physical phenomena typically have many parameters that interact in a non-trivial manner. Therefore, in order to investigate the behavior of such models, the simulation is often repeated multiple times with varied settings of the control parameters. The resulting data (which is often referred to as an ensemble simulation) is a collection of values which co-exists for the same data attribute at each spacetime location.

In the analysis, the data is often aggregated, for example by computing statistical properties with respect to all simulation runs. However, additional insight could be gained by analyzing the source data to extract interesting patterns and trends that occur in different runs, to investigate how many of the runs exhibit a certain pattern, or to study correlations between input and output parameters.

This chapter presents an approach for result-driven exploration of physically-based multi-run simulations. In particular, the work focuses on applications in the motion picture industry, where computational fluid dynamics simulations are frequently used to generate realistic visual effects. Each volumetric time sequence is first split into temporally similar segments and thereafter grouped across different runs using a density-based clustering algorithm. The results of this process are then explored in an interactive graphical environment. The presented pipeline allows users to quickly find parameter settings which result in the desired spatio-temporal simulation behavior and to explore their variations.

The author developed the initial idea and and refined it in discussions with Torsten Möller. He performed the complete implementation and wrote the article. The protocol of the user study was developed by the author and the study was conducted together with Torsten Möller. The paper was presented at the *IEEE Visualization 2010* conference and published in the journal *IEEE Transactions on Visualization and Computer Graphics*.

**The following chapter was originally published as:**

S. Bruckner and M. E. Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum*, 28(3):775–782, 2009.

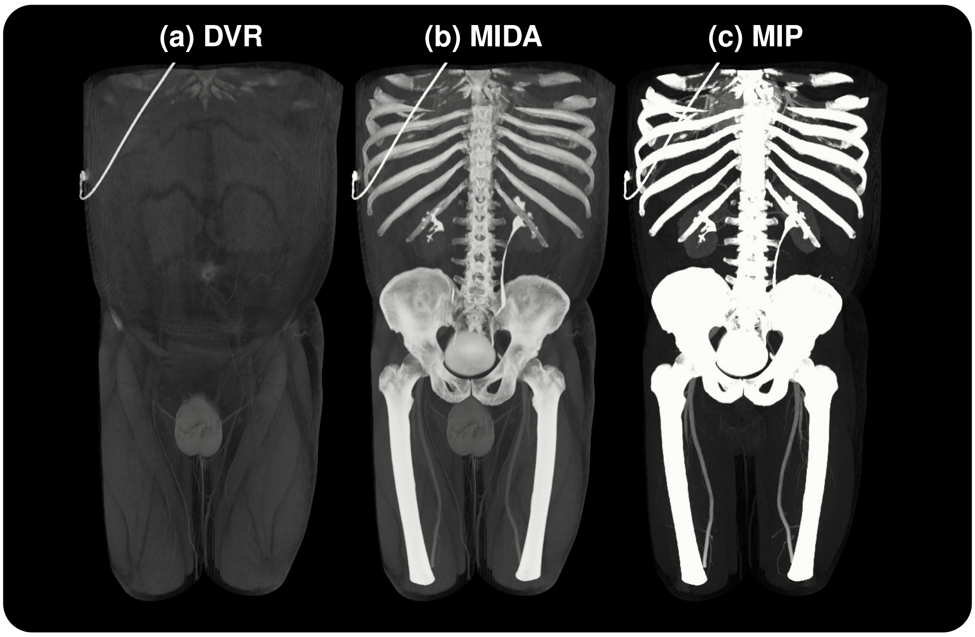


**Figure 2.1 –** Full body CT angiography rendered using (a) DVR, (b) MIDA, and (c) MIP. Data set courtesy of the OsiriX Foundation (http://www.osirix-viewer.com).

Science may be described as the art of
systematic oversimplification.
— *Karl Popper*

CHAPTER

# 2

# Instant Volume Visualization using Maximum Intensity Difference Accumulation

It has long been recognized that transfer function setup for Direct Volume Rendering (DVR) is crucial to its usability. However, the task of finding an appropriate transfer function is complex and time-consuming even for experts. Thus, in many practical applications simpler techniques which do not rely on complex transfer functions are employed. One common example is Maximum Intensity Projection (MIP) which depicts the maximum value along each viewing ray. In this paper, we introduce Maximum Intensity Difference Accumulation (MIDA), a new approach which combines the advantages of DVR and MIP. Like MIP, MIDA exploits common data characteristics and hence does not require complex transfer functions to generate good visualization results. It does, however, feature occlusion and shape cues similar to DVR. Furthermore, we show that MIDA – in addition to being a useful technique in its own right – can be used to smoothly transition between DVR and MIP in an intuitive manner. MIDA can be easily implemented using volume raycasting and achieves real-time performance on current graphics hardware.

## 2.1 Introduction

DIRECT Volume Rendering (DVR) and Maximum Intensity Projection (MIP) are two of the most common methods for the visualization of volumetric data. DVR employs a physically-motivated absorption-plus-emission optical model and frequently utilizes gradient-based shading to emphasize surface structures. The basis of MIP, on the other hand, is the assumption that the most relevant structures for the investigation at hand have higher intensity values. In practice, this is achieved through special scanning protocols or the administration of contrast agents. Common examples include CT and MRI angiography as well as Positron Emission Tomography (PET), but also different microscopic imaging modalities where structures of interest are frequently highlighted using fluorescent marker proteins.

One of the biggest advantages of MIP over DVR is that it does not require the specification of complex transfer functions to generate good visualization results. A major disadvantage is, however, that due to the order-independency of the maximum operator, spatial context is lost. This paper introduces a novel method which aims to combine the advantages of DVR and MIP. Our new approach is able to generate meaningful visualizations using a class of very simple linear transfer functions specified using standard window/level controls.

The remainder of this paper is structured as follows: In Section 2.2 we discuss the foundations of DVR and MIP and identify the drawbacks of these common methods. Section 2.3 reviews related work. Our novel rendering technique is detailed in Section 2.4. In Section 2.5 we present an approach to smoothly transition between DVR and MIP using our new method as an intermediate step. Results are presented and discussed in Section 2.6. The paper is concluded in Section 2.7.

## 2.2 Background

MIP works by traversing all viewing rays and finding the maximum data value along each of them. This maximum is then mapped to a color value and displayed to the user. In most cases this mapping process is simply a linear transformation of data values to pixel intensity. As only a single value is displayed along each ray, MIP images lack depth information which can lead to visual ambiguities. Two approaches have been proposed to eliminate this drawback: Depth-Shaded Maximum Intensity Projection (DMIP) and Local Maximum Intensity Projection (LMIP). In DMIP [67], the data value is additionally weighted by a depth-dependent term. This reduces the likelihood of high data values being projected if they are located far away from the image plane. While this approach can help to regain spatial context, it may also hide certain high-intensity regions. For LMIP [137], the first local maximum which is above a user-defined threshold is depicted. If no value above the threshold is found along a ray, the global maximum along the ray is used for the pixel. This approach also adds spatial information at the cost of introducing an additional parameter which can greatly affect the visualization.

DVR commonly employs a simplified model of light propagation in participating media [110]. It only accounts for emission and absorption of light but neglects scattering effects. Emission and absorption properties are specified using a transfer function which assigns color and opacity to each data value. The final color along each viewing ray is then determined through accumulation of colors and opacities at constant intervals – an approximative solution of the volume rendering integral. In order to enhance the appearance of surface structures in the volume, the vector of first partial derivatives along the three major axes – the gradient – can be used to evaluate a surface shading model. The color at each sample point along a ray is additionally modulated by the result of this evaluation. Surface-based shading generally helps to enhance fine details in areas with high contrast, but can lead to artifacts in

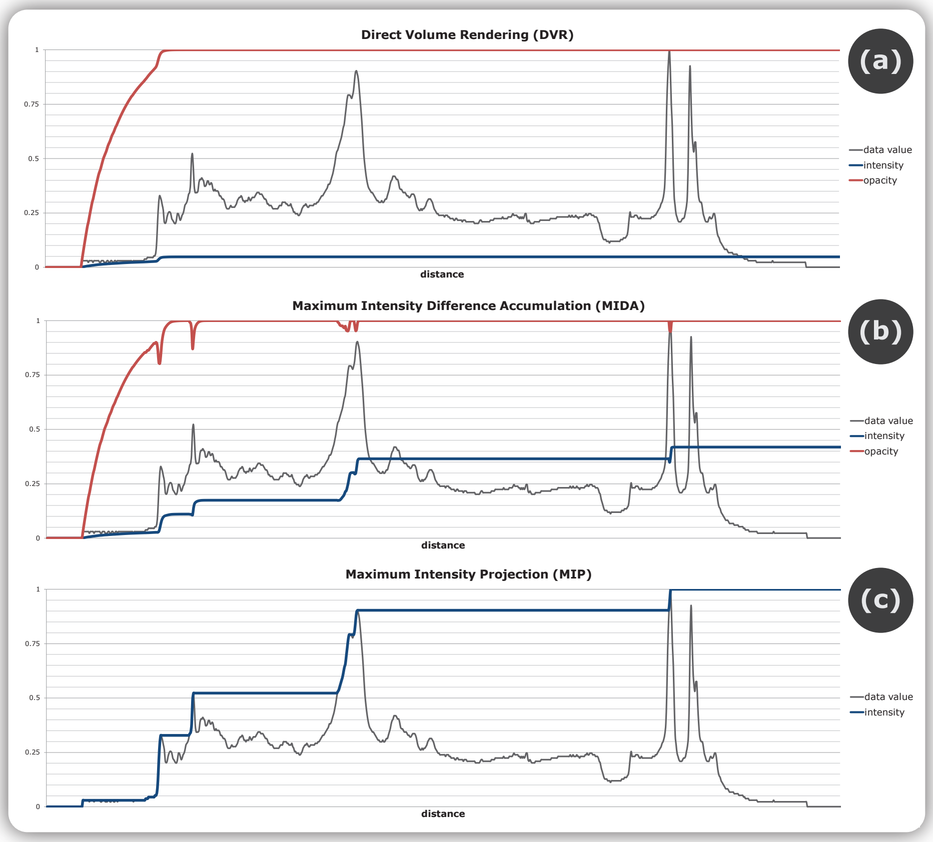


**Figure 2.2 –** Typical ray profiles for (a) DVR, (b) MIDA, and (c) MIP.

nearly homogeneous areas. A major problem of DVR is the specification of an appropriate transfer function since assigning high opacity to a certain data range may occlude other structures of interest [125]. Thus, MIP is often the method of choice as it does not require additional parameter tuning even if DVR could lead to additional insight. In particular, shading information in DVR can help to interpret certain structures such as blood vessels.

In this paper, we propose a new rendering technique which aims to fuse the complementary advantages of DVR and MIP. Specifically, we want to preserve the practically parameterless nature of MIP and combine it with the added spatial context of DVR provided by accumulation and shading.

## 2.3 Related Work

DVR was introduced by Drebin et al. [41] and Levoy [96]. Since then, much work has focused on the task of improving the specification of transfer functions. Kniss et al. [86] use a two-dimensional transfer function based on scalar value and gradient magnitude to effectively extract specific material boundaries and convey subtle surface properties. Correa and Ma [31] propose the use of transfer functions based on the the relative size of features to improve classification. In order to cope with the complexity of transfer function specification, several automatic and semi-automatic approaches for their generation have been proposed. Bergman et al. [13] present an interactive approach for guiding the process of colormap selection. The contour spectrum, introduced by Bajaj et al. [9], assists iso-value selection by presenting the user with 2D plots of several properties computed over the data range. He et al. [66] treat the search for a good transfer function as an optimization problem and employ stochastic techniques for this purpose. Marks et al. [108] sample the vast parameter space to find a set of input-parameter vectors that optimally disperses the output-value vectors, organizing the resulting graphics for easy and intuitive browsing by the user. Kindlmann and Durkin [83] employ histogram volumes of the data value and its first and second directional derivatives along the gradient direction to find a transfer function which makes boundaries more visible. Tzeng et al. [155] propose the use of machine learning to find an optimal classification for user-specified regions of interest defined using a painting metaphor. The drawback of automatic approaches is that they require substantial pre-processing which limits their use in interactive settings where a quick exploration of the data is required.

Other approaches rely on properties of the data acquisition process to quickly generate meaningful visualizations. MIP, first introduced by Wallis et al. [161] as Maximum Activity Projection for Positron Emission Tomography (PET) data, emphasizes high-intensity data values. In addition to several optimization techniques [116] it has been attempted to re-introduce missing spatial context information to MIP. Heidrich et al. [67] proposed to use a depth-based weighting of the data values along a ray. Sato et al. [137] project the first local maximum instead of the global maximum. Hauser et al. [64] fuse DVR and MIP rendering for a single data set. Based on a pre-classification different compositing strategies are used for structures in a data set. Similarily, Straka et al. [148] employ a combination of rendering techniques for the visualization of vascular structures. Mora and Ebert [115] experiment with different order-independent operators such as maximum-projection and summation for the generation of overview images.

The area of illustrative visualization has investigated techniques to enhance visual comprehension based on concepts common in traditional illustration. Rheingans and Ebert [132] present several illustrative techniques which enhance features and add depth and orientation cues. They also propose to locally apply these methods for regional enhancement. Csébfalvi et al. [34] present a non-photorealistic technique to quickly generate contour-based overview images. Viola et al. [157] introduce importance-driven volume rendering which generates cut-away views based on the importance of preclassified objects. Bruckner et al. [22] present an illustrative volume rendering technique inspired by ghosted views. Rezk-Salama and Kolb [131] introduce opacity peeling for the extraction of feature layers to enable the visualization of structures which are difficult to classify using transfer functions. Malik et al. [106] extend this work with a more detailed analysis of ray profiles. Locally adaptive volume rendering, presented by Marchesin et al. [107], attempts to reduce occlusion by dynamically adapting the opacity of sample contributions.

## 2.4 Maximum Intensity Difference Accumulation

The basic idea behind our approach is to alter the accumulation behavior of DVR to exhibit characteristics similar to MIP. Conventionally in DVR, when tracing viewing rays starting from the eye, the accumulated opacity is a monotonically growing function. This means that structures located behind thick non-transparent regions tend to have less influence on the final image. Because of this, a simple transfer function, such as a linear ramp, frequently causes structures of interest to be immersed in "fog", i.e., they are occluded by irrelevant material. While more complex transfer functions can be employed to remedy this problem, their specification is significantly more time-consuming. MIP, on the other hand, completely disregards such occlusion relationships. We want to adapt the behavior of DVR to prevent local maxima from becoming completely occluded while still preserving opacity-based accumulation.

We assume a continuous scalar-valued volumetric function $f(P)$ of normalized data values in the range $[0,1]$. At the $i$-th sample location $P_i$ along a viewing ray, $f_{P_i}$ denotes the data value at location $P_i$ and $f_{max_i}$ is the current maximum value along the ray. Front-to-back traversal is used. We use $c(f_{P_i})$ and $\alpha(f_{P_i})$ to denote the color and opacity, respectively, of the sample value as classified by a transfer function. The accumulated color and opacity at the $i$-th sample position along the ray are denoted by $c_i$ and $\alpha_i$, $c_0$ and $\alpha_0$ are initialized to zero.

We are interested in regions where the maximum along the ray changes. Specifically, when the maximum changes from a low to a high value, the corresponding sample should have more influence on the final image compared to the case where the difference is only small. We use $\delta_i$ to classify this change at every sample location:

$$\delta_i = \begin{cases} f_{P_i} - f_{max_i} & \text{if } f_{P_i} > f_{max_i} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

Whenever a new maximum is encountered while traversing the ray, $\delta_i$ is nonzero. These are the cases where we want to override occlusion relationships. For this purpose, the previously accumulated color $c_{i-1}$ and opacity $\alpha_{i-1}$ are weighted by a factor $\beta_i = 1 - \delta_i$. For our new model, which we term

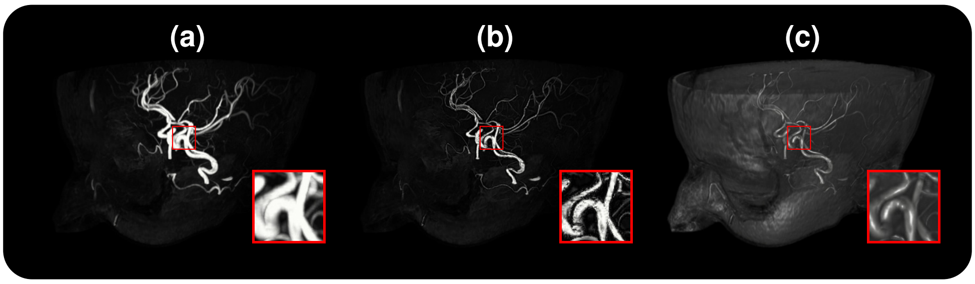


**Figure 2.3 –** Cranial MRI angiography rendered using (a) MIP without shading, (b) MIP with gradient-based shading, and (c) MIDA with gradient-based shading.

Maximum Intensity Difference Accumulation (MIDA), the accumulated color $c_i$ and opacity $\alpha_i$ for the $i$-th sample are then:

$$\begin{aligned} c_i &= \beta_i c_{i-1} + (1 - \beta_i \alpha_{i-1}) \alpha(f_{P_i}) c(f_{P_i}) \\ \alpha_i &= \beta_i \alpha_{i-1} + (1 - \beta_i \alpha_{i-1}) \alpha(f_{P_i}) \end{aligned} \tag{2.2}$$

Equation 2.2 only differs from standard DVR compositing by the additional weighting of $c_{i-1}$ and $\alpha_{i-1}$ with $\beta_i$. One way to interpret the modulation of previously accumulated color and opacity performed in MIDA is a particular importance function which assigns highest prominence to local maxima. In particular, maxima which occur in a rather discontinuous manner cause more modulation than smoothly increasing ray profiles.

Figure 2.1 shows images generated using (a) DVR, (b) MIDA, and (c) MIP. Figure 2.2 depicts a typical ray profile for each of these techniques. A linear mapping of data values to grayscale intensities and (for DVR and MIDA) opacities is used. In the case of DVR, opacity is accumulated quickly resulting in a low overall intensity which manifests itself in the corresponding rendition as dark fog. For MIP, the intensity value along the ray is always equal to the current maximum. Using MIDA, due to the modulation of already accumulated intensities and opacities when a new maximum value is encountered, the intensity profile closely mimics the behavior of MIP. As visible by comparing Figure 2.1 (a) and (b), MIDA is able to immediately depict high-intensity features not visible in DVR without further transfer function modification. In contrast to the MIP result depicted in Figure 2.1 (c), however, MIDA features additional spatial cues due to accumulation which help to interpret feature locations.

### 2.4.1 Shading

In addition to accumulation, surface-based shading in DVR can provide important visual cues. It allows better judgement of shapes and fine details. MIP images, however, are usually devoid of this information. One reason for this is that in many cases the maximum along a ray may be located in a homogeneous area where surface shading produces bad results. Moreover, as the spatial location of the maximum between neighboring rays can vary considerably, lacking coherence between pixels can lead to disturbing artifacts. One major advantage of MIDA compared to common MIP is that the binary decision of which value to depict for each ray is replaced by accumulation. As the maximum normally does not change abruptly due to partial volume effects, samples in the boundary regions of high-intensity structures contribute to the ray color. This leads to improved coherency and allows us to perform surface-based shading as it is common in DVR without introducing the artifacts that would occur if it was applied to MIP. Figure 2.3 shows a comparison between (a) MIP, (b) MIP with shading applied, and (c) shaded MIDA. While shaded MIP fails to capture surface characteristics and introduces distracting artifacts, MIDA gives a good indication of the vascular shape. The magnified area shows how shading information can be helpful – a small aneurysm which is almost invisible in the MIP image is clearly recognizable using MIDA.

The gradient is a bad predictor for the surface normal orientation in nearly homogeneous regions due to the increased influence of noise. One approach to remedy this problem is to use the magnitude of the gradient vector to determine the degree of shading that is applied. In cases of low gradient magnitude the unshaded color is used, while higher gradient magnitudes result in an increased degree of shading. To achieve this, we linearly interpolate between the unshaded and the shaded color with an interpolation weight of *smoothstep*$(|\nabla f_{P_i}|, g_l, g_h)$ where $|\nabla f_{P_i}|$ is the gradient magnitude and $g_l$, $g_h$ are lower and upper thresholds. The smoothstep function smoothly transitions from zero to one as the first argument varies between $g_l$ and $g_h$ and is commonly implemented as a cubic polynomial. For the thresholds we use the empirically determined values $g_l = 0.125$ and $g_h = 0.25$ which have delivered universally good results in all our experiments and do not require user adjustment. This modulation is used in all images which feature shading throughout this paper.

### 2.4.2 Classification

A major advantage of MIDA, in contrast to DVR, is that it does not require complex transfer functions as the opacity profile is modified based on the difference between sample value and the current maximum. The result is an image which depicts the same essential features as a MIP rendering. We can therefore limit transfer function modification to brightness and contrast adjustment using common window/level controls which are almost universally incorporated in applications for the visualization of volume data. The user alters two parameters: the window $w$ and the level $l$. The interval $[l - 0.5w, l + 0.5w]$ is then linearly mapped to the full range of grayscale intensities from black to white and opacities from zero to one. Alternatively, the user can choose other color maps based on domain conventions, but the opacity function does not require modification. Window/level adjustment is a routine task for domain experts. Additionally, methods for automatically finding good window/level settings are frequently employed at the time of acquisition and parameter values are commonly stored together with the data (e.g., in the DICOM format). However, MIDA already provides acceptable results for an even more restricted case: if the opacity is one and the color is white for all data values, MIDA – due to shading – depicts the essential features of the data set. Both, DVR and MIP would simply show a white image.

## 2.5 Combining DVR, MIP, and MIDA

Instead of advocating the complete replacement of DVR and MIP by our new method, we recognize that MIDA represents a good middle ground between these standard techniques. Some data sets can be visualized better with DVR, for others MIP is more suitable. In practice, some characteristics of both are desirable in many cases [51]. Therefore one valuable approach is to use MIDA as a basis for a smooth transition between these different algorithms. We present a method where the user can smoothly make a visualization more DVR-like or MIP-like. As demonstrated, MIDA represents a hybrid between both methods and is therefore an ideal starting point. We introduce a new parameter $\gamma \in [-1,1]$ defined as follows: For $\gamma = -1$, the rendering result is unmodified DVR, if $\gamma = 0$ only MIDA is used, and for $\gamma = 1$ the resulting image will be a MIP rendering. Values of $\gamma \in (-1,0)$ will result in a smooth transition between DVR and MIDA, while values of $\gamma \in (0,1)$ cause a transition from MIDA to MIP.

**MIDA to DVR.** As discussed in Section 2.4, MIDA introduces an additional modulation of the previously accumulated color and opacity along a viewing ray. If the modulation factor $\beta_i = 1$ for all samples along the ray, the result is DVR. For a smooth transition between MIDA and DVR, we simply modify $\beta_i$ so that it approaches one for all samples when $\gamma$ changes from MIDA to DVR in the following manner:

$$\beta_i = \begin{cases} 1 - \delta_i(1+\gamma) & \text{if } \gamma < 0 \\ 1 - \delta_i & \text{otherwise} \end{cases} \tag{2.3}$$

Visually, this results in high intensity values smoothly fading out as $\gamma$ is reduced. As the user moves $\gamma$ towards DVR, more and more occlusion occurs.

**MIDA to MIP.** We could use a similar approach for the transition from MIDA to MIP, having $\beta_i$ approach one only for samples where the maximum changes and letting it approach zero for all other cases. Essentially, regions where accumulation is performed would be "thinned" to the point where only one value along the ray is accumulated. However, shading would lead to problems in this case – the same artifacts as in Figure 2.3 (b) would arise as $\gamma$ gets closer to one. Thus, we choose a different approach where the transition

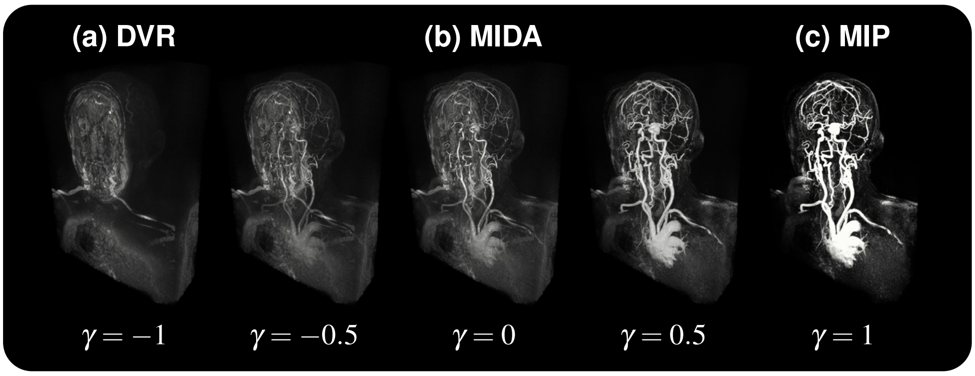


**Figure 2.4 –** MRI scan rendered using (a) DVR, (b) MIDA, and (c) MIP. Data set courtesy of the OsiriX Foundation (http://www.osirix-viewer.com).

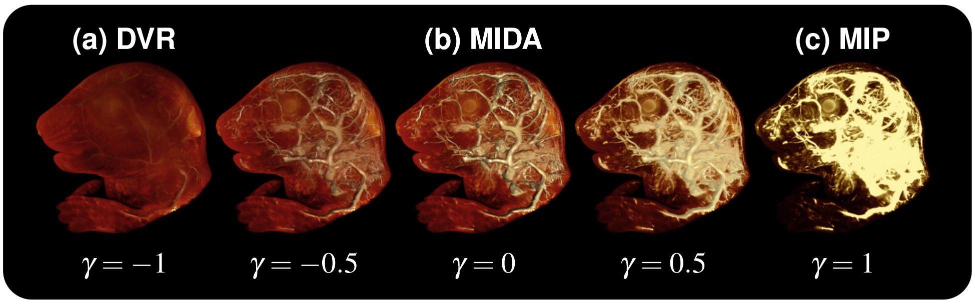


**Figure 2.5 –** Ultramicroscopy of a mouse embryo rendered using (a) DVR, (b) MIDA, and (c) MIP. Data set courtesy of Dodt et al. [40].

between MIDA and MIP is performed in image space. If $\gamma > 0$, we linearly interpolate between the accumulated MIDA color and opacity and the color and opacity of the maximum value after the ray has been traversed using $\gamma$ as the interpolation weight. Interpolation is performed using opacity-weighted colors. Since MIDA and MIP images share the same basic characteristics, the major visual impacts of this transition are the gradual reduction of shading and the darkening of areas where much accumulation is performed.

## 2.6 Results and Discussion

The user can employ $\gamma$ as a simple means for making a visualization result more DVR-like or more MIP-like. Instead of discretely switching between rendering methods, this allows a continuous manipulation of image characteristics on a clearly defined scale. The effects of modifying $\gamma$ are easy to interpret: moving from MIDA ($\gamma = 0$) towards DVR ($\gamma = -1$) increases occlusion – higher data values shine through less and less. Going from MIDA ($\gamma = 0$) to MIP ($\gamma = 1$), accumulation and shading are reduced.

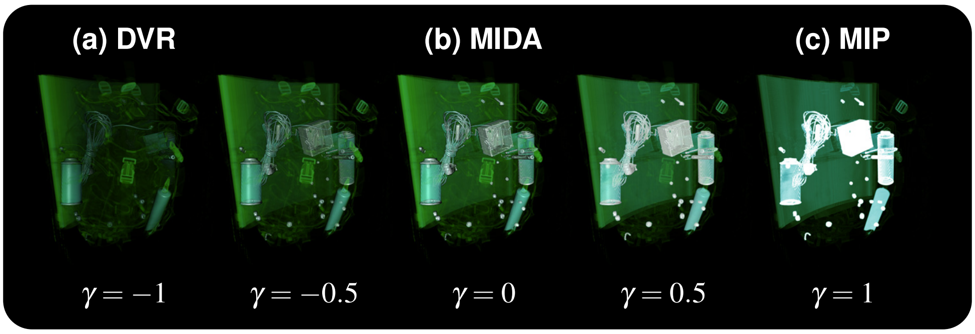


**Figure 2.6 –** CT scan of a backpack filled with various items rendered using (a) DVR, (b) MIDA, and (c) MIP. Data set courtesy of Kevin Kreeger, Viatronix Inc. (http://www.volvis.org).

In our experiments, we compared DVR, MIDA, and MIP applied to a number of different data sets. Figure 2.4 depicts an MRI angiography data set rendered using DVR, MIDA, and MIP. MIDA, like MIP, is able to depict the high-intensity vascular structures without further transfer function adjustment. However, MIDA provides additional spatial context as well as shape cues due to shading. In biomedical research, structures under investigation are commonly highlighted using fluorescent markers. MIP is therefore frequently employed for 3D visualization. DVR transfer functions are particularly difficult to find for these data sets, as there is no well-defined scale for the measured quantity. We experimented with several data sets from this field and MIDA helps to improve spatial comprehension compared to MIP. An example is shown in Figure 2.5, where a mouse embryo imaged using ultramicroscopy [40] is depicted. Due to shading, MIDA depicts vascular structures more clearly. Furthermore, as MIDA can generate meaningful images without requiring much transfer function tuning, it is well-suited for visualization tasks with strict time constraints. One example is the screening of luggage based on tomographic modalities such as CT. While conventional X-Ray is still the default modality for examining items in security-critical environments, it is often difficult to recognize potential threats due to its two-dimensional nature. Thus, in recent years CT-based screening has found increasing adoption in this area and is currently employed at many institutions such as airports and government buildings [177]. Figure 2.6 shows a CT scan of a backpack filled with various items – MIDA instantly allows the operator to inspect the contents and provides more information about the 3D structure of the individual items than MIP.

An existing DVR implementation can be extended to MIDA in a straightforward manner by modifying compositing as described in Section 2.4. In terms of performance, the additional instructions required by MIDA typically can be neglected. However, as the opacity along a ray is no longer monotonically increasing, early ray termination is not possible. Also, in contrast to MIP, ray traversal can not be terminated when the current maximum is the highest intensity value in the data set. Despite the lack of these optimizations, current graphics hardware is still easily capable of rendering typical data sets at interactive frame rates. The average frame rates of our implementation measured on an NVidia GeForce 8800 GTX GPU for the standard UNC head data set ($256 \times 256 \times 224$) with a viewport size of $512 \times 512$ and an object sample distance of 1 were 9.9 (DVR), 9.1 (MIDA), and 18.2 (MIP). As MIDA performs only slightly worse than DVR, we have not further investigated potential acceleration techniques.

## 2.7 Conclusions

In this paper, we introduced MIDA, a simple unifying extension of DVR and MIP which incorporates characteristics of both techniques. In contrast to DVR, MIDA does not rely on complex transfer functions but still features important spatial cues due to accumulation and shading. Furthermore, we presented an approach for smoothly interpolating between DVR, MIDA, and MIP. This allows users to enhance visualizations generated using DVR to include characteristics of MIP and vice versa. MIDA can be incorporated into existing volume rendering systems in a straight-forward manner. In experiments, our new technique has shown to achieve promising results for a wide range of different types of volume data.

**The following chapter was originally published as:**

S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010.

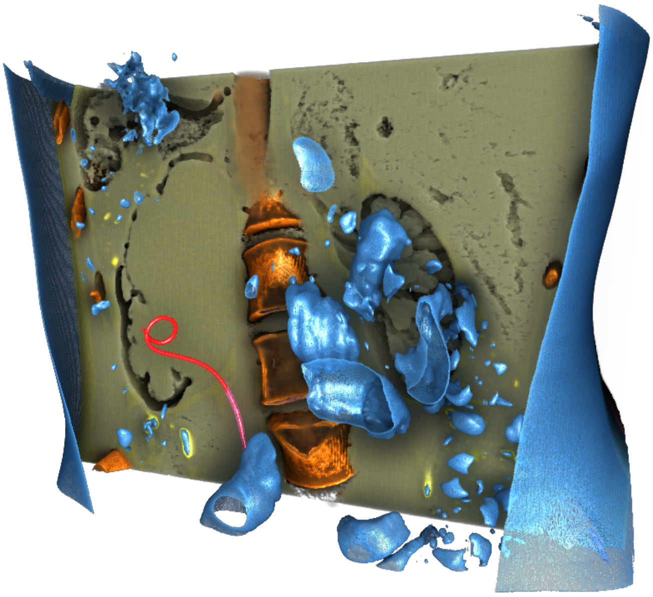


**Figure 3.1 –** Automatically classified CT data set using the eight most representative isovalues.

> The criterion of truth is that it works even if nobody is prepared to acknowledge it.
>
> — *Ludwig von Mises*

CHAPTER

# 3

# Isosurface Similarity Maps

In this paper, we introduce the concept of isosurface similarity maps for the visualization of volume data. Isosurface similarity maps present structural information of a volume data set by depicting similarities between individual isosurfaces quantified by a robust information-theoretic measure. Unlike conventional histograms, they are not based on the frequency of isovalues and/or derivatives and therefore provide complementary information. We demonstrate that this new representation can be used to guide transfer function design and visualization parameter specification. Furthermore, we use isosurface similarity to develop an automatic parameter-free method for identifying representative isovalues. Using real-world data sets, we show that isosurface similarity maps can be a useful addition to conventional classification techniques.

## 3.1 Introduction

The field of volume visualization aims to provide insightful depictions of three-dimensional data. Modalities such as CT, MRI, or laser-scanning confocal microscopy allow physicians, scientists, and engineers to investigate the interior of complex objects. However, providing clear visualizations of the structures contained in a volume data set is a major challenge. One of the issues is the lack of explicit geometric information and limited semantics. A volume data set contains a large number of isosurfaces at different target scalar field values, while its structure is typically characterized by a finite number of feature isosurfaces that segment the data set into several important components. The data may be visualized directly by mapping the scalar values and/or derived attributes to optical properties, or a geometric surface representation may be extracted using techniques like the popular Marching Cubes algorithm [98]. Irrespective of the chosen visualization method, providing guidance in the identification of salient isovalues plays an important role in improving the exploration process. In this paper, we present a new approach for the visualization and detection of relevant isovalues which provides additional information compared to conventional histograms.

Histogram-based methods typically infer similarity from the frequency at which isovalues occur. While there are cases where this assumption holds, in general only limited information can be deduced using this approach. Moreover, the presence of large homogenous regions, acquisition artifacts, and noise introduces additional problems. Instead of the frequency of individual data values and/or derived attributes, our method is based on the global notion of isosurface similarity. Using the information-theoretic measure of mutual information, we compare all combinations of isosurfaces to determine their degree of dependency. This process results in an isosurface similarity map which provides a compact overview of the similarities.

The main contribution of this paper is a new approach for quantifying and visualizing the similarity between isosurfaces in a scalar field. We demonstrate its applicability for simplifying isovalue selection and enhancing scalar-field visualization. However, we want to stress that we do not directly compete with the plethora of techniques which employ multi-dimensional transfer functions based on local voxel properties. We recognize that it is useful and, for many types of data, unavoidable to employ such classifiers in order to separate features which share the same value ranges. Indeed, as the isovalue is one axis in many multi-dimensional transfer function domains, our approach complements these techniques.

## 3.2 Related Work

Due to the complex nature of volumetric data sets, techniques for providing a simplified view of the data are frequently used. Their purpose is to guide the user to interesting regions which can subsequently be investigated in a three-dimensional view. Histograms are one of the most common representations employed for this purpose. They visualize the data set by depicting the number of voxels for each data value. Carr et al. [26], in a result later refined by Scheidegger et al. [139], demonstrated that this actually converges to the distribution of isosurface areas but is formally equivalent to the nearest neighbor interpolant. They proposed several practical measures which show better convergence behavior. Isosurface area, however, still conveys limited information about the nature and structure of a data set. For this reason, Bajaj et al. [9] displayed a variety of additional isosurface statistics in their contour spectrum. Pekar et al. [123] suggested the use of a Laplacian-weighted histogram to assist in the detection of significant isovalues.

Several approaches apply topological analysis to volume data. The contour tree [27] is an abstraction of a scalar field that encodes the nesting relationships of isosurfaces. Takahashi et al. [149] employed a volume skeleton tree to identify isosurface embeddings in order to provide additional structural information. Hyper Reeb graphs, proposed by Fujishiro et al. [54], capture the topological skeleton of a volumetric data set and can serve as a reference structure for designing comprehensible volume visualizations. One problem of these methods is that they rely on geometric extraction processes which suffer from noise in real-world data.

Much work has focused on the collection of local properties such as first and second derivatives. Plotting these attributes against isovalues provides guidance for feature selection. Kindlmann and Durkin [83] demonstrated that a two-dimensional histogram of data value and gradient magnitude enables the identification of boundary regions which manifest themselves as arches. Kniss et al. [86] extended this idea to the common notion of two-dimensional transfer functions and developed a direct manipulation interface for their specification. Lum et al. [100] used additional gradient-aligned samples depicted in a line-based histogram. Šereda et al. [159] extended this idea by searching for high and low values in paths that follow the gradient. Tenginakai et al. [150] employed multi-dimensional histograms based on local higher-order moments to detect important data values. To better characterize the shape of local features, Sato et al. [138] used the matrix of second derivatives. The role of curvature was investigated by Kindlmann et al. [84] and Hladůvka et al. [69]. Roettger et al. [134] extended histograms by incorporating spatial information. Lundström et al. [101] proposed the use of local histograms to better represent the distribution of intensity values in a given neighborhood which improves tissue separation for the case of overlapping intensity ranges. Correa and Ma proposed classification approaches based on size [31], occlusion [33], and visibility [32].

Due to this large number of different classification criteria, several approaches proposed the use of dimensionality reduction techniques to identify regions based on high-dimensional voxel signatures. Tzeng et al. [154] used machine learning methods to generate classifications based on a simple painting interface. They also presented a cluster-space approach for interacting with multiple classification criteria [155]. Šereda et al. [142] employed hierarchical clustering of material boundaries. Pinto et al. [126] utilized self-organizing maps to reduce the dimensionality of the classification space.

Previous methods attempt to characterize volume data sets by analyzing global isosurface statistics, extracting topological relationships, or collecting local voxel signatures. Our approach is fundamentally different in that we measure similarities of isosurfaces as a whole based on a robust information-theoretic measure. We show that this notion can yield additional insights into the structure of the data and can serve as the basis for enhanced visualization.

## 3.3 Isosurface Similarity

Isosurfaces play a crucial role in visualizing and interpreting volumetric data. In many cases they represent important object and/or material boundaries. However, in practice it is difficult to identify salient isovalues. Traditionally, one- and two-dimensional histograms have been employed to assist the user in this process. These approaches depict the frequency of isovalues and other attributes (e.g, gradient magnitude). Peaks or clusters in these plots then guide the exploration and visualization process. Frequency, however, can be a problematic measure as large regions such as the background intensity tend to dominate.

We propose *isosurface similarity* as an alternative measure for identifying features and guiding visualization parameter specification. Instead of collecting statistics of individual isosurfaces and using their variation to obtain information about their significance, we are interested in investigating similarities between individual isosurfaces directly, i.e., how much does knowledge about one surface tell us about the others. In this section, we first introduce our new measure for isosurface similarity. We then apply this measure to the isosurfaces of a scalar field to provide an overview of the mutual similarities which complements traditional frequency-based depictions.

### 3.3.1 Similarity Measure

An *isosurface*, or level set, of a volumetric scalar-valued function $f : \mathbb{R}^3 \to \mathbb{R}$ is the locus of all points at which $f$ attains the *isovalue* $h$:

$$L_h = \left\{x \in \mathbb{R}^3 : f(x) = h\right\} \tag{3.1}$$

A popular information-theoretic measure of similarity which has been applied in many areas including shape registration [73], multi-modality fusion [57], and viewpoint selection [158] is *mutual information*. The mutual information of two discrete random variables $X$ and $Y$ can be defined as [176]:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x,y) \log\left(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\right) \tag{3.2}$$

where $p_{X,Y}$ is the joint probability distribution function of $X$ and $Y$, and $p_X$ and $p_Y$ are the marginal probability distribution functions of, respectively, $X$ and $Y$. Mutual information quantifies the information that $X$ and $Y$ share by measuring how much knowing one of these variables reduces the uncertainty about the other. In other words, mutual information measures the dependence between the joint distribution of $X$ and $Y$ and what the joint distribution would be if $X$ and $Y$ were independent. The mutual information of $X$ and $Y$ is zero if and only if they are statistically independent. In the case of identity of $X$ and $Y$, the mutual information is equal to the uncertainty associated with the random variable, i.e., its entropy. Mutual information can be equivalently expressed in terms of entropy as:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.3}$$

with

$$
\begin{aligned}
H(X) &= -\sum_{x \in X} p_X(x) \log\left(p_X(x)\right) && (3.4)\\
H(Y) &= -\sum_{y \in Y} p_Y(y) \log\left(p_Y(y)\right) && (3.5)\\
H(X,Y) &= -\sum_{x \in X}\sum_{y \in Y} p_{X,Y}(x,y) \log\left(p_{X,Y}(x,y)\right) && (3.6)
\end{aligned}
$$

where $H(X)$ and $H(Y)$ denote the marginal entropies and $H(X,Y)$ is the joint entropy of $X$ and $Y$. As it is convenient to work with values in $[0,1]$, a normalized measure can be obtained by [92]:

$$\hat{I}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)} \qquad (3.7)$$

Volume data frequently exhibit an onion-peel-like structure and contain material inhomogeneities, partial volume effects, and noise. This results in several redundant isosurfaces, i.e., they do not represent substantial additional information. We would like to obtain a measure which classifies these kind of isosurfaces as similar. In the registration literature, shape representations based on implicit distance functions are commonly used as they have proven to be stable and robust to shape perturbations and noise [73]. For these reasons, we choose to represent individual isosurfaces using their *distance transform*. The distance transform $D_h$ of an isosurface with isovalue $h$ gives the minimum distance of a point $x$ to the surface [77]:

$$D_h(x) = \min_{\forall y \in L_h} d(x,y) \qquad (3.8)$$

where $d$ is a distance measure (for the remainder of this paper we will assume the Euclidean distance). We can now consider the distances from any point to a pair of isosurfaces $L_p$ and $L_q$ as random variables $X$ and $Y$. In order to compute the mutual information between the two isosurfaces, we need to estimate the joint distribution of $X$ and $Y$. This can be accomplished using the joint histogram of $D_p$ and $D_q$: for every voxel position $x$, we record the distances $D_p(x)$ and $D_q(x)$ in a two-dimensional histogram where each bin represents a certain range of distances. The marginal probability distributions of $X$ and $Y$ can be estimated by summing over the columns and rows, respectively, of the joint histogram. This allows us to directly compute $H(X,Y)$, $H(X)$, and $H(Y)$ to evaluate the normalized mutual information, as defined in Equation 3.7, of the two isosurfaces as a measure of their similarity.

### 3.3.2 Similarity Maps

In order to obtain information about the similarity relationships in a data set, we introduce the *isosurface similarity map*. Given the discrete set of $N$ isovalues

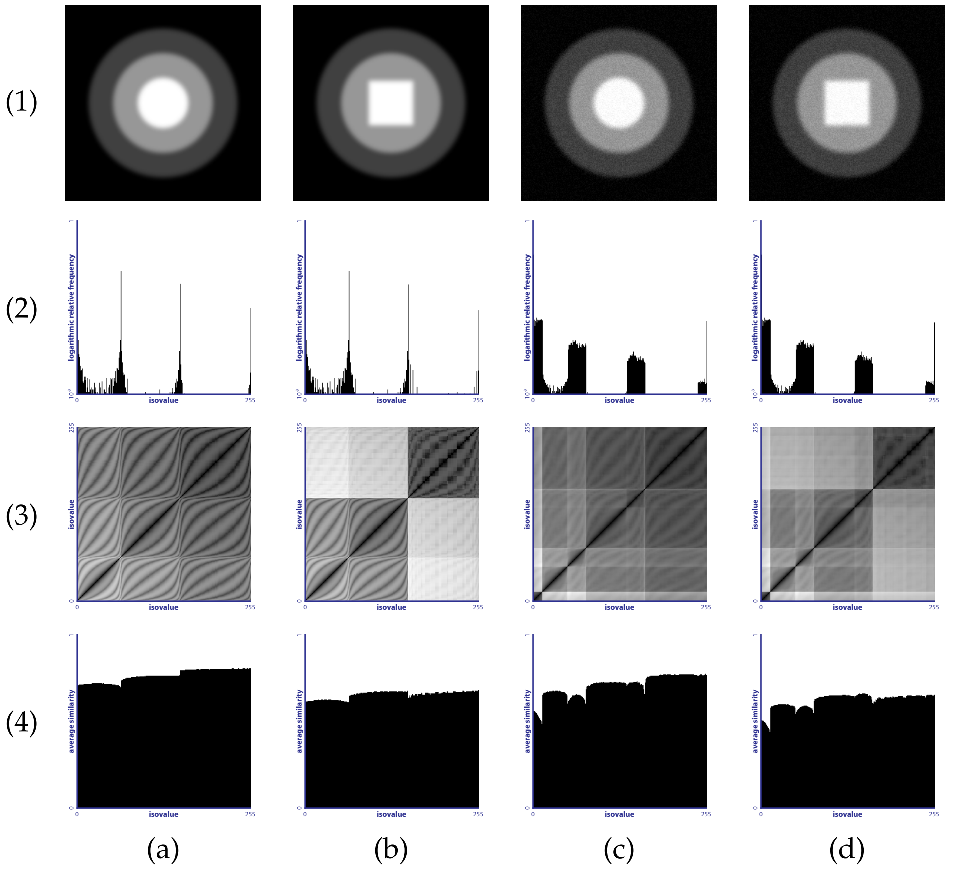
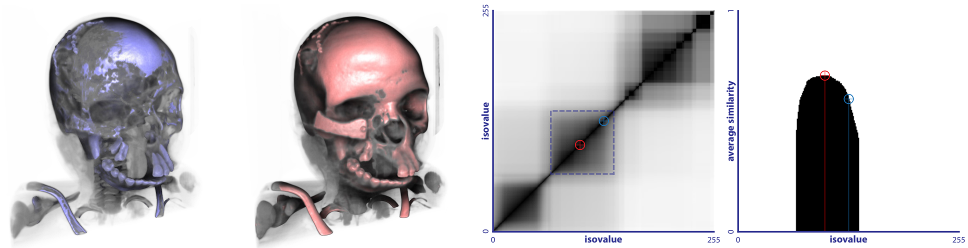


**Figure 3.2 –** Example of our isosurface similarity measure. Row (1): column (a) shows an image of three concentric circles, in column (b) the innermost sphere is replaced by a square of approximately equal area, and columns (c) and (d) show the same two images with added noise. In row (2) the respective histograms are shown. Row (3) depicts the corresponding isosurface similarity maps. Row (4) shows the isosurface similarity distributions.

$V = \{h_1, \ldots, h_N\}$ in a data set, we generate a $N \times N$ matrix $SM_V(i,j)$ containing the isosurface similarity, computed as described in the previous section, for each combination of isovalues $h_i$ and $h_j$. We will also use the notation $SM_V(x,y)$ to denote the matrix element $SM_V(i,j)$ with $x = h_i$ and $y = h_j$ when convenient. Due to the properties of mutual information, the map is symmetric and one along the main diagonal. As it records the similarity between every pair of isosurfaces, it provides an overview of the similarity relationships in the data set. In contrast to histograms, which visualize the frequency of individual values, it instead depicts the similarity of isosurfaces measured by the mutual information of their distance fields.

By summing over the rows (or columns) of the isosurface similarity map and normalizing the result by the number of isovalues, we obtain the *isosurface*



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

**Figure 3.3 –** Result images for similarity-enhanced isosurface visualization are shown in (a) and (b), (c) depicts the isosurface similarity map of the data set, and (d) shows the isosurface similarity distribution for the highlighted region in the similarity map. The isovalues depicted in (a) and (b) are marked with corresponding colors in (c) and (d).

*similarity distribution* $SD_V$:

$$SD_V(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} SM_V(i,j) \tag{3.9}$$

The isosurface similarity distribution describes the average similarity for each individual isosurface. Peaks in the similarity distribution correspond to those isosurfaces which are most similar to others while valleys indicate regions of rapid change. As will be shown in Section 3.4, it is frequently useful to investigate the similarity distribution for a specific subset of isovalues. In practice, a summed-area table of the similarity map enables the efficient evaluation of similarity distribution values for arbitrary continuous subranges.

Figure 3.2 shows a simple example. The image in row (1), column (a) contains three concentric circles with different isovalues. In column (b) the innermost circle is replaced by a square of approximately equal area. Expectedly, the corresponding histograms shown in row (2) are essentially identical. The isosurface similarity maps depicted in row (3), however, show considerable differences. Similarity is linearly mapped to grayscale intensity where white means a similarity of zero and black corresponds to a similarity of one. While row (3), column (a) shows a high degree of mutual similarity between the three spheres, the square's presence is clearly indicated in row(3), column (b). This is also reflected in the corresponding similarity distributions depicted in row (4). Columns (c) and (d) of the figure demonstrate that the basic structure of similarity map and distribution remains the same even though noise has been added to the images.

## 3.4 Applications

In this section, we present applications of isosurface similarity maps and distributions for the visualization of volume data. We do not advocate replacement of well-proven methods such as histograms which are clearly useful for many purposes nor do we propose similarity as a sole classification criterion. Instead, we want to demonstrate that isosurface similarity provides additional information which can be exploited to build improved tools for volume visualization.

### 3.4.1 Similarity-Enhanced Isosurface Visualization

A common problem in volume visualization is that even minor changes in the selected isovalues can have dramatic impact on the depicted features. The importance of providing the user with information on this kind of uncertainty was demonstrated by Lundström al al. [102] in the context of stenosis assessment. In a similar spirit, isosurface similarity allows us to indicate the stability of an isosurface by visually encoding the similarity of a sample point with respect to a specified isovalue.

We want to depict the isosurface at an user-selected isovalue $h_u$ in a focus+context manner using direct volume rendering [64]. In addition to the isosurface itself (focus), information about similar regions (context) should be conveyed to the user. To prevent visual overload, the contextual regions should be depicted in a sparse way. We use the following importance function $\gamma(x)$ which determines the degree-of-interest in a sample at location $x$:

$$\gamma(x) = \prod_{y \in C(x)} SM_V\left(h_u, f(y)\right) \tag{3.10}$$

where $f$ is the scalar field and $C(x)$ denotes a local neighborhood of samples around $x$. In practice, we reuse the sample values required for gradient estimation using central differences as the sample neighborhood. The effect of this function is that, due to the product of similarities of neighboring voxels, only contextual regions with high local similarity will tend to exhibit high importance. For visualizing the isosurface $h_u$ we specify the opacity of a sample $\alpha(x)$ as:

$$\alpha(x) = \begin{cases} 1 & \text{if} f(x) \geq h_u \\ \gamma(x) & \text{otherwise} \end{cases} \tag{3.11}$$

To clearly distinguish between focus and context, $\gamma$ is also used to control the color and the degree of surface shading. Additionally, the directional occlusion model of Schott et al. [141] is used uniformly for all samples.

An example is shown in Figure 3.3. The isovalue selected in Figure 3.3 (a) is very unstable – the extent of the cloud surrounding the surface indicates that it only captures part of the structure of interest. Figure 3.3 (b) shows more

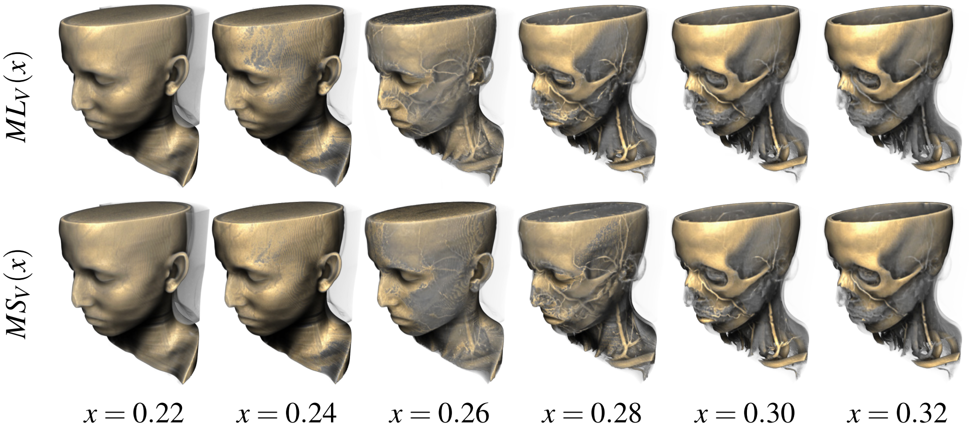


**Figure 3.4 –** Transition using linear mapping $ML_V(x)$ (top row) and similarity-based mapping $MS_V(x)$ (bottom row) for $x \in [0.22, 0.32]$.

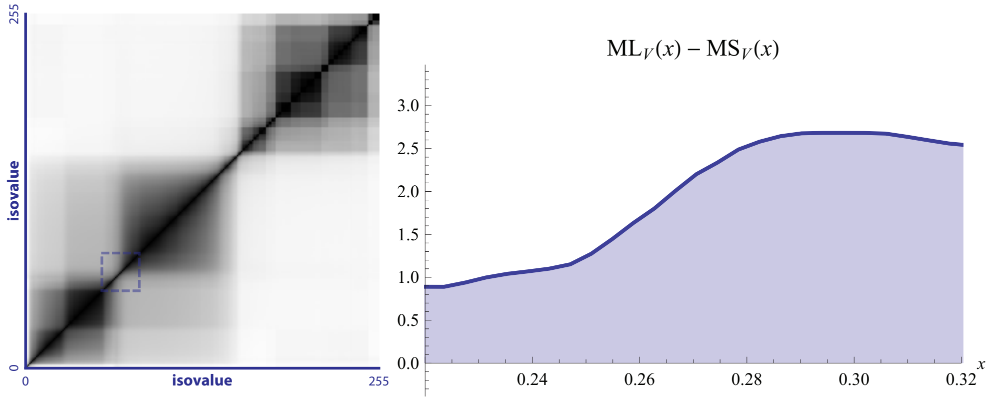
stability. Note that the similarity cloud has the same shape in both images – this means that both isosurfaces are part of a cluster of high mutual similarity. This is confirmed by locating the isovalues in the similarity map shown in Figure 3.3 (c) – both lie within one large cluster. Figure 3.3 (d) shows the similarity distribution for the indicated square region. It can be seen that the isovalue used in Figure 3.3 (a) has low average similarity, while the value from Figure 3.3 (b) is located at a peak, i.e., it represents the region well.

### 3.4.2 Similarity-Based Isovalue Remapping

Isovalues are typically selected and modified using user interface widgets such as sliders or by linearly mapping them to mouse movement. This, however, can be quite non-intuitive: if a subrange of isovalues corresponds to very similar isosurfaces, large changes of the value will have almost no effect on the depicted structures. Conversely, in regions of high dissimilarity even a minor modification can completely alter the appearance. Ideally, the effects in the visualization should correspond to the magnitude of change in the corresponding user interface component. Thus, instead of directly translating changes of the controlling element to changes in the isovalue, we use a nonlinear mapping based on isosurface similarity. Let $ML_V$ be the conventional direct mapping function which maps values $[0,1]$ linearly to the range of isovalues $[h_{min}, h_{max}]$ in the set $V$:

$$ML_V(x) = h_{min} + x(h_{max} - h_{min}) \tag{3.12}$$

The idea is to use a monotonous function $MS_V(x)$ with $MS_V(0) = ML_V(0)$ and $MS_V(1) = ML_V(1)$ whose derivative is controlled by the similarity of neighboring isovalues. For this purpose, we define the *cumulative similarity* of



(a) (b)

**Figure 3.5 –** Similarity-based isovalue remapping. (a) Isosurface similarity map for the data set shown in Figure 3.4 – the highlighted area indicates the interval of isovalues in the transition. (b) Plot of the difference between the linear mapping function $ML_V(x)$ and the similarity-based mapping function $MS_V(x)$ for $x \in [0.22, 0.32]$.

an ordered set of isovalues $V$:

$$SC_V(i) = \sum_{j=1}^{i} SM_V(j-1, j) \tag{3.13}$$

where $SM_V(0,1) = 0$. Our similarity-based mapping function can then be written as:

$$MS_V(x) = ML_V\left(\frac{SC_V\left(x(|V|-1)+1\right)}{SC_V(|V|)}\right) \tag{3.14}$$

The fact that this function is piecewise constant does not matter in practice, since we are typically only interested in discrete isovalues. However, one can simply use any interpolant for $SC_V$ to remedy this.

Figure 3.4 depicts a transition using $ML_V(x)$ (top row) and $MS_V(x)$ (bottom row) with $x$ varying from 0.22 to 0.32 in increments of 0.02. Even though the differences are subtle, the images generated using the similarity-based mapping function show a more uniform progression. The isosurface similarity map for the data set is shown an Figure 3.5 (a) – the highlighted area indicates the range of isovalues of the transition. Note that the dissimilar nature of the isosurfaces in the interval is clearly visible. The graph in Figure 3.5 (b) depicts the function $ML_V(x) - MS_V(x)$ for the chosen interval, i.e., the difference in isovalue resulting from using the similarity-based mapping function instead of the linear one.

### 3.4.3 Representative Isovalue Selection

An important problem in volume visualization has been the identification of relevant isovalues. Many approaches combine different isosurface statistics to

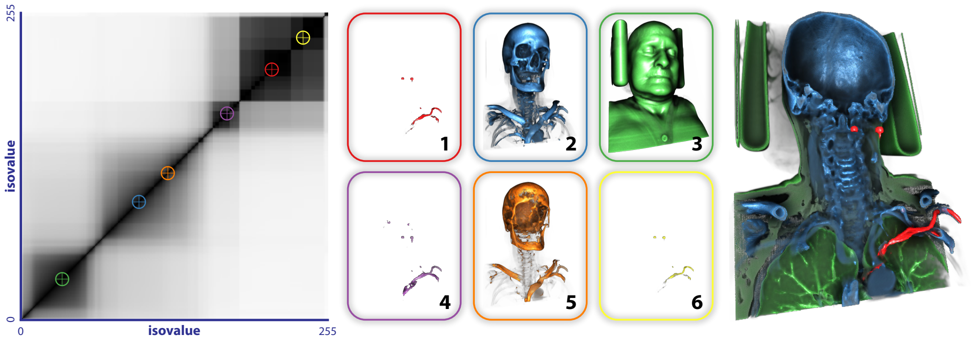


**Figure 3.6 –** Representative isovalue selection algorithm applied to a CT data set. The isosurface similarity map is shown on the left and the six most representative isovalues are marked. The corresponding isosurfaces are depicted in the middle section numbered from one to six with decreasing relevance. The image on the right shows a cutaway view of the data set classified according to maximum similarity with the six isovalues.

infer salient isovalues which characterize the function well. The isosurface similarity map can be used to provide guidance in finding representative isovalues. By selecting rectangular regions in the map and investigating their similarity distribution, insight on the relationships of isovalues and the corresponding structures can be gained. While manual analysis is useful and unavoidable for many exploratory tasks as the intent of the user is not known, the highly structured nature of the isosurface similarity map also provides us with means to automatically identify relevant isovalues. Regions of high similarity manifest themselves as distinct squares in the isosurface similarity map. We developed a simple algorithm which allows us to automatically identify these regions and, based on their similarity distributions, select the most representative isovalues for each of them:

**Step 1 –** Our aim is to identify representative isovalues, i.e., values with high similarity to many other values. Initially, the value with the highest average similarity to all others is identified. Then, the maxima of the similarity distribution for only the values below and above this value are chosen, one so on. Thus, our strategy recursively partitions the set of isovalues $V$ by selecting the maximum of the similarity distribution for the current subset. The chosen isovalue $m$ is inserted into a priority queue $Q$ based on its similarity distribution value weighted by the number of isovalues in the current subset. The following procedure

**prioritize(*Q*,*V*)** summarizes these operations:

$$\begin{aligned} m &\leftarrow \arg\max_{h_i \in V} SD_V(i) \\ p &\leftarrow |V| SD_V(m) \\ &\quad \text{enqueue}(Q, m, p) \\ V_1 &\leftarrow \{h_i \in V : 1 \leq i < m\} \\ &\quad \text{prioritize}(Q, V_1) \\ V_2 &\leftarrow \{h_i \in V : m < i \leq |V|\} \\ &\quad \text{prioritize}(Q, V_2) \end{aligned} \tag{3.15}$$

**Step 2 –** Next, we remove the isovalue $h_m$ with maximum priority from the queue. To prevent similar values from being chosen, all remaining entries are penalized based on their similarity with the selected value in the following manner:

$$p_i \leftarrow \frac{p_i}{1 + SM_V(h_m, h_i)} \tag{3.16}$$

where $p_i$ is the priority of the $i$-th item in the queue and $h_m$ is the selected value with the highest priority. This process repeats until no more items remain in the queue.

This simple algorithm results in a reordering of the isovalues. Early values in the resulting order have high similarity with many other isovalues, i.e., they represent a certain range of isovalues well, but low mutual similarity meaning that they are likely to correspond to distinct structures. One major advantage of this approach is that it does not require any kind of threshold or parameter. The user can simply examine the isosurfaces in the order generated by the algorithm. After the first few isovalues corresponding to distinct features of the data set, subsequent values will typically be less representative values for the same structures as no further dissimilar values can be found.

An example is shown in Figure 3.6. The six most representative isovalues of a CT data set determined using the described algorithm are marked in the isosurface similarity map and the corresponding isosurfaces are shown numbered from 1 to 6 with decreasing relevance. While the first three values correspond to distinct structures, the remaining ones only partially segment these features. The righthand side of the figure depicts a cutaway view where each voxel is classified according to its maximum similarity with any of the six isovalues – as the first three isovalues exhibit more similarity, the three less relevant isosurfaces are not visible. A further result for the classification of CT data using the most representative isovalues is shown in Figure 3.1. In CT data sets, there is a clear correspondence between isovalues and different tissue types. Even though other types of volume data do not exhibit the same kind

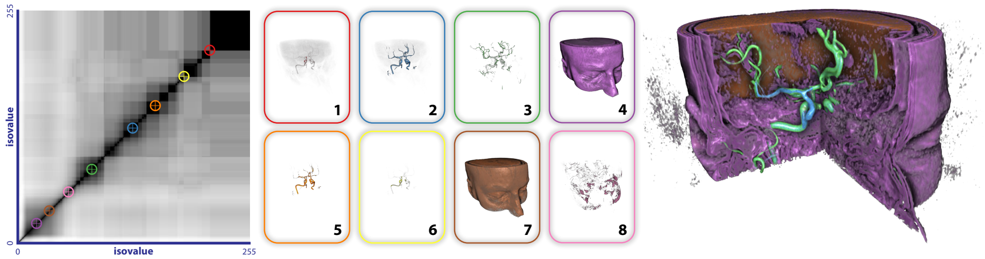


**Figure 3.7 –** Representative isovalue selection algorithm applied to an MRI angiography data set. The isosurface similarity map is shown on the left and the eight most representative isovalues are marked. The corresponding isosurfaces are depicted in the middle section numbered from one to eight with decreasing relevance. The image on the right shows a cutaway view of the data set classified according to maximum similarity with the eight isovalues.

of relationships and are therefore difficult to classify based on isovalues alone, our approach can still identify salient structures in these cases. Figure 3.7 shows an MRI data set and its eight most representative isovalues as identified by our method. In all depicted examples the only manual interventions were specification of the viewpoint and clipping.

## 3.5 Implementation

Our tool for the generation of isosurface similarity maps was implemented in C++. The computation process involves two steps. First, the distance transforms for all isovalues are computed. As this can, depending on the resolution, require a substantial amount of space they are immediately written to disk in compressed form. In the second step, we compute the mutual information of the distance transforms for each pair of isovalues. Since mutual information is a symmetric measure, only half of the combinations need to be evaluated. The computation is performed by generating the joint histogram of the two distance transforms which allows estimation of the joint and marginal entropies as described in Section 3.3.1 using the CUDA-based implementation of Shams and Barnes [143].

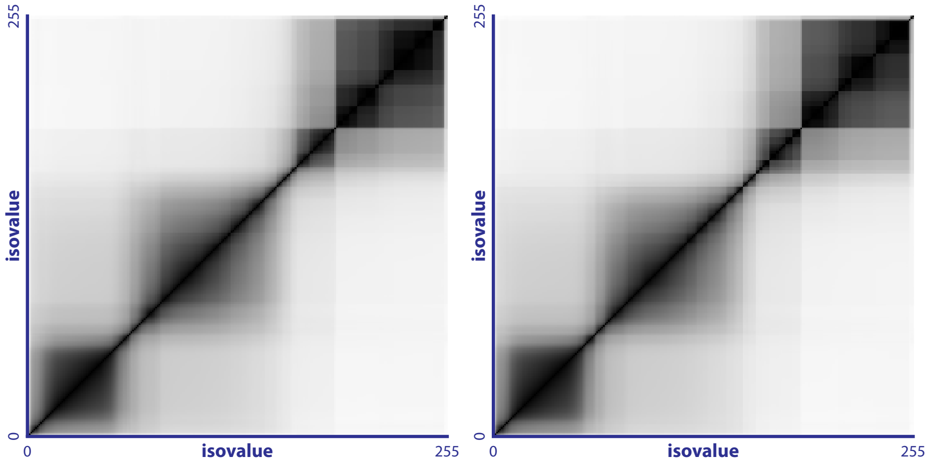
## 3.6 Discussion

In our experiments we found that isosurface similarity maps provide a concise overview of a data set. Distinct features manifest themselves as squares and their size informs the user about the corresponding value range. Transitional regions can be detected through the nesting structure of these squares. In contrast to histograms, large regions do not tend to dominate the depiction. Due to the choice of mutual information as a similarity measure, uncorrelated noise has little influence. These properties indicate that the presented concept has the potential to improve the visualization and analysis of volume data even beyond the examples shown in this paper.

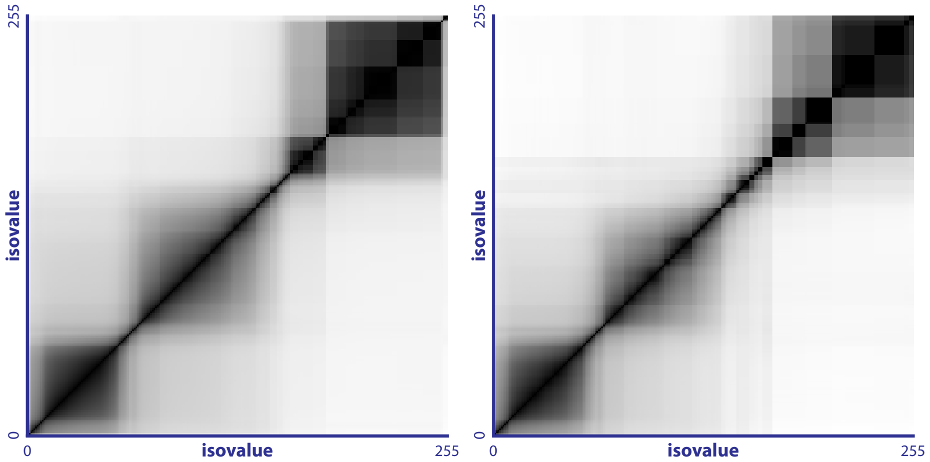
One obvious disadvantage of our approach is the considerable cost of generating the isosurface similarity map. Our implementation can require several hours of processing time. Even though this is a one-time preprocess, we consider this fact a serious limitation of our current method. The most performance-critical component is the construction of the joint histogram of two distance transforms since it has to be performed for each pair of isovalues. In order to reduce the computation time, we performed experiments with downsampled distance transforms. Interestingly, it seems that the resolution can be considerably reduced without substantial changes in the resulting similarity map. Figure 3.8 compares similarity maps computed from distance transforms at several resolutions. Note that the distance transforms are generated at the original resolution of the data set and then downsampled as opposed to computing them from a downsampled version of the volume. We believe that the reason for this stability is that the distance transform captures the unique characteristics of an isosurface even at reduced resolutions. Since it is not used to perform accurate spatial comparisons but rather as a shape descriptor, we consider downsampling a viable strategy. Table 3.1 lists the distance transform resolutions and similarity map computation times for all data sets used in the paper. On a typical notebook, generation of the isosurface similarity map using a distance transform resolution of approximately $64 \times 64 \times 64$ takes about 25 minutes for standard data sets. Throughout the paper, we used a fixed number of $128 \times 128$ bins in the computation of the joint histogram.

Even though lowering the resolution of the distance transform dramatically reduces the computation time to a level we consider acceptable, our generation method is still a brute-force approach. For a more fundamental improvement, one potential direction is the use of a different method for joint probability density estimation. While joint histograms are widely used, other methods are gaining increasing recognition. It may even be possible to use an alternative approach which does not require explicit computation of the distance transform. Interesting recent work by Rajwade et al [128] points in this direction and remains to be explored. Additionally, an adaptive strategy for approximating the full similarity map could also be employed. While our current method for generating isosurface similarity maps can be considered a proof-of-concept, we have shown that they have useful applications for the visualization of volume data. There are several other areas, however, where the proposed concept may be of interest. In closing, we would like to briefly list some avenues which could be promising directions for further exploration:

**Volume quantization and compression.** The notion of similarity may be useful in developing new approaches for quantizing and/or compressing



(a) $256 \times 256 \times 230$ (b) $128 \times 128 \times 115$



(c) $64 \times 64 \times 57$ (d) $32 \times 32 \times 28$

**Figure 3.8 –** Comparison of isosurface similarity maps computed with different distance transform resolutions for the data set shown in Figure 3.6. The total compuation times (in minutes) were (a) 569.1, (b) 35.8, (c) 22.1, and (d) 20.6.

volume data with higher fidelity. Isosurfaces which exhibit a high degree of redundancy could be grouped together while still preserving essential features in the data set.

**Volume segmentation.** Isosurface similarity could also be employed as an alternate metric for segmentation algorithms such as region growing. These methods typically use similarity criteria based on the difference between isovalues, so our measure may help to increase robustness.

| Figure(s) | Orig. Resolution | DT Resolution | Time |
|---|---|---|---|
| 3.3 | $512 \times 512 \times 361$ | $64 \times 64 \times 45$ | 23.9 |
| 3.4, 3.5 | $512 \times 512 \times 333$ | $64 \times 64 \times 41$ | 21.8 |
| 3.6 | $256 \times 256 \times 230$ | $64 \times 64 \times 57$ | 22.1 |
| 3.1 | $512 \times 512 \times 361$ | $64 \times 64 \times 45$ | 23.7 |
| 3.7 | $512 \times 512 \times 125$ | $64 \times 64 \times 15$ | 21.6 |

**Table 3.1 –** Depicting figure, original data set resolution, resolution of the downsampled distance transform, and total computation time (in minutes) for the isosurface similarity maps used in the paper are listed. System configuration: Intel Core 2 Duo 2.53 GHz CPU, 4 GB RAM, NVidia GeForce 9600M GT GPU.

**Multi-dimensional classification.** While there is nothing in our approach that prevents combination with multi-dimensional classification approaches using gradient magnitude [86], curvature [84], occlusion [33], or other measures proposed in the literature, we did not investigate this area. Indeed, as the isovalue typically represents one axis in multi-dimensional transfer function schemes, similarity could help to better separate features and to decrease the influence of noise.

## 3.7 Conclusions

In this paper, we introduced the notion of isosurface similarity for the visualization of volume data. This new measure quantifies the similarity of two isosurfaces as the normalized mutual information of their respective distance transforms. The resulting isosurface similarity map provides a visualization of these similarities and gives an overview of a data set which complements traditional depictions. Additionally, the similarity map can be used to improve rendering and parameter specification. Its structured nature enables automatic detection of representative isovalues to assist the exploration process. The presented concept opens up several interesting directions for future investigation.

**The following chapter was originally published as:**

M. Haidacher, S. Bruckner, and M. E. Gröller. Volume analysis using multimodal surface similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, 2011.

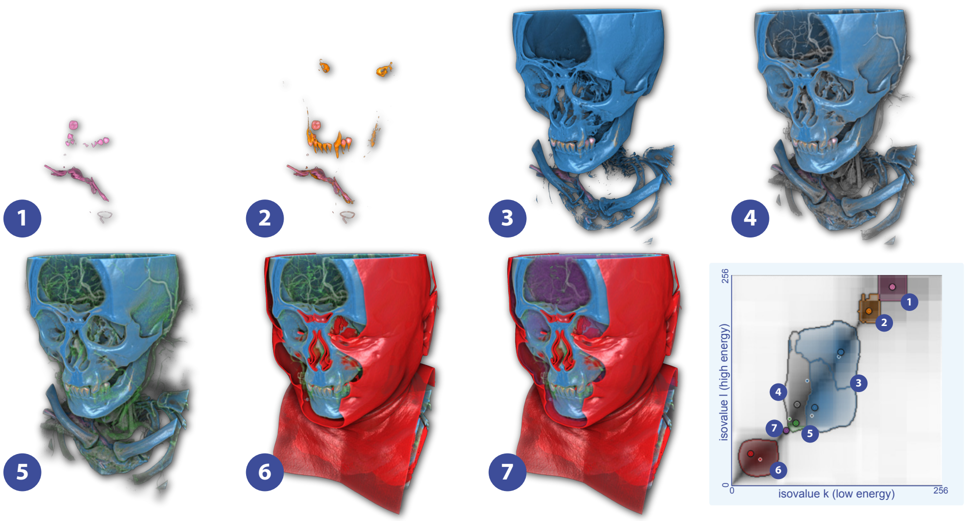


**Figure 4.1 –** Iterative control point specification for similarity-based classification of a dual energy CT (DECT) angiography data set. The individual steps are numbered from 1 to 7.

> One can measure the importance of a scientific work by the number of earlier publications rendered superfluous by it.
>
> — *David Hilbert*

# CHAPTER 4

# Volume Analysis Using Multimodal Surface Similarity

The combination of volume data acquired by multiple modalities has been recognized as an important but challenging task. Modalities often differ in the structures they can delineate and their joint information can be used to extend the classification space. However, they frequently exhibit differing types of artifacts which makes the process of exploiting the additional information non-trivial. In this paper, we present a framework based on an information-theoretic measure of isosurface similarity between different modalities to overcome these problems. The resulting similarity space provides a concise overview of the differences between the two modalities, and also serves as the basis for an improved selection of features. Multimodal classification is expressed in terms of similarities and dissimilarities between the isosurfaces of individual modalities, instead of data value combinations. We demonstrate that our approach can be used to robustly extract features in applications such as dual energy computed tomography of parts in industrial manufacturing.

## 4.1 Introduction

IMAGING modalities have different advantages and disadvantages typically related to the physical principles they use to scan a specimen. They may suffer from different kinds of artifacts, can be differently affected by noise, may be able to distinguish different materials or tissues, and can have differences with respect to contrast and resolution. In order to gain insight into the phenomenon under investigation, it is essential to integrate this information effectively. The work presented in this paper focuses on the analysis and fusion of two registered volume data sets of the same specimen. While the data generated by each modality may be visualized separately, it is difficult to mentally integrate multiple three-dimensional sources, particularly if spatial relationships are important. Thus, the effective visual fusion of multiple volume data sets has long been an active area of research. As discussed by Cai and Sakas [25], this combination can occur at different stages. At the extreme ends of the spectrum, the two data sets are treated separately and are only blended at the image level, or, conversely the data values at each position are combined at the very beginning of the pipeline to form a single merged volume. Most commonly visual fusion is performed during the rendering phase which provides spatial integration and allows for a flexible mapping of data attributes to optical properties [85].

While straightforward blending can be an effective technique in 2D slice views, it has many disadvantages in 3D visualization. In particular, the projection of multiple volumetric data sets onto a single 2D image can quickly lead to visual clutter. Hence, it is important to provide the user additional guidance about the spatial similarities and differences between the individual modalities to enable goal-directed selection of features. Approaches which attempt to identify correspondences based only on the frequency of data values, however, suffer from the fact that data value ranges of corresponding structures of interest may differ significantly. In order to address this challenge, we propose *multimodal surface similarity* as a measure for identifying similarities and dissimilarities between two volumetric scalar fields. Instead of collecting statistics about the frequency of data values, we quantify spatial similarities between isosurfaces across two modalities, i.e., how much does knowledge about one surface tell us about the others. By generating a multimodal similarity map, which encodes the similarity between all combinations of isosurfaces from two modalities, we can provide a concise overview of the differences between two scalar fields. This information can then be used to guide the identification of structures of interest. Based on this concept, we present a novel method for feature classification in similarity space which enables the user to easily take advantage of the complementary information provided by two modalities.

The remainder of the paper is structured as follows. In Section 4.2 we review related work on the visualization of multimodal volume data and other approaches connected to our work. Section 4.3 provides background information on the types of data we focus on. In Section 4.4, the general concept of multimodal surface similarity is introduced. In Section 4.5, we show how multimodal surface similarity can be used in the visualization process. Results obtained with our approach are presented in Section 4.6. Section 4.7 discusses implementation details. The implications of our approach as well as its limitations are discussed in Section 4.8. Finally, the paper is concluded in Section 4.9.

## 4.2 Related Work

As mentioned in the introduction, two volumes can be fused in different stages of the visualization pipeline [50, 53]. The fusion in image space is covered by the field of image processing [166]. The drawback of the fusion in image space is the loss of 3D information. For the fusion in volume space,

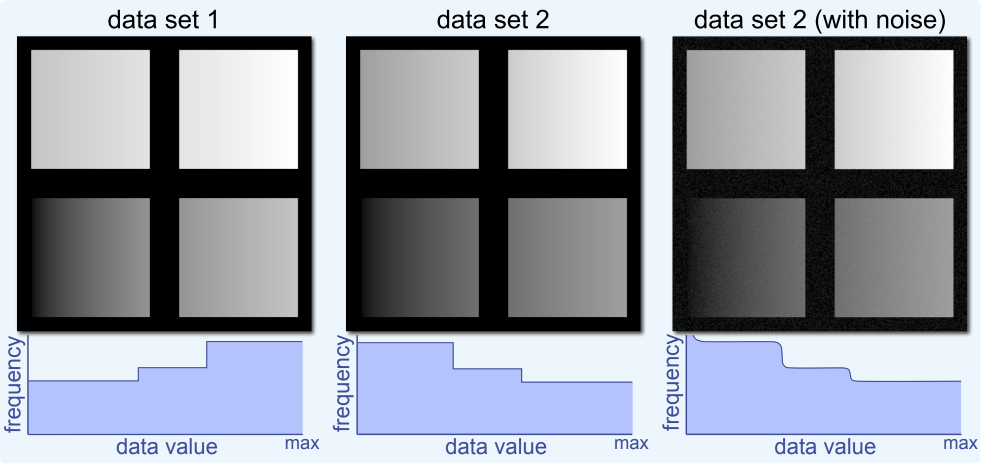


**Figure 4.2 –** Data sets containing the same structures with different value ranges (supplementary data). The histograms show the distributions of the data values.

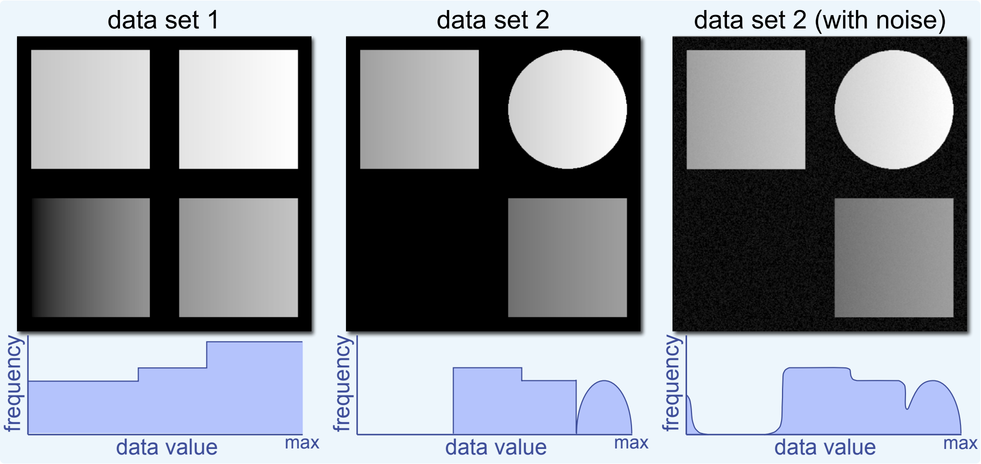


**Figure 4.3 –** Two synthetic data sets which represent complementary data. Data set 2 contains structures which are different from data set 1.

the spatial information of the data sets can be used to improve the fusion quality. The first methods for volume fusion were based on extracted surfaces. Levin et al. [95] generated a surface model from an MRI scan and mapped PET-derived measurements onto this surface. Evans et al. [45] generated an integrated volume visualization from the combination of MRI and PET. Noz et al. [119] introduced a framework for 3D registration and fusion of CT/MRI and SPECT data sets based on a polynomial warping technique. These works mainly focused on the combination of anatomical and functional modalities. A more general approach for the fusion of modalities was introduced by Zuiderveld and Viergever [179]. For this method an additional segmentation of the volumes is necessary to decide which one to show at a given sample point. Heinzl et al. [68] introduced a processing pipeline for surface extraction in dual energy CT.

Alternatively, fusion can be performed without an intermediate feature extraction step. A straightforward method is fusion by linear intermixing of the data values. Such an approach is used for volumetric CSG construction where different volumetric parts are fused into a single object [29, 47, 165]. Hong et al. [71] describes how fusion techniques in volume space can be efficiently implemented using the graphics hardware. Eusemann et al. [44] have shown that this intermixing can be improved for dual energy CT by adapting the intermixing ratio to different tissues. A case study on visualization of multivariate data where multiple values are present at each sample point was presented by Kniss et al. [85]. In this work the idea of multi-dimensional transfer functions for assigning optical properties to a combination of values was used. Akiba and Ma [3] used parallel coordinates for the visualization of time-varying multivariate volume data. Multimodal visualization of medical data sets by using multi-dimensional transfer functions was discussed by Kniss et al. [89]. Kim et al. [82] presented a technique which simplifies transfer function design by letting the user define a separate transfer function for each modality. Their combination defines a two-dimensional transfer function. Haidacher et al. [57] defined a data fusion and transfer function space for multimodal visualization based on the information content of the individual modalities which aims to reduce the loss of information. In contrast to our method, this approach is only based on the global frequency distribution of values and not on structural similarities between the individual modalities.

In our approach, we use information theory [144] to measure similarities between the different modalities, but it has been applied to many aspects of visualization [163]. In flow visualization, for instance, Xu et al. [174] used information theory to select meaningful streamlines. Feixas et al. [49] presented an information-theoretic approach for optimal viewpoint selection. Chen and Jänicke [28] discussed a general information-theoretic framework for scientific visualization.

For many applications, such as industrial CT [68], surfaces are of particular interest. Surfaces can be used to represent the interfaces between different materials. In order to extract a stable isosurface, the selection of the isovalue is crucial. Khoury and Wenger [81] use the fractal dimension to measure how stable an isovalue is. The lower the dimension, the less noisy the corresponding isosurface is. The contour tree [27] is used to topologically analyze volume data. It is able to encode the nesting relationships of isosurfaces. Takahashi et al. [149] employed a volume skeleton tree to identify isosurface embeddings in order to provide additional structural information. Kindlmann and Durkin [83] introduced a transfer function space in which the gradient magnitude is used as additional classification dimension. Interfaces between materials show up as arches in this transfer function space. In LH histograms,

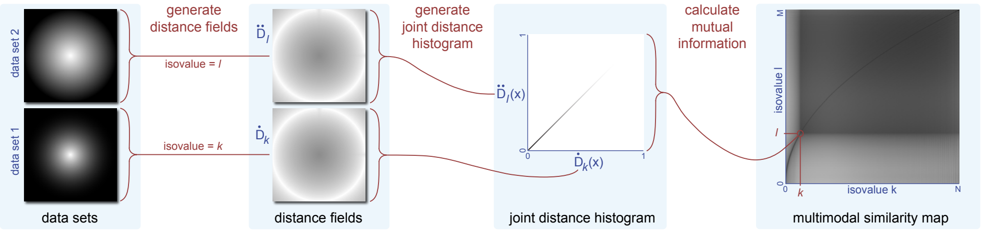


**Figure 4.4 –** Pipeline for the generation of a multimodal similarity map. The illustration shows which steps are necessary to calculate the similarity of isosurfaces for the isovalues $k$ and $l$.

introduced by Šereda et al. [159], the highest and lowest value along a local streamline in the gradient field are used for the classification. Sample points at interfaces between materials form clusters in this space, which represent stable surfaces.

Bruckner and Möller [20] introduced similarity maps which represent the similarity of isosurfaces for different isovalues. For the measurement of the similarity mutual information is used. In a similarity map clusters with high mutual information can be detected. In our approach we extend the idea of the similarity maps to multimodal data. The resulting multimodal similarity maps are used for analysis, fusion, and classification of multimodal data.

## 4.3 Multimodal Volume Data

There are different reasons for seeking to combine the information from multiple modalities. In medicine, for instance, it is frequently desired to simultaneously depict anatomical and functional data. Functional data contains information about physiological activities, such as metabolism or blood flow, within a certain tissue or organ. Anatomical imaging modalities, on the other hand, present structural information and typically provide higher resolution. In other fields, such as non-destructive testing, multiple industrial CT scans with different parameters are used for scanning an object. These parameters can affect the contrast and amount of artifacts in different regions. Thus, the goal in this case is to combine the advantages of different scans in order to obtain a better visualization of the object.

For the further description of our approach we will differentiate between two types of multimodal data which are depicted in this section. We will introduce synthetically generated data sets which represent the two different types of multimodal data sets. The data sets are all 3D data sets, where the slices are duplicates of the same image with a size of $512 \times 512$ pixels. To investigate the influence of noise on our method, we added Gaussian white noise with a standard deviation $\sigma = 1.5\%$ ($SNR = 17.6$) to one modality. In the subsequent sections the synthetic data sets are used to highlight the properties of multimodal similarity maps.

### Supplementary Data

Multimodal data is often used to eliminate the drawbacks of a certain imaging modality. This is necessary when one modality contains undesirable noise or other artifacts in certain regions. In this case a second modality is used to compensate for these artifacts. In this paper we will refer to this type of data as supplementary data. Basically both modalities depict the same structures, but disadvantages of one modality may be compensated by the other and vice versa. An example for supplementary data is dual energy CT. It is used in medicine and industrial applications. The most common artifacts in CT scans in general are noise-induced streaks, beam hardening, partial volume effects, aliasing, and scattered radiation [12, 72]. Due to the fact that different energy levels have different attenuation characteristics, some of these artifacts appear prominently only in one energy level. Hence it is desired to reduce artifacts by the fusion of CT data sets of different energy levels.

We generated two synthetic data sets in order to simulate multimodal data with supplementary characteristics. In Figure 4.2 these two data sets and their histograms are depicted. Due to scaling reasons the frequency of background points with a value of zero is omitted in the histograms. On the right side of Figure 4.2, data set 2 with additional Gaussian white noise is shown. Both data sets contain four squares with gradually changing data values from left to right. Their value ranges are different in both data sets to simulate the effects of varying attenuation characteristics in modalities such as dual energy CT.

### Complementary Data

In some cases, it is also advantageous to combine the information of modalities with more distinct characteristics. In such a scenario, a significant amount of information differs or is not represented in one of the modalities. We will refer to this type of multimodal data as complementary data. Complementary data is commonly encountered in medicine. Modalities such as CT and MRI measure different physical characteristics of the human body, and thus there are substantial differences between two such scans of the same patient. An even more pronounced example is the combination of anatomical and functional modalities, such as CT and PET. There is only a rough correspondence between the two modalities, as CT images contain no functional information at all.

We will use the synthetic data sets illustrated in Figure 4.3 to represent complementary data. Data set 1 contains four squares while data set 2 contains two squares and a circle. The missing square and the circle in data set 2 represent the complementary nature of the data. With the circle we want to show how differences in shape are depicted in the multimodal similarity map. The missing square is used to show the effect if one object is completely omitted from one modality. In the next section these synthetic data sets are used to explain multimodal surface similarity measurement.

## 4.4 Multimodal Surface Similarity

Isosurfaces are important features in volumetric data. An isosurface of a volumetric scalar field $f : \mathbb{R}^3 \to \mathbb{R}$ is the locus of all points in the scalar field at which $f$ attains an *isovalue* $k$:

$$L_k = \left\{ x \in \mathbb{R}^3 : f(x) = k \right\} \tag{4.1}$$

In many cases, different material types correspond to different value ranges in the data set. For example, in medical CT data sets there are typically well-defined intensity ranges associated with soft tissue, fat, and bone. The important characteristic parameter is the intensity isovalue which defines an isosurface representing the boundary of a particular region. Isosurfaces, however, also exhibit a significant amount of redundancy and small variations caused by noise and partial volume effects will result in many similar isosurfaces. Histograms and other isosurface statistics [26, 139] can be used to obtain a better characterization of a data set by depicting distributions of isosurface properties over the range of data values. They are limited, however, in that they treat each isosurface in isolation and therefore cannot capture the spatial relationships between multiple structures.

As an alternative, the measure of isosurface similarity was introduced by Bruckner and Möller [20] to quantify how much information two isosurfaces have in common. They used a matrix of isosurface similarity for all combinations of isovalues within a single data set as the basis for identifying relevant isovalues. We will refer to this method as *self similarity maps* since the measurement of the similarity is between isosurfaces of a single data set. For multimodal data, in particular, it is difficult to investigate differences and similarities based on isosurface statistics as the order and range of corresponding data values may vary significantly. Thus, in order to characterize the correspondences between multiple modalities, we extend the idea to *multimodal similarity maps* which quantify the similarity between all combinations of isosurfaces from two scalar fields.

In the following, we first briefly revisit isosurface self similarity maps as presented by Bruckner and Möller [20] and then describe our extension to multimodal data.

### 4.4.1 Self Similarity Maps

A common measure for similarity is *mutual information*. Mutual information is a basic concept from information theory, measuring the statistical dependence between two random variables or the amount of information that one variable contains about the other. It is a particularly attractive measure because no assumptions are made regarding the nature of this dependence and because of its robustness against perturbations [167]. Therefore, mutual information has been applied in many areas including shape registration [73], multi-modality fusion [57], and viewpoint selection [158]. The mutual information of two discrete random variables $X$ and $Y$ can be defined as [176]:

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{4.2}$$

where $H(X,Y)$ is the joint entropy and $H(X)$ and $H(Y)$ are the marginal entropies of random variables $X$ and $Y$. Since the mutual information is limited by the average marginal entropies, it can be normalized to a value range in $[0,1]$ by [92]:

$$\hat{I}(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)} \tag{4.3}$$

As a measure of isosurface similarity, Bruckner and Möller [20] proposed the normalized mutual information of the respective isosurface *distance fields*. For a given isovalue $k$ and an isosurface $L_k$ the distance field $D_k$ can be defined as follows [77]:

$$D_k(x) = \min_{\forall y \in L_k} d(x,y) \tag{4.4}$$

where $d$ is a distance measure between the points $x$ and $y$. To measure the similarity between two isosurfaces $L_i$ and $L_j$, the distance fields for both isosurfaces $D_i(x)$ and $D_j(x)$ are used as discrete random variables $X$ and $Y$ for the calculation of the mutual information based on Equation 4.3. This leads to a single quantity between 0 and 1 which expresses the similarity between isosurface $L_i$ and isosurface $L_j$. Higher values mean that the isosurfaces are considered to be more similar.

If we consider $N$ different isovalues $V = \{k_1,...,k_N\}$ then the self similarity map can be defined as an $N \times N$ matrix $\mathrm{SSM}(i,j)$. Each element $(i,j)$ of the matrix represents the normalized mutual information for a combination of isovalues $i$ and $j$.

### 4.4.2 Multimodal Similarity Maps

In this paper, we extend the concept of isosurface similarity maps to multimodal data. Instead of investigating the similarity of isosurfaces in a single data set, we explore the similarity of two different data sets representing the same object. The isosurfaces of both modalities are represented by:

$$\dot{L}_k = \left\{x \in \mathbb{R}^3 : \dot{f}(x) = k\right\} \quad \ddot{L}_l = \left\{x \in \mathbb{R}^3 : \ddot{f}(x) = l\right\} \tag{4.5}$$

where $k$ and $l$ are the two isovalues, and $\dot{f}$ and $\ddot{f}$ are the scalar-valued functions representing the two modalities. Based on the two isosurfaces, two distance fields $\dot{D}_k$ and $\ddot{D}_l$ can be generated:

$$\dot{D}_k(x) = \min_{\forall y \in \dot{L}_k} d(x,y) \quad \ddot{D}_l(x) = \min_{\forall y \in \ddot{L}_l} d(x,y) \tag{4.6}$$

Figure 4.4 illustrates how the mutual information for a combination of isovalues $l$ and $k$ is calculated. The first step is the generation of the distance fields $\dot{D}_k$ and $\ddot{D}_l$ for the isosurfaces $\dot{L}_k$ and $\ddot{L}_l$. In the next step the distances $\dot{D}_k(x)$ and $\ddot{D}_l(x)$ for each point $x$ in the volume space are used to generate a *joint distance histogram*. The joint distance histogram represents the joint probability for a point $x$ to have the distance $\dot{D}_k(x)$ to isosurface $\dot{L}_k$ and $\ddot{D}_l(x)$ to isosurface $\ddot{L}_l$. In Figure 4.4 an example of a joint distance histogram is shown for two identical isosurfaces. In this case, all points $x$ in the volume space have the same distance to $\dot{L}_k$ and $\ddot{L}_l$.

Finally, the mutual information is calculated based on Equation 4.3. The joint and marginal probabilities for the calculation of the joint and marginal entropies can be directly retrieved from the joint distance histogram. For the example in Figure 4.4 the mutual information of isosurfaces for the isovalues $k$ and $l$ is maximal since the isosurfaces are identical.

To generate the entire multimodal similarity map, the steps in Figure 4.4 are repeated for every possible combination of isovalues in both modalities. If we assume that modality 1 has $N$ different isovalues $\dot{V} = \{k_1, ...k_N\}$ and modality 2 has $M$ different isovalues $\ddot{V} = \{l_1, ..., l_M\}$ then the multimodal similarity map can be defined as an $N \times M$ matrix $\mathrm{MSM}(i, j)$. Each entry of the multimodal similarity map represents the similarity between the isosurface $\dot{L}_i$ of modality 1 with the corresponding isovalue $i$ and the isosurface $\ddot{L}_j$ of modality 2 with the corresponding isovalue $j$. On the right side of Figure 4.4 the complete multimodal similarity map for the example data sets is shown. The dark line represents the combinations of isovalues $i$ and $j$ which result in identical isosurfaces $\dot{L}_i$ and $\ddot{L}_j$. For all other combinations of isovalues the similarity of their corresponding isosurfaces is lower.

Figures 4.5 and 4.6 show the multimodal similarity maps for the synthetic data sets introduced in Section 4.3. Dark regions denote a high similarity in these figures. For both types of multimodal data, the similarity maps for the combination of data sets without noise and with noise are shown.

For the supplementary data in Figure 4.5, both data sets contain four squares at the same locations. In the MSM each of the squares is represented by a rectangular area of higher similarity. In Figure 4.5 corresponding squares and rectangular regions are emphasized by colored frames. The band with the maximum similarity represents the combinations of isovalues $k$ and $l$ at which both data sets represent exactly the same isosurfaces. Due to different value ranges in both data sets this band does not follow the diagonal of the multimodal similarity map. In contrast to self similarity maps, multimodal

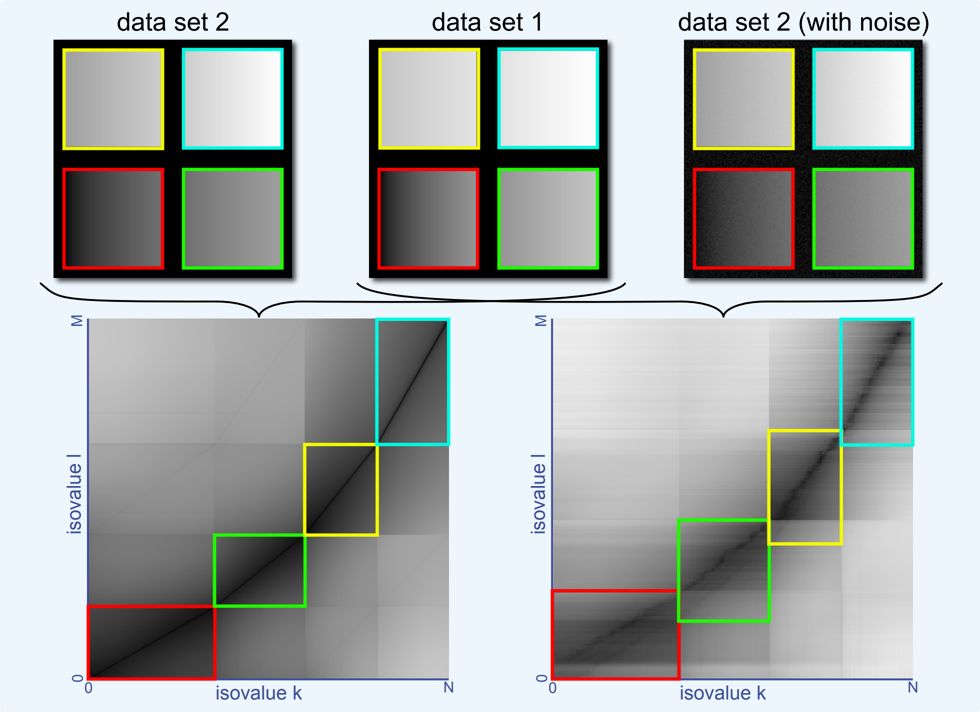


**Figure 4.5 –** Multimodal similarity maps for supplementary data.

similarity maps are not symmetrical along the main diagonal. If we investigate the influence of noise in the multimodal similarity map on the right side of Figure 4.5, it can be seen that the band with the higher similarity is expanded. The expansion of the band gets smaller the higher the data values are. This is due to a higher *SNR* and therefore a smaller impact of the noise on the similarity measurement for higher data values.

In Figure 4.6 the multimodal similarity maps for our complementary test data sets are shown. The regions in which both data sets contain contradictive information are clearly visible in the similarity map. In contrast to Figure 4.5, the lower left rectangular area (red frame) in Figure 4.6 has a considerably lower similarity. Furthermore the band with maximum similarity is missing since there are no isosurfaces for the corresponding isovalues in data set 2 as one square is completely omitted in data set 2. In the same area of the multimodal similarity map with the noisy data set we get a higher variation of similarity values. This is due to the similarity between the square in data set 1 and structures generated by the noise in the background areas of data set 2.

Another interesting area in the multimodal similarity map of Figure 4.6 is the rectangle in the upper right corner (cyan frame). This rectangular area represents the similarity between the square in one data set and the circle in

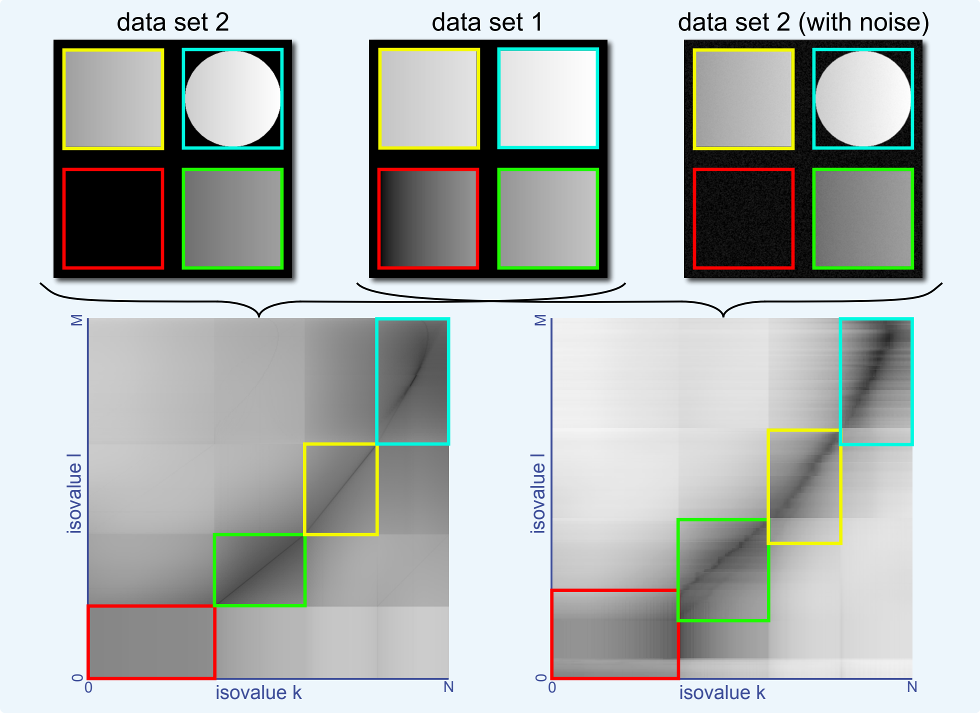


**Figure 4.6 –** Multimodal similarity maps for complementary data.

the other data set. Because of the different shapes of the objects the isosurfaces are similar but not identical. In the similarity map this can be seen by the expanded band of maximum similarity.

## 4.5 Similarity-Based Multimodal Volume Visualization

In this section, we discuss how the additional information provided by multimodal similarity maps can guide the process of exploring and analyzing multimodal volume data. It directs the user towards regions of high similarity or dissimilarity among the two modalities. We first show how salient regions in the similarity map can assist the user in identifying features. Next, we describe a simple approach for providing insight into the spatial differences in a multimodal data set by automatically identifying the most similar isosurfaces in two modalities. Finally, we present a novel approach for similarity-based classification of multimodal volume data.

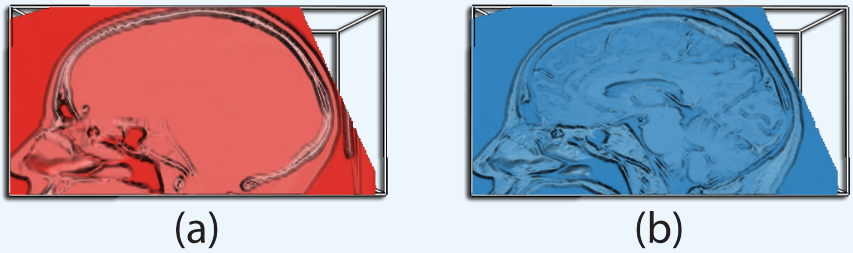
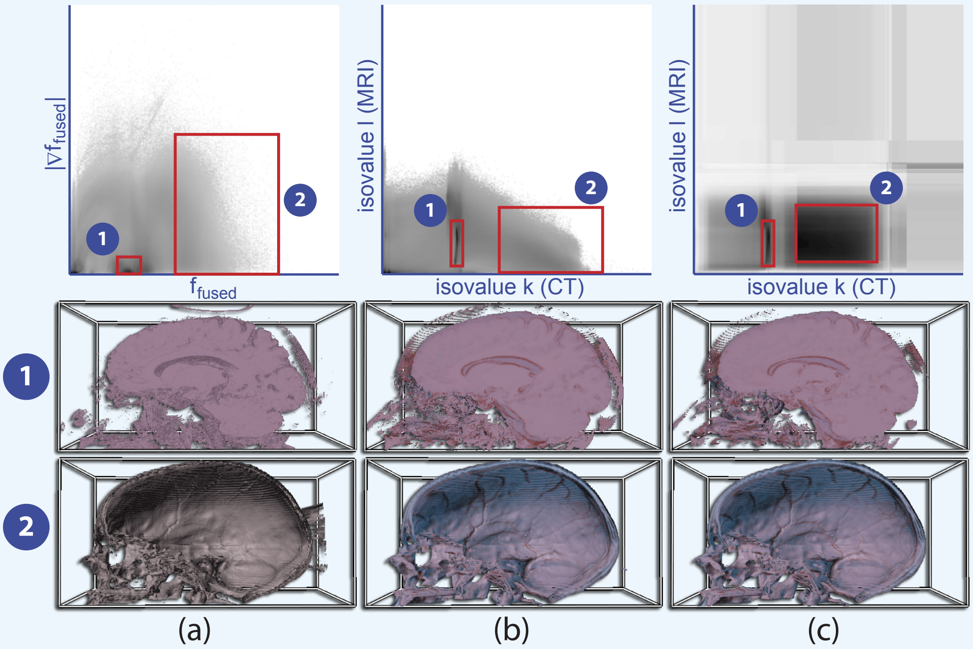

(a) (b)

**Figure 4.7 –** Clipping plane through a (a) CT and (b) MRI scan of a human brain.



(a) (b) (c)

**Figure 4.8 –** Comparison of (a) a fused transfer function space [57], (b) a dual histogram, and (c) a multimodal similarity map for selection guidance. Region 1 classifies brain tissue and region 2 classifies the cranial bone.

### 4.5.1 Similarity-Based Exploration

Multimodal similarity maps can be used to enable better selection of features in multimodal volume data in a straightforward manner. As the coordinate system of the similarity map is defined by the data values in both modalities, a simple approach is to allow the user to select a region of interest (e.g., by specifying a rectangular selection) and to restrict the visualization to data values which lie within this range. The color and opacity maps are defined separately for each modality.

To demonstrate the advantages of similarity maps over other techniques, we choose a common example: the combination of CT and MRI data. While

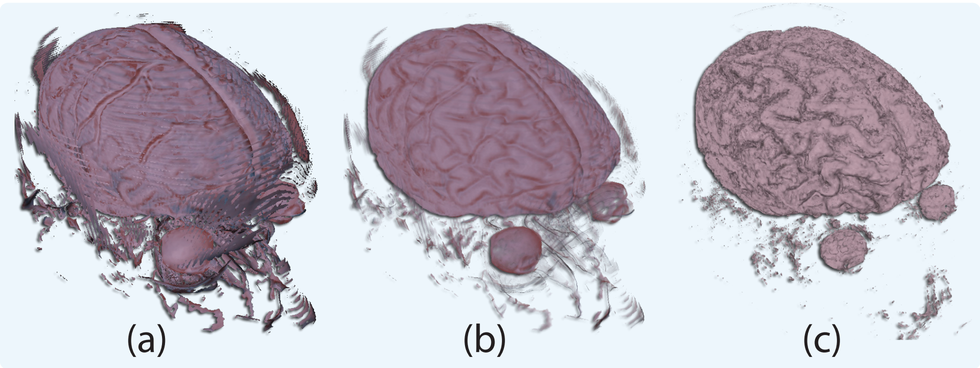


**Figure 4.9 –** Selection of brain tissue (a) without similarity weighting, (b) with similarity weighting, and (c) using the method of Haidacher et al. [57] after manually adjusting their $\delta$ weighting function to achieve the optimal result.

CT offers a standardized scale for identifying certain types of tissue, MRI provides significantly higher contrast in soft tissue regions. Figure 4.7 shows (a) a CT and (b) an MRI data set each rendered using a simple linear color map. MRI depicts more details in the brain tissue. But bone, due to its low water content, cannot be distinguished from air. In the CT scan, on the other hand, bone can be clearly identified.

We compare our method with the multimodal transfer function space presented by Haidacher et al. [57] as well as a simple dual histogram. The transfer function space of Haidacher et al. [57] presents a fused data value on the horizontal axis and a fused gradient magnitude on the vertical axis. The fusion is performed based on point-wise mutual information. However, in contrast to our approach this measure is not based on spatial information, but only on the estimated probability of occurrence of a data value combination. In the dual histogram, the frequency of each data value combination is represented by the intensity of the corresponding pixel with darker regions corresponding to higher frequencies. The resulting parameter spaces are depicted in the top row of Figure 4.8, where (a) shows the fused transfer function space, (b) depicts the dual histogram, and (c) presents the multimodal similarity map. It can be seen that all methods give a salient representation for regions corresponding to brain tissue. However, both the fused transfer function space and the dual histogram fail to give a clear indication of bone as they do not take into account spatial information. The data value ranges corresponding to bone are vastly different in CT and MRI data. Using the multimodal similarity map, on the other hand, a region corresponding to bone can be easily identified. The middle and bottom rows of Figure 4.8 show 3D visualizations of the data value ranges corresponding to the highlighted selection regions. In the case of the fused transfer function space and the dual histogram, the selection rectangle for bone had to be placed by trial-and-error.

The previous example employed a binary selection in the multimodal similarity map. In order to exploit the information provided by the similarity map we can further use the similarity directly to modulate the opacity of a sample within the selected region:

$$A'(x) = A(x)\frac{\mathrm{MSM}(\dot{f}(x), \ddot{f}(x)) - \mathrm{MSM}_{\mathrm{min}}}{\mathrm{MSM}_{\mathrm{max}} - \mathrm{MSM}_{\mathrm{min}}} \tag{4.7}$$

where $A'(x)$ is the modulated opacity, $A(x)$ is the original opacity, and $\dot{f}(x)$ and $\ddot{f}(x)$ denote the data value of modality 1 and modality 2, respectively, at a sample position $x$. $\mathrm{MSM}_{\mathrm{min}}$ and $\mathrm{MSM}_{\mathrm{max}}$ are the minimum and maximum similarity values in the selected region. The result of this weighting is an enhancement of similar structures in both modalities while dissimilar structures are suppressed.

Figure 4.9 illustrates the effect of this weighting. In Figure 4.9 (a) no weighting is applied, while Figure 4.9 (b) shows the result obtained with weighting. For comparison, Figure 4.9 (c) depicts the method of Haidacher et al. [57] after manually adjusting their $\delta$ weighting function. The selected regions used for the fusion correspond to the regions for the brain in Figure 4.8. The results in Figure 4.9 show that similarity weighting produces comparable results to Haidacher et al. [57] without the necessary user interaction for the adjustment of the $\delta$ windowing function.

This example illustrates how multimodal similarity maps can be used to provide assistance in identifying features across multiple modalities. A main advantage over the method of Haidacher et al. [57] is that the original data values can be retained instead of combining them during preprocessing. Furthermore, we do not introduce a new transfer function space which may be unfamiliar to users and difficult to understand.

### 4.5.2 Maximum Similarity Isosurfaces

With the multimodal similarity map we gain information about the similarity of certain combinations of isovalues. The multimodal similarity maps for the examples in Section 4.4 have shown that combinations of isovalues for isosurfaces with a high similarity can be identified easily even if their ranges differ significantly. In many applications, such as industrial CT, users want to compare how well the object of interest is depicted in both modalities. Finding the isovalues which best represent the structure of interest in both scans, however, is difficult and requires time-consuming manual tuning.

Using the multimodal similarity map, we can automatically identify the isovalue for the isosurface in one modality which maximizes the similarity to a specific isosurface from another modality. If we assume that a user has specified an isovalue $k$ for an isosurface in one modality, the isovalue $\hat{k}$ with

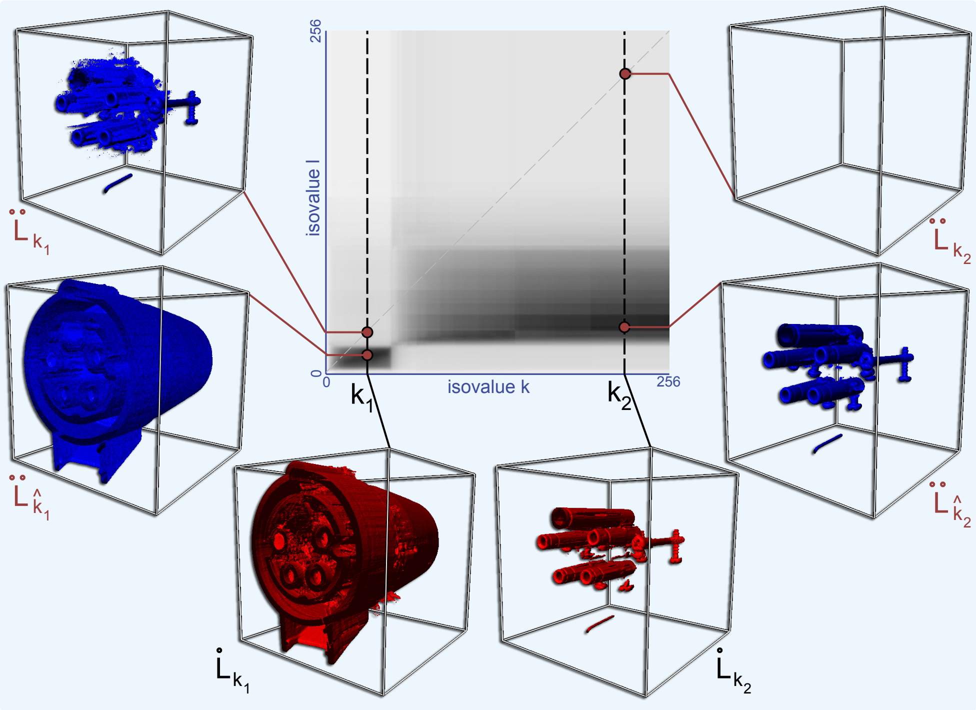


**Figure 4.10 –** Maximum similarity isosurface detection for two different isovalues $k_1$ and $k_2$. The results in the middle row show the most similar isosurfaces ($\ddot{L}_{\hat{k}_1}$, $\ddot{L}_{\hat{k}_2}$). The results in the top row show the isosurfaces for a naive selection of the isovalues, i.e., in both data sets the same isovalue is chosen.

the most similar isosurface in the second modality can be obtained by:

$$\hat{k} = \arg\max_{j} \mathrm{MSM}(k, j) \tag{4.8}$$

Using this simple approach, it is possible to specify an arbitrary isovalue in either modality and instantly visualize the corresponding isosurfaces from both modalities. A typical setup may depict these isosurfaces side-by-side in linked views enabling the user to quickly identify the spatial differences between two volumetric data sets by browsing through the range of isovalues.

Figure 4.10 depicts an example for a dual energy CT scan. Due to the different attenuation characteristics for different energy levels, the value ranges in both data sets are different. This can be seen in the multimodal similarity map in the center of Figure 4.10. The images in the bottom row show isosurfaces for isovalues $k_1$ and $k_2$ in modality 1. The top row shows the isosurfaces for the same isovalues in modality 2. The middle row shows the isosurfaces in modality 2 for the isovalues $\hat{k}_1$ and $\hat{k}_2$ with the maximum similarity to $k_1$ and $k_2$. The isosurfaces for $\hat{k}_1$ and $\hat{k}_2$ match the isosurfaces in modality 1 much better than the isosurfaces for the naive selection of isovalues.

### 4.5.3 Similarity-Based Classification

Selection of simple regions in the multimodal similarity map, as described in the Section 4.5.1, allows quick exploration of multimodal data. This approach can be useful when only few specific features are of interest. For generating more complex visualizations, which depict multiple volumetric structures and take advantage of the additional information provided by multiple modalities, classification in the joint data space is necessary. The multimodal similarity map also opens up new avenues to assist in this process. Our idea is to use a nearest neighbor classifier in similarity space to determine the optical properties of a sample. Intuitively, instead of trying to relate the two modalities in terms of their data values, we instead classify samples, i.e., combinations of data values from both modalities, according to their similarity to a set of user-specified isosurfaces from both modalities.

We assume two continuous three-dimensional scalar fields $\dot{f}, \ddot{f} : \mathbb{R}^3 \to \mathbb{R}$ which represent two co-registered input volumes. For multimodal volume visualization, we assign a color and opacity to every point $x \in \mathbb{R}^3$ in space based on the value of these functions. Our method takes as input a set of isovalue pairs $h_i = (\dot{h}_i, \ddot{h}_i)$ where $\dot{h}_i$, $\ddot{h}_i$ correspond to isovalues of $\dot{f}$ and $\ddot{f}$ respectively. Each pair of isovalues has an assigned color $c_i$, opacity $\alpha_i$, and optional weight $w_i$.

For two data values $k \in \dot{f}$ and $l \in \ddot{f}$, we evaluate their multimodal similarity to the $i$-th isovalue pair in the following manner:

$$\dot{s}_i(k) = \mathrm{MSM}(k, \ddot{h}_i) \quad \ddot{s}_i(l) = \mathrm{MSM}(\dot{h}_i, l) \tag{4.9}$$

where MSM is the multimodal similarity map. This means that $\dot{s}_i$ is the similarity of the isosurface $k$ of $\dot{f}$ and the isosurface $\ddot{h}_i$ of $\ddot{f}$ and $\ddot{s}_i$ is the similarity of the isosurface $l$ of $\ddot{f}$ and the isosurface $\dot{h}_i$ of $\dot{f}$.

Based on the similarities $\dot{s}_i$ and $\ddot{s}_i$ we can now define a combined measure $s_i$ of similarity between $h_i$ and the two isovalues $k \in \dot{f}$ and $l \in \ddot{f}$ in multimodal similarity space:

$$s_i(k,l) = \dot{s}_i(k)\ddot{s}_i(l) \tag{4.10}$$

The rationale behind this choice is that we interpret the similarities $\dot{s}_i(k)$, $\ddot{s}_i(l)$ as independent probabilities of $k$ being similar to $\ddot{h}_i$ and $l$ being similar to $\dot{h}_i$. Thus, the joint probability of $(k,l)$ being similar to $h_i$ is the product $\dot{s}_i(k)\ddot{s}_i(l)$. Alternatively, we could consider $\dot{s}_i$ and $\ddot{s}_i$ as the membership functions of two fuzzy sets and $s_i$ as the membership function of their intersection. In this case, another possible definition would be $s_i(k,l) = \min(\dot{s}_i(k), \ddot{s}_i(l))$ [178]. In our experiments, we found that both approaches lead to similar results.

Having defined a measure of closeness between two points in similarity space, we now let each pair of isovalues $h_i$ to determine the optical properties of points that are closer to $h_i$ than to any other isovalue pair $h_j$ $(i \neq j)$. This means a pair of data values $(k,l)$ with $k \in \dot{f}$, $l \in \ddot{f}$ will assume the color and opacity of the isovalue pair $h_{m(k,l)}$ which maximizes $s_i(k,l)$:

$$m(k,l) = \arg\max_i s_i(k,l) w_i \tag{4.11}$$

where $w_i$ is a weight which allows additional control over the influence of the isovalue pair $h_i$. During rendering, we can now evaluate this maximum for every sample location $x \in \mathbb{R}^3$ in space:

$$m_x = m(\dot{f}(x), \ddot{f}(x)) \tag{4.12}$$

Thus, $m_x$ denotes the index of the isovalue pair which maximizes the similarity to the data value $\dot{f}(x)$, $\ddot{f}(x)$ at the sample location $x$. To visually encode the similarity of the sample to $h_{m_x}$, we additionally weight the sample opacity based on the similarity $s_{m_x}$. The color $C(x)$ and opacity $A(x)$ at the sample position $x$ are then simply:

$$C(x) = c_{m_x} \quad A(x) = \alpha_{m_x} s_{m_x} \tag{4.13}$$

In practice, in order to obtain crisp boundaries, it is convenient to define an additional threshold $t$ which specifies the minimum similarity of a sample with any of the isovalue pairs in order to be visible. If $s_{m_x} < t$, the sample is considered to be fully transparent.

In volume rendering, it is common to evaluate a local illumination model using the normalized gradient of the scalar field as the normal vector. To enable volume shading, we can combine the gradient information of both modalities using a similarity-based weighting:

$$g(x) = \frac{\dot{s}_{m_x}(\dot{f}(x))\nabla\dot{f}(x) + \ddot{s}_{m_x}(\ddot{f}(x))\nabla\ddot{f}(x)}{\dot{s}_{m_x}(\dot{f}(x)) + \ddot{s}_{m_x}(\ddot{f}(x))} \tag{4.14}$$

### 4.5.4 Classification Specification

The described classification is equivalent to a generalized Voronoi decomposition of similarity space, i.e., using non-Euclidean distances defined by our similarity measure. Every sample, which is a pair of values from the two modalities, is assigned to the most similar isovalue pair which determines its color and opacity. We also visualize this classification on the similarity map itself by simply evaluating Equation 4.11 for each location, i.e., each combination of data values, in the similarity map and coloring the corresponding pixel accordingly. When depicted on the two-dimensional similarity map, where the coordinate system is defined by data values, these regions may be disconnected and non-convex (see, for example, Figures 4.11 and 4.12 which are discussed in detail below). Furthermore, based on the structure of the similarity map, the site, i.e., the isovalue pair that defines a region may not be contained within this region. While this may initially sound counter-intuitive,

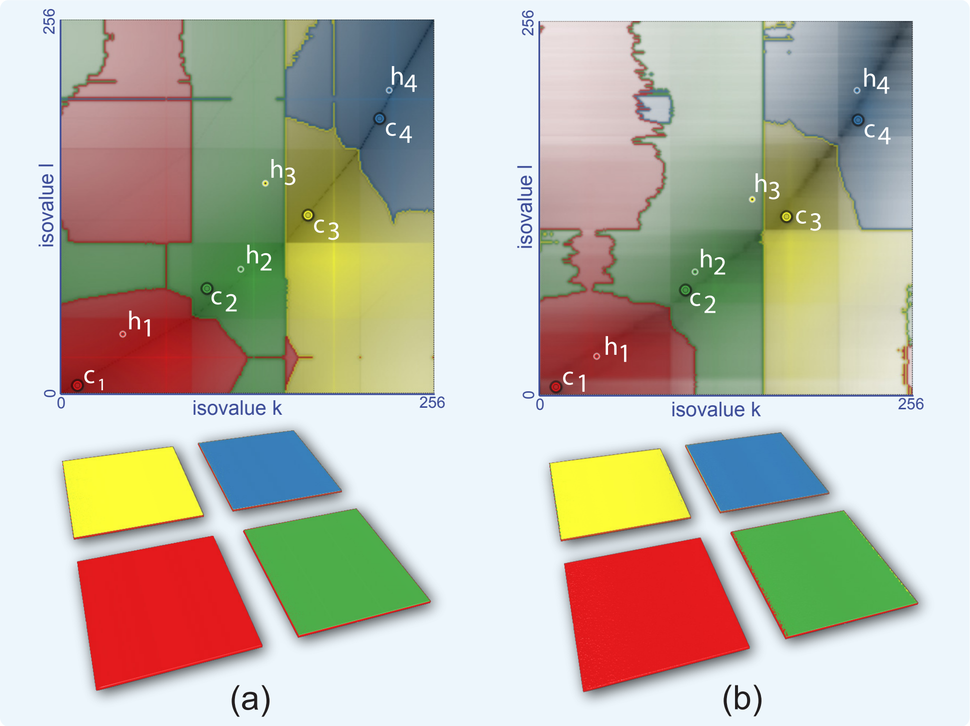


**Figure 4.11 –** Classification based on multimodal surface similarity for supplementary data (a) without noise and (b) with added noise.

the following situation exemplifies such a case: Assume two isosurfaces for $\dot{h}_i$ and $\ddot{h}_i$ which are highly dissimilar. There will likely be other isosurfaces they are more similar to than to each other.

To provide an additional means for manipulating the classification regions instead of directly modifying the isovalues themselves, we define a user-specified control point $c_i = (\dot{c}_i, \ddot{c}_i)$ for each isovalue pair $h_i$, which can be freely moved. $h_i$ is initialized with $c_i$ and is then used to compute the similarity-weighted centroid of the region it defines. The isovalue pair $h_i$ is then moved to the position of the centroid:

$$h_i = \frac{\sum\limits_{(k,l)\in R(c_i)} (k,l)\, s_{m(k,l)}(k,l)}{\sum\limits_{(k,l)\in R(c_i)} s_{m(k,l)}(k,l)} \tag{4.15}$$

where $R(c_i) = \{(k,l) | m(k,l) = i\}$ is the similarity-space region assigned to $c_i$. This essentially corresponds to one iteration of Lloyd's algorithm [97]. Note, however, that we do not perform the full relaxation since our goal is not to perform a full centroidal decomposition of the similarity space. Instead, our aim is for regions to follow their control points.

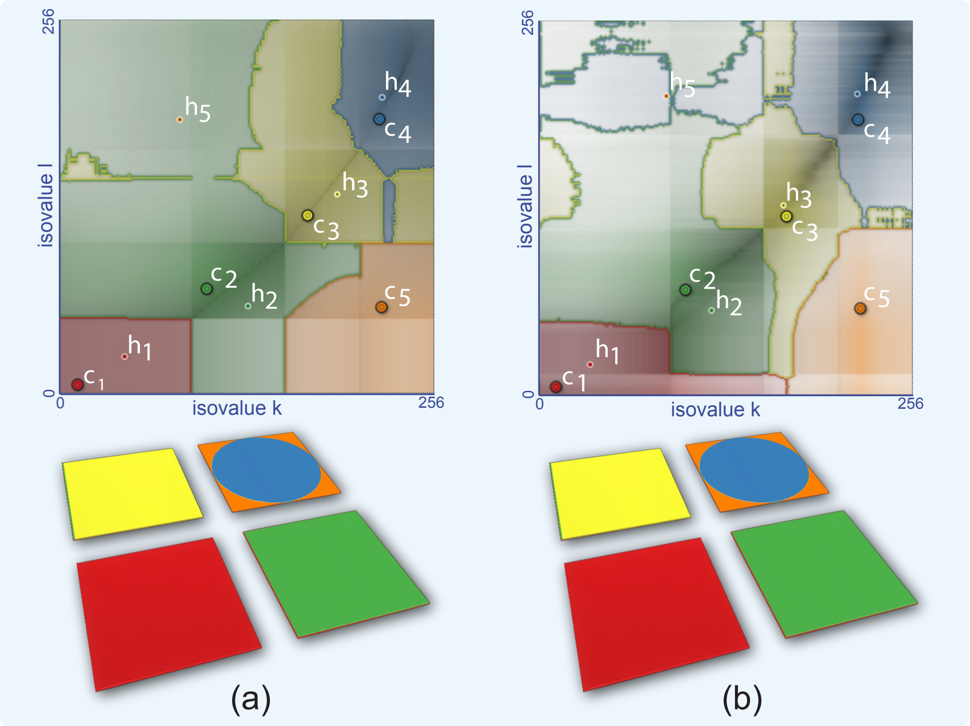


**Figure 4.12 –** Classification based on multimodal surface similarity for complementary data (a) without noise and (b) with added noise.

Based on this approach, we developed a simple user interface for similarity-based classification of multimodal volume data. The user is presented with the multimodal similarity map and can interactively add and remove control points, move them on the similarity map, and change their colors and opacities. When moving control points they behave similar to well-known ”magic wand”-type selection tools – the regions they define snap to clusters in the similarity map. Slightly modifying a control point will, in accordance with the structure of the similarity map, not cause major changes of the classification result.

Figures 4.11 and 4.12 show examples of our classification approach using the previously introduced synthetic data sets. The colored regions encode the nearest neighbors, in similarity space, of each of the white-outlined points for isovalue pairs $h_i$ in the corresponding color. This means a point on the similarity map is assigned to a region if it is more similar to the corresponding isovalue pair than to any other one. This also means that during volume rendering a sample with the corresponding value combination will be assigned the respective color and opacity (see Equation 4.13). The control points $c_i$ are depicted as the slightly larger points with dark outlines. It can be seen that in regions of high similarity the control points $c_i$ will be close to the corresponding isovalue pairs $h_i$, but in other areas this is not necessarily the case. Figure 4.11 (a) illustrates that our approach is successful in identifying the correspondences between both data sets. By placing control points along the band of maximum similarity, the resulting regions will subdivide the map such that all rectangles in the data are assigned the user-specified color even though their data value ranges vary. Small perturbations in the placement of the control points leave the resulting classification unaffected. Adding noise to one of the data sets has little effect on the visual result, as shown in Figure 4.11 (b). Figure 4.12 shows that this approach makes it possible to easily exploit complementary information from both data sets. The red square, which is only present in one data set, as well as the blue circle and orange square can all be separated. This enables the generation of a combined visualization which contains all these features. Using a conventional approach, where the user defines a 1D transfer function for each modality and the results are blended, it is much more difficult to separate the features, since their corresponding data value ranges significantly vary. The additional weights $w_i$ used in Equation 4.11 therefore allow to control the sizes of the respective regions and can be interactively modified. In Figures 4.11 and 4.12 all weights $w_i$ are set to one.

## 4.6 Results

An application for which our similarity-based classification approach is particularly suitable is the study of industrial parts where the goal is to detect manufacturing defects. Dual energy CT is of interest in such scenarios, since different materials can cause scanning artifacts at certain energy levels. The low energy scan typically has high precision but is affected by severe artifacts, while the high energy scan is nearly artifact-free but suffers from reduced precision and noise. It is desirable to combine the advantages of both energy levels, i.e., to generate a visualization which uses the global structure from the high energy scan to remove the artifacts from the low energy scan while preserving subtle details. Heinzl et al. [68] presented a processing pipeline for extracting surfaces from dual energy CT scans. With our similarity-based classification approach it is now possible to directly visualize structures which exhibit high surface similarity between both modalities. An example is given in Figure 4.13. Figure 4.13 (a) shows the low energy scan of a 400 Volt power connector rendered using a conventional 1D transfer function. It is not possible to find an opacity setting which suppresses all artifacts but leaves the surface intact. In Figure 4.13 (b) the corresponding high energy scan, also rendered using a 1D transfer function, is shown. This result gives a better impression of the actual surface, but is noisy and lacks details. Using our similarity-based classification approach, as shown in Figure 4.13 (c), we can remove the artifacts by choosing control points which select regions of high

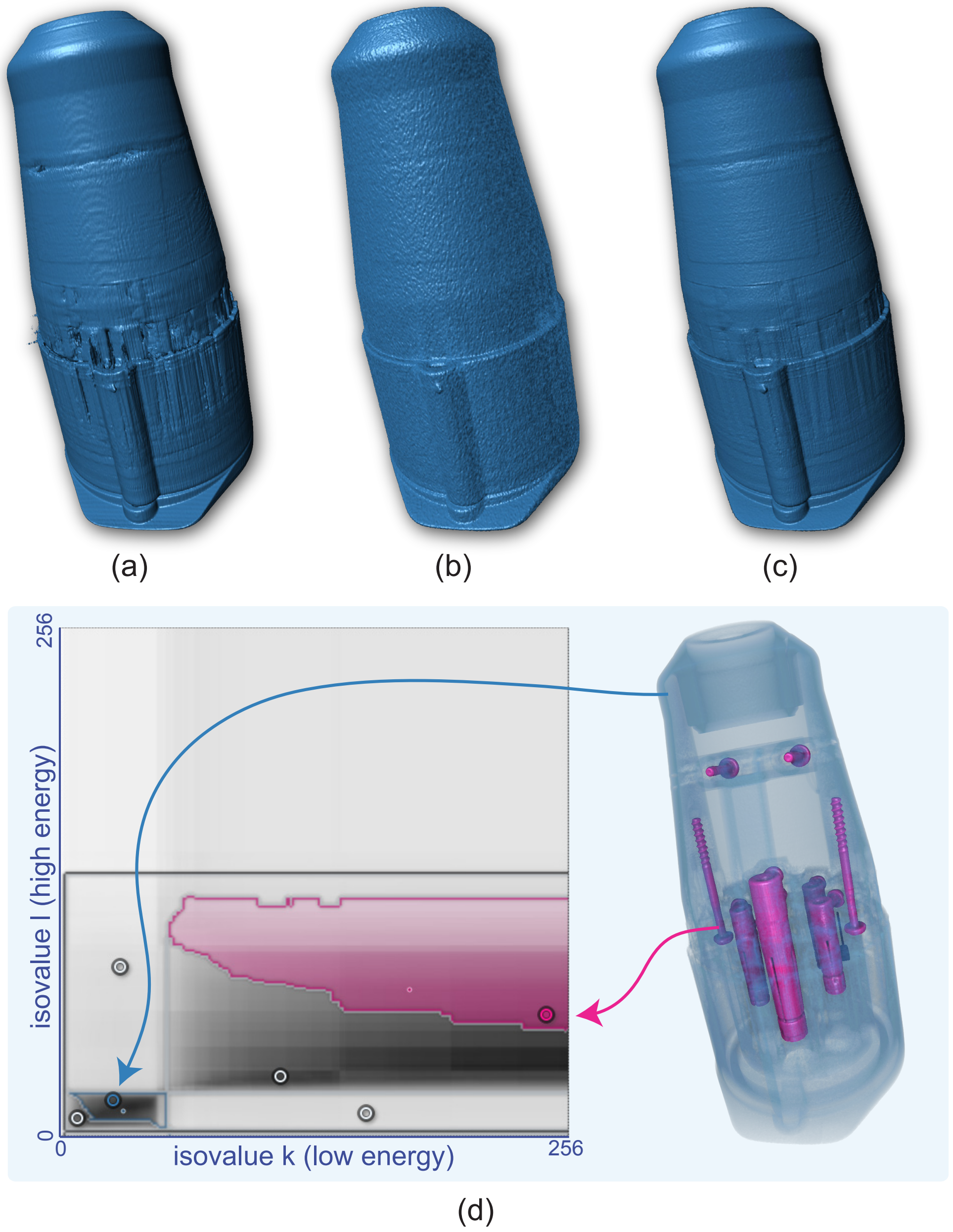


**Figure 4.13 –** Similarity-based fusion of a dual energy CT scan of a power connector. The low-energy scan (a) and the high-energy scan (b) provide supplementary information which can be used to remove most of their respective drawbacks, as shown the similarity-based classification (c). The corresponding similarity map (d) shows the placement of the control points. In the image to its right the opacity of the outer surface has been reduced to reveal the interior parts of the connector.

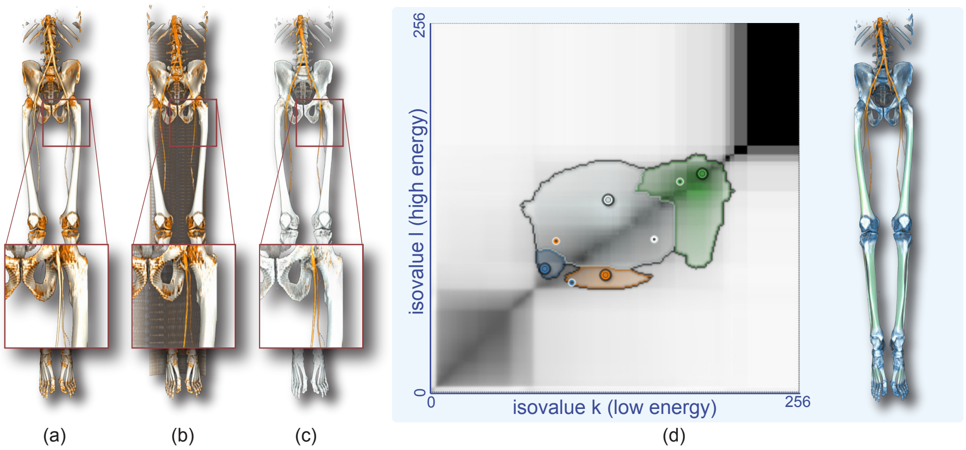


**Figure 4.14 –** Similarity-based classification of blood vessels in a dual energy CT angiography scan of the lower extremities. When using only the information from the low energy scan (a) or the high energy scan (b), it is not possible to separate blood vessels and bones. Using multimodal similarity (c) this can be achieved. The corresponding control points are shown on the similarity map (d) where different colors have been assigned to vessels and bones of different densities.

dissimilarity and setting their opacity to zero. The feature of interest, the outer surface of the connector, is similar in both data sets. Since the opacity of a sample is based on the global surface similarity of its data values to the isosurface pair, holes and artifacts in the low energy scan can be remedied using information from the high energy scan. Figure 4.13 (d) shows the similarity map together with the specified control points. The image to the right of the similarity map uses the same control points as Figure 4.13 (c), but the opacity of the outer surface has been lowered to reveal the interior parts of the connector.

Furthermore, our approach can be used to assist the classification of ambiguous structures. One example is CT angiography, where it is desired to clearly separate contrast-enhanced blood vessels from bone. In a single modality scan this is typically not possible as the data values of the contrast agent partially overlap with lower-density bone regions and cartilage. This is illustrated in Figure 4.14 (a), where it was attempted to specify different colors for bones and vessels using a 1D transfer function on a low energy CT scan of the lower extremities. While it is also not possible to achieve this separation using a high-energy scan, as shown in Figure 4.14 (b), it can be seen that the classified structures are slightly different in both modalities. Using similarity-based classification, we can therefore achieve a better separation, as depicted in Figure 4.14 (c). The corresponding similarity map is shown in Figure 4.14 (d). In order to illustrate how regions in the similarity map correspond to structures of different ossification levels, they have been assigned different colors in the rightmost image. Vessels are orange and different bone structures are white, blue, and green.

A further example is shown in Figure 4.1. In this case, a dual energy CT data set of a human head is used. The similarity map, shown on the bottom right, provides good guidance for iteratively selecting the individual tissues numbered from 1 to 7. The information provided by the two energy levels is sufficient to allow differentiation between bone (selected in step 3), major vessels (step 4), and minor vessels (step 5).

## 4.7 Implementation

The calculation of the multimodal similarity map is a preprocessing step which is implemented in C++ and runs on the CPU. It has to be performed only once for a single multimodal data set. After the preprocessing step the multimodal similarity map is simply represented as a two-dimensional image. During rendering, the similarity of a combination of isovalues from the two modalities can be retrieved by a single lookup in a 2D texture.

The user interface for our similarity-based classification approach was implemented using the Qt toolkit. The user-interface widget generates a set of isovalue pairs, colors, and weights, which are passed to a GPU-based volume renderer implemented in GLSL. In the shader, the similarity between the data values at the current sample point and each isovalue pair is determined using two texture lookups (see Equations 4.9 and 4.10) and the maximum is computed. While this is more expensive than a conventional transfer function lookup, the additional costs are limited due to the fact than only few control points will be required in many applications. In our implementation, for a typical number of five control points, the average render time increases by a factor of approximately 1.4 compared to a single conventional transfer-function lookup. The color and opacity of the maximally similar isovalue pair then determines the color and opacity of the current sample, as described in Section 4.5.3. The gradient vectors of both modalities are computed by central differences, combined with Equation 4.14, and used to evaluate a local illumination model if shading is enabled.

## 4.8 Discussion

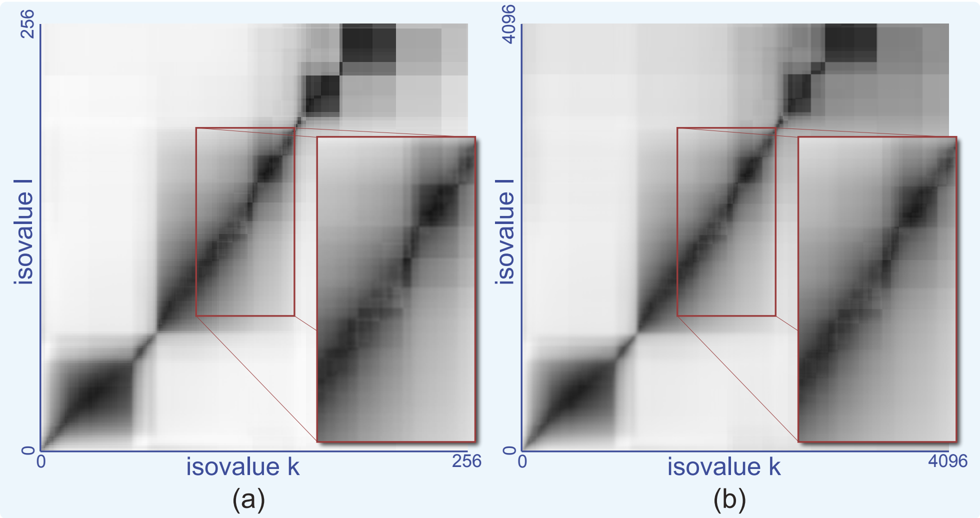
As shown in our examples, multimodal surface similarity can provide a useful tool for the visual analysis of multimodal volume data. However, isosurface similarity as a measure is only useful in cases where there is some correspondence between features and isosurfaces. For example, in data where textures or patterns are of central importance, isosurface similarity will likely fail to provide valuable insights. While this is a clear limitation of our approach, we want to emphasize that also the lack of distinct structures in a multimodal

**Table 4.1 –** Computation times for the multimodal similarity maps shown in the paper as measured in an Intel Core i7 950 CPU with a clock rate of 3.07 GHz and 12 GB RAM. The first column reports the data type and size. The second column gives the downsampling rate for the distance fields. The last column gives the total computation time for the similarity maps which is the sum of the computation times for all distance transforms (third column) and the mutual information of all isovalue combinations (fourth column).

| Data Set | Downsample Rate | Distance Transform | Mutual Information | Total |
|---|---|---|---|---|
| Supplementary<br>$512 \times 512 \times 6$ | 2 | 28.57s | 32.97s | 61.54s |
| Complementary<br>$512 \times 512 \times 6$ | 2 | 21.17s | 32.41s | 53.58s |
| CT-MRI<br>$256 \times 256 \times 128$ | 4 | 4.64s | 32.86s | 37.50s |
| Industrial DECT<br>$425 \times 551 \times 895$ | 16 | 2.12s | 12.08s | 14.20s |
| Head DECT<br>$512 \times 512 \times 575$ | 16 | $2.62s$ | $11.30s$ | $13.92s$ |
| Extremities DECT<br>$512 \times 512 \times 855$ | 16 | $2.86s$ | $14.55s$ | $17.41s$ |

similarity map provides additional information to the user. As our approach deliberately avoids to position itself as a new technique central to the visualization process, the lack of distinct features (like the lack of distinct features in a histogram) simply means that little additional guidance can be provided for the particular data set. However, in our experiments we found that even for challenging data combinations, such as CT and PET, which exhibit little correspondence, multimodal surface similarity is still able to assist in finding joint data value ranges which correspond to joint structures of interest.

The computation time for the multimodal similarity map of two data sets is approximately twice the computation time of a self similarity map for a data set of the same size. This is due to the lack of symmetry. As reported by Bruckner and Möller [20], a feasible strategy to limit the duration of this pre-processing step is to use downsampled versions of the distance transforms (which are computed at the original data set resolution) for the mutual information computation. The computation times for all data sets used in this paper are given in Table 4.1. The second column in the table lists the downsampling rate for the respective volume which is automatically chosen to limit the computation time to approximately one minute. Even though downsampling is performed quite aggressively, a distance field is a rather redundant representation and the downsampled version essentially acts as a shape descriptor and is not used for precise spatial measurements.



(a) (b)

**Figure 4.15 –** A comparison between the multimodal similarity map with a isovalue precision of 8 bits (a) and 12 bits (b) for the data set shown in Figure 4.1.

To the results of Bruckner and Möller [20] we can also add information about additional experiments on the effects of quantization in the value domain. We found that for real-world data a quantization to 8 bits results in practically no structural differences in the similarity map, as exemplified in Figure 4.15.

Another limitation of our work is that the described approach only considers data sets consisting of two modalities. While this applies to many application scenarios, a solution for a larger number of modalities would be desirable. A multi-dimensional similarity map of similarities between all isovalue combinations of the respective data sets, however, would be computationally infeasible. A potential solution could be to only consider pair-wise similarities between the individual modalities resulting in a matrix of multimodal similarity maps. The investigation of whether such an approach is effective is an interesting topic for future research. Furthermore, our technique could also be applied to investigate time-dependent data by generating a set of similarity maps between subsequent time steps. Temporal similarity maps could help to identify stable features and to pinpoint discontinuities.

A further limitation of similarity maps in general is that they do not contain frequency information, i.e., small structures which exhibit high similarity receive the same prominence as very large regions with a similar degree of similarity. This can be regarded as an advantage with respect to histograms, where large regions tend to dominate and logarithmic scaling is typically required. It can also be a drawback since data value combinations which do not occur at all are not clearly indicated. Ideally, a combination of both types of information would be desired, but identifying a good visual encoding for this purpose is not straightforward and remains an area of future research.

## 4.9 Conclusion

In this paper, we introduced multimodal surface similarity maps as a tool for the investigation of multimodal volume data sets. The multimodal similarity map provides an overview of the differences and similarities between the isosurfaces of two modalities in a compact manner. The analysis of parameter spaces is an increasingly important topic for knowledge discovery in scientific data. Our approach showed that spatial similarity information can assist the visualization process by guiding the selection of features. By exploiting similarity information, we introduced a novel way for the interactive classification and visualization of multimodal volume data.

**The following chapter was originally published as:**

S. Bruckner, V. Šoltészová, M.E. Gröller, J. Hladůvka, K. Bühler, J. Y. Yu, and B. J. Dickson. BrainGazer – visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1497–1504, 2009.

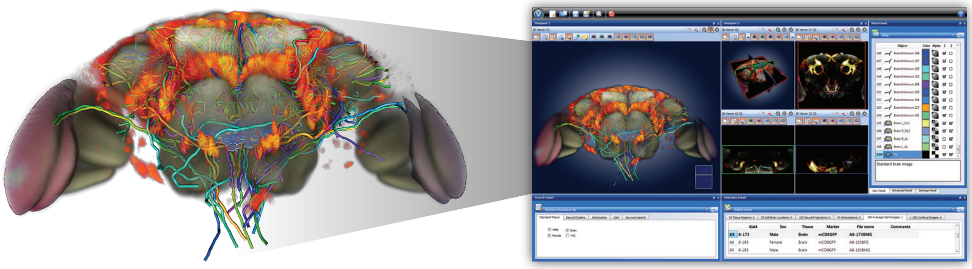


**Figure 5.1 –** Neural projections in the brain of the fruit fly visualized using the *BrainGazer* system.

> Science is the belief in the ignorance of the experts.
>
> — *Richard Feynman*

# CHAPTER 5

# BrainGazer – Visual Queries for Neurobiology Research

Neurobiology investigates how anatomical and physiological relationships in the nervous system mediate behavior. Molecular genetic techniques, applied to species such as the common fruit fly *Drosophila melanogaster*, have proven to be an important tool in this research. Large databases of transgenic specimens are being built and need to be analyzed to establish models of neural information processing. In this paper we present an approach for the exploration and analysis of neural circuits based on such a database. We have designed and implemented *BrainGazer*, a system which integrates visualization techniques for volume data acquired through confocal microscopy as well as annotated anatomical structures with an intuitive approach for accessing the available information. We focus on the ability to visually query the data based on semantic as well as spatial relationships. Additionally, we present visualization techniques for the concurrent depiction of neurobiological volume data and geometric objects which aim to reduce visual clutter. The described system is the result of an ongoing interdisciplinary collaboration between neurobiologists and visualization researchers.

## 5.1 Introduction

A major goal in neuroscience is to define the cellular architecture of the brain. Mapping out the fine anatomy of complex neuronal circuits is an essential first step in investigating the neural mechanisms of information processing. This problem is particularly tractable in insects, in which brain structure and function can be studied at the level of single identifiable neurons. Moreover, in the fruit fly *Drosophila melanogaster*, a rich repertoire of molecular genetic tools is available with which the distinct neuronal types can be defined, labeled, and manipulated [122]. Because of the high degree of stereotypy in insect nervous systems, these genetic tools make it feasible to construct digital brain atlases with cellular resolution [112]. Such atlases are an invaluable reference in efforts to compile a comprehensive set of anatomical and functional data, and in formulating hypotheses on the operation of specific neuronal circuits.

One approach in generating a digital atlas of this kind is by acquiring confocal microscope images of a large number of individual brains. In each specimen, one or more distinct neuronal types are highlighted using appropriate molecular genetic techniques. Additionally, a general staining is applied to reveal the overall structure of the brain, providing a reference for non-rigid registration to a standard template. After registration, the specific neuronal types in each specimen are segmented, annotated, and compiled into a database linked to the physical structure of the brain. The complexity and sheer amount of these data necessitate effective visualization and interaction techniques embedded in an extensible framework. We detail *BrainGazer*, a novel visualization system for the study of neural circuits that has resulted from an interdisciplinary collaboration. In particular, in addition to visualization, we focus on intuitively querying the underlying database based on semantic as well as spatial criteria.

The remainder of the paper is structured as follows: Related work is discussed in Section 5.2. Section 5.3 outlines the data acquisition workflow and gives a conceptual overview of our system. In Section 5.4 we detail the visualization techniques employed. Our novel approach for semantic and spatial visual queries is presented in Section 5.5. Section 5.6 provides details on the implementation of our system. Results are discussed in Section 5.7. The paper is concluded in Section 5.8.

## 5.2 Related Work

Excellent starting points to get insight into the world of neuroscientists, their data, the huge data collections, and related problems are given by Koslow and Subramaniam [90] as well as Chicurel [30]. Data acquired to study brain structure captures information on the brain on different scales (e.g., molecular, cellular, circuitry, system, behavior), with different focus (e.g., anatomy, metabolism, function) and is multi-modal (text, graphics, 2D and 3D images, audio, video). The establishment of spatial relationships between initially unrelated images and information is a fundamental step towards the exploitation of available data [17]. These relationships provide the basis for the visual representation of a data collection and the generation of further knowledge. Jenett et al. [75] describe techniques and workflow for quantitative assessment, comparison, and presentation of 3D confocal microscopy images of Drosophila brains and gene expression patterns within these brains. An automatic method to analyze and visualize large collections of 3D microscopy images has been proposed by de Leeuw et al. [37].

Brain atlases are a common way to spatially organize neuroanatomical data. The atlas serves as reference frame for comparing and integrating image data from different biological experiments. Maye et al. [112] give an introduction and survey on the integration and visualization of neural structures in brain atlases. A classical image-based neuroanatomical atlas of *Drosophila melanogaster* is the FlyBrain atlas[1], spatially relating a collection of 2D drawings, microscopic 2D images and text. The web interface provides visual navigation through the data by clicking on labeled structures in images. Brain Explorer [93], an interface to the Allen Brain Atlas, allows the visualization of mouse brain gene expression data in 3D. An example for a 3D atlas of the developing Drosophila brain has been described by Pereanu and Hartenstein [124]. Segmentation, geometric reconstruction, annotation, and rendering of the neural structures was performed using Amira[2]. The Neuroterrain 3D mouse brain atlas [14] consists of segmented 3D structures represented as geometry and references a large collection of normalized 3D confocal images. An interface to interact with the data has been described for neither of these atlases. NeuARt II [24] provides a general 2D visual interface to 3D neuroanatomical atlases including interactive visual browsing by stereotactic coordinate navigation. The CoCoMac-3D Viewer developed by Bezgin et al. [15] implements a visual interface to two databases containing morphology and connectivity data of the macaque brain for analysis and quantification of connectivity data. It also allows graphical manipulation of entire structures.

Most existing interfaces to neuroanatomical databases provide only very limited tools for visual analysis, although there exist powerful general methods for the exploration of multidimensional and spatial data. Surveys of concepts for visual analysis of databases and visual data mining have been published by Derthick et al. [38] and Keim [80]. The most prominent techniques are interactive filtering by dynamic queries [1] and brushing and linking for the exploration of multidimensional data [109]. Special focus on visual analytics of spatial databases discussing multidimensional access methods is subject of the survey by Gaede and Günther [55]. Examples for visual navigation through spatial data can be mainly found in geographical information systems (e.g., Google maps or the public health surveillance system proposed by Maciejewski et al. [105]).

An example for interfaces to neuroanatomical image collections and databases realizing more elaborate visual query functionalities is the European Computerized Human Brain Database (ECHBD) [52]. It connects a conventional database with an infrastructure for direct queries on raster data. Visual queries on image contents can be directly realized by interactive definition of a volume of interest in a 3D reference image. Direct search by drawing regions of interest in a 2D image to query injection and label sites on a set of related studies has been realized by Press et al. [127] as interface to the XANAT database. Ontology-based high-level queries in a database of bee brain images based on pre-generated 3D representations of atlas information have been recently

---

[1] `http://flybrain.neurobio.arizona.edu`

[2] `http://www.amira.com`

proposed by Kuß et al. [91]. The interactive definition of volumes of interest directly on the 3D data for queries on pre-computed fiber tracts of a Diffusion Tensor Imaging (DTI) data set has been proposed by Sherbondy et al. [145]. Several approaches for the 3D visualization of neurons based on microscopy data have been presented [8, 78, 111, 136]. Rendering of pure geometric representations of large neural networks has been addressed recently by de Heras Ciechomski et al. [36].

Nevertheless, the visual presentation of 3D neuroanatomical image data in query interfaces to large data collections is currently mainly realized as geometric representations of atlas information in combination with or alternatively to axis aligned 2D sections of the image data. The system presented in this paper combines state-of-the-art 3D visualization techniques for neurobiological data with a novel visual query interface, thereby integrating semantic and spatial information.

## 5.3 System Overview

The nervous system is composed of individual neurons, which process and transmit information in the form of electrochemical signals. They are the basic structural and functional units of an organism's nervous system and are therefore of primary interest when studying brain function. Different types of specialized neurons exist and knowledge about their arrangement, connectivity, and physiology allows neuroscientists to derive models of cognitive processes. In an interdisciplinary collaboration between neurobiologists and visualization researchers, we investigate neural circuits in the fruit fly *Drosophila melanogaster*. Conserved genes and pathways between flies and other organisms, together with the availability of sophisticated molecular genetic tools make Drosophila a widely used model system for elucidating the mechanisms that affect complex traits such as behavior. This section gives an overview on the basic methodology we use for these studies and the visualization system which has been developed.

### 5.3.1 Data Acquisition

We use the Gal4/UAS system [18] to label and manipulate specific neurons in the fly brain and ventral nerve cord. The brain and nerve cord are separately dissected. Specific neurons are stained with a green fluorescent protein (GFP). Additionally, separate neuropil staining is used to facilitate registration – it highlights regions of high synaptic density which provide a stable morphological reference. After preparation and staining, the tissues are scanned using a Zeiss LSM 510 laser scanning confocal microscope with a 25X objective. Data sets of 165 slices at a 1 μm interval and an image resolution of 768 × 768 pixels are generated.

The neuropil staining is then used to perform non-rigid registration [135] of the scans to a corresponding template for either brain or ventral nerve cord, similar to the approach described by Jenett et al. [75]. The template was generated by averaging a representative set of scans registered against a reference scan. The registration process itself is automatic, but results are manually verified and additional image processing operations may be applied to reduce noise. Only scans considered to be registered with sufficient accuracy are used for the database.

Each neuron is characterized by three types of features: a *cell body*, the *neural projection*, which is an elongated structure that spreads over large areas, and *arborizations*, which contain synapses where communication with other neurons occurs. Neurons are classified based on the morphology or shape of these features. Neurons that share similar cell bodies, patterns of projections, and arborizations, as well as expression of the same Gal4 drivers, are tentatively considered to belong to the same type. Types of neurons having these anatomical properties may perform similar functions.

Standardized volumes are created by generating averages for each Gal4 line which allow evaluation of the biological variability of the corresponding expression patterns. Amira is used to segment cell body locations and arborizations from these average volumes. The resulting objects are therefore representations of the typical locations and shapes of these structures. They are examined together with the corresponding average volumes and individual confocal scans in order to assess their constancy between multiple specimens. Neural projections are traced from individual images using the skeletonizer plugin for Amira [140]. Surface geometry is generated for cell body locations and arborizations, while neural projections are stored as skeleton graphs. References to these files, the original confocal volumes, the average Gal4 volumes, the templates, and template regions (surface geometry based on a template volume representing particular parts of the anatomy such as the antennal lobes) are stored in a relational database. The central entities within this database are *neural clusters* which group cell body locations, neural projections, and arborizations. These neural clusters correspond to particular neuronal types.

### 5.3.2 Visualization and Interaction

One of the goals in developing the *BrainGazer* system was to facilitate the study of neural mechanisms in the mating behavior of Drosophila using the acquired data. The research challenge is to reveal how chemical and auditory cues are detected and processed in the fly's brain, how these signals are interpreted in the context of internal physiological states and past experience, and how this information is used to make decisions that are fundamental to the animal's reproductive success [39]. By visualizing individual neurons on a common reference template, potential connections between these neurons

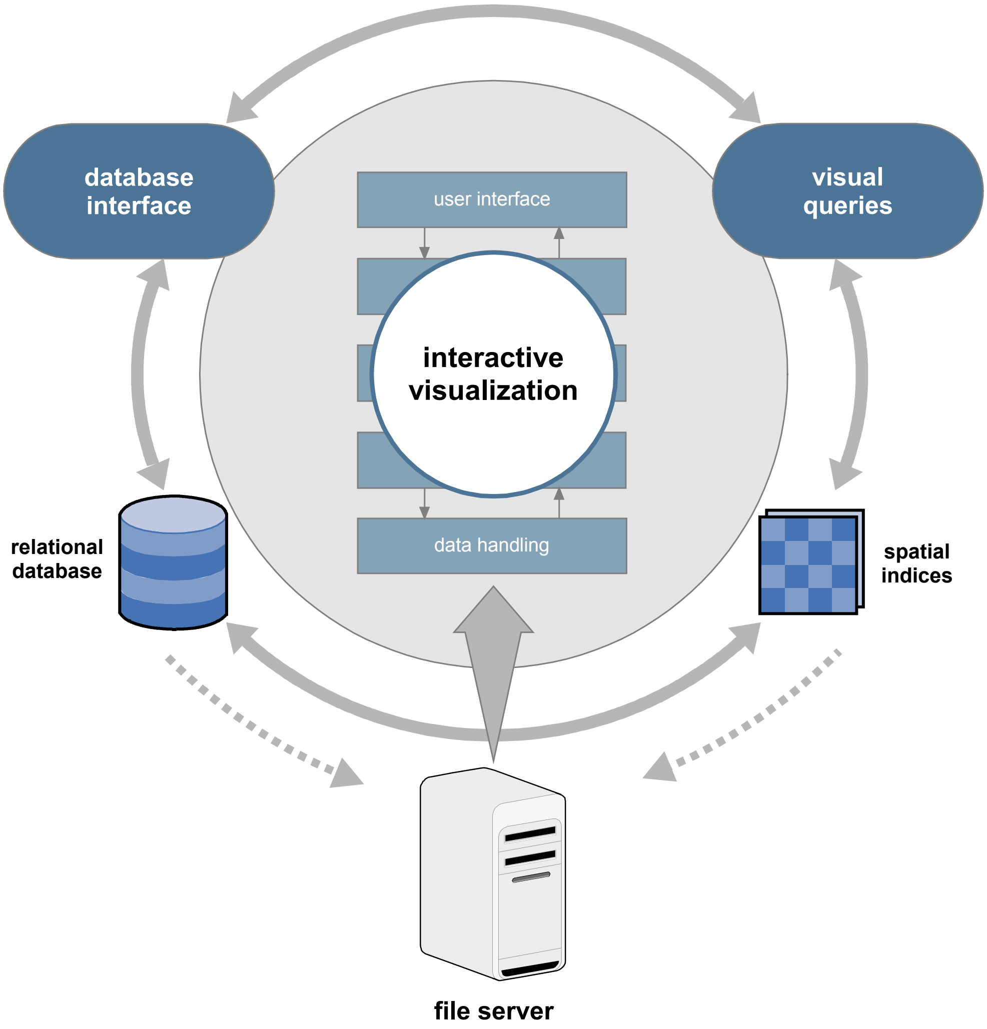


**Figure 5.2 –** Conceptual overview of the *BrainGazer* system.

based on the spatial colocalisation of their arbor densities can be identified. This information is used to generate network diagrams which allow us to formulate specific hypotheses of circuit function. For example, we have used this principle to identify the neuronal types that constitute a putative pathway for sensing and processing pheromone signals and triggering courtship behavior. In order to facilitate this type of examination, it is important to provide efficient means for interactively accessing the generated database. *BrainGazer* provides two distinct paths to select data for display and analysis (see Figure 5.2):

**Database interface.** A traditional table-view database interface allows users to filter and select items based on combinations of different criteria, such as gender or neuronal type. A result view is updated immediately when query parameters change. The user can then select the desired data items and load them into the application. Additionally, it is possible to perform a full-text search of the database to quickly access specific data sets.

**Visual queries.** While a traditional database interface is useful for quickly accessing a known subset of the data, it is also important to be able to visually search the whole set of available data based on spatial relationships. The visual query interface is displayed directly in the visualization window and provides instant access to contextual information and related structures for selected items and regions of interest.

All data sets are stored on a central file server and transferred on-demand. The relational database storing references to these files is also accessed over the network. To facilitate visual queries, a set of spatial indices is maintained and updated whenever changes to the data occur.

The application itself comprises a rich set of standard tools for 2D/3D navigation (rotation, zoom, pan, slicing), rendering (orthographic and perspective projection, clipping planes, cropping boxes, transfer functions, windowing), multiple linking of 3D and 2D views, multi-screen support, and image and video capture. Working sessions together and all current settings can be saved to disk and later restored with automatic transfer of all loaded data sets. As these features are common in similar systems, we restrict our further discussion to novel aspects of our approach. A typical screenshot of *BrainGazer* is shown in Figure 5.1.

## 5.4 Visualization

One of the challenges in developing *BrainGazer* was the concurrent visualization of many different anatomical structures while minimizing visual clutter. As semi-transparent volume data is depicted together with geometric objects, care has to be taken to avoid occlusion while preserving the ability to identify spatial relationships. In this section, we describe the visualization techniques we employ for this purpose.

Template regions, cell body locations, and arborizations are available as triangle meshes. We render them using standard per-pixel Phong illumination. Neural projections are given as skeleton graphs with optional diameter information. As the diameter values can be unreliable and misleading, the projections are preferably viewed with a constant diameter. However, we provide the option of using the available diameter values as well. The skeleton graph is extruded to cylinders and rendered as polygonal geometry which also enables simple and fast rendering of object outlines in 2D slice views (in the future we also plan to investigate more advanced techniques to improve the visual quality, such as convolution surfaces [121] or self-orienting surfaces [113]). The remaining data sets, templates, average Gal4 volumes, and confocal scans are volume data. The users of *BrainGazer* want to visualize them together with the geometric objects.

### 5.4.1 Volume Rendering

Volume data acquired by confocal microscopy is frequently visualized using Maximum Intensity Projection (MIP) as the stained tissues have the highest data values. When concurrently depicting several different scans, however, the disadvantage of MIP is that spatial relationships are lost. Using Direct Volume Rendering (DVR), on the other hand, suffers from occlusion. This is particularly problematic as it is frequently necessary to visualize several confocal scans together with a template volume which provides anatomical context. The template should not occlude features highlighted in the other data sets but is important for spatial orientation. Thus, in *BrainGazer* we chose to employ a variant of Maximum Intensity Difference Accumulation (MIDA) [19]. As MIDA represents a unifying extension of both DVR and MIP, it is well suited for our problem. We extended the method to enable the concurrent rendering of multi-channel data sets [25].

MIDA uses a generalization of the over operator where the previously accumulated color and opacity are modulated by an additional factor. The accumulated opacity $A_i$ and color $C_i$ at the $i$-th sample position $P_i$ along a viewing ray traversed in front-to-back order are computed as:

$$\begin{aligned} A_i &= \hat{B}_i A_{i-1} + (1 - \hat{B}_i A_{i-1})\hat{A}_i \\ C_i &= \hat{B}_i C_{i-1} + (1 - \hat{B}_i A_{i-1})\hat{A}_i \hat{C}_i \end{aligned} \tag{5.1}$$

where $\hat{A}_i$ and $\hat{C}_i$ are the opacity and color, respectively, of the sample and $\hat{B}_i$ is the modulation factor. In the original method, which focused on single-channel data sets, $\hat{B}_i$ was defined based on the absolute difference between the current maximum along the ray and the data value at a sample point. This approach gives increased visual prominence to local maxima. The resulting images share many of the characteristics with MIP, but feature additional spatial cues due to accumulation.

In the following, we present a simple extension of MIDA to multi-channel data. We assume a multi-channel data set consisting of $N$ continuous scalar-valued volumetric functions $f_1(P), \ldots, f_N(P)$ of normalized data values in the range $[0, 1]$. Each channel has an associated color function $c_1(P), \ldots, c_N(P)$ and opacity function $\alpha_1(P), \ldots, \alpha_N(P)$.

Like Kniss et al. [88], at the $i$-th sample position $P_i$ along a ray, we sum the opacities and average the colors for the overall opacity $\hat{A}_i$ and color $\hat{C}_i$ of the sample (opacities larger than one are subsequently clamped):

$$\hat{A}_i = \sum_{j=1}^{N} \alpha_j(P_i) \quad \hat{C}_i = \frac{\sum_{j=1}^{N} \alpha_j(P_i) c_j(P_i)}{\sum_{j=1}^{N} \alpha_j(P_i)} \tag{5.2}$$

We want to enhance regions where the maximum along the ray changes for any channel. Specifically, when the maximum changes from a low to a high value, the corresponding sample should have more influence on the final image compared to the case where the difference is only small. We use $\delta_j$ to classify this change at every sample location $P_i$:

$$\delta_j(P_i) = \begin{cases} f_j(P_i) - \max_{k=1}^{i-1} f_j(P_k) & \text{if } f_j(P_i) > \max_{k=1}^{i-1} f_j(P_k) \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Whenever a new maximum for channel $j$ is encountered while traversing the ray, $\delta_j$ is nonzero. These are the cases where we want to override occlusion relationships. For this purpose, the modulation factor $\hat{B}_i$ from Equation 5.1 is defined as:

$$\hat{B}_i = 1 - \max_{j=1}^{N} \delta_j(P_i) \frac{\alpha_j(P_i)}{\alpha_{\max}(P_i)} \tag{5.4}$$

In Equation 5.4 the maximum of $\delta_j(P_i)$ weighted by the ratio between each channel's opacity $\alpha_j(P_i)$ and the maximum opacity $\alpha_{\max}(P_i)$ of all channels is computed. The additional weighting ensures that invisible samples have no influence on the final image. If the maximum opacity is zero, i.e., no channel is visible at the current sample location, we set $\hat{B}_i$ to one.

The advantage of this approach is that it enables a clear depiction of stained data sets but does not require complex transfer functions to resolve occlusion problems. In our system, transfer function specification is typically performed by defining a linear opacity mapping using standard window/level controls and choosing a pre-defined color map.

Using this volume rendering technique, a high-intensity stained structure immersed in the relatively homogeneous template, for example, will be distinctly visible while still featuring subtle transparency as an additional occlusion cue. Channels with no distinct maxima will appear DVR-like while stained data will exhibit visual characteristics very similar to MIP. In contrast to two-level volume rendering [64] or the approach of Straka et al. [148], no pre-classification of structures of interest is required.

Figure 5.3 shows the template volume of the ventral nerve cord (in gray tones) together with two stained average Gal4 volumes (depicted in shades of red and blue) rendered using (a) DVR, (b) MIDA, and (c) MIP. The same color and opacity transfer functions are used and all three techniques combine the individual channels using Equation 5.2. While considerable parts of the stained tissue are occluded in DVR, MIDA and MIP clearly depict the stained

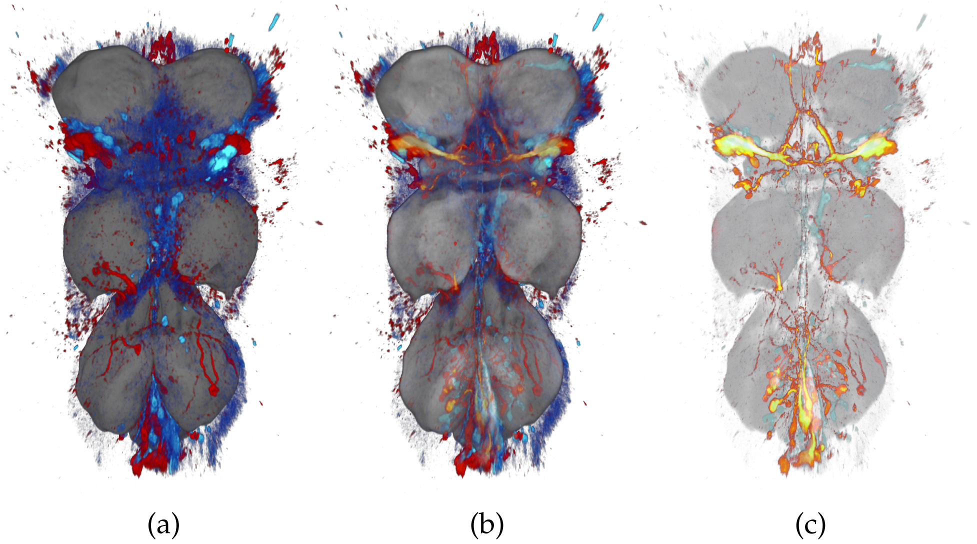

(a) (b) (c)

**Figure 5.3 –** The ventral nerve cord of a fly rendered using (a) DVR, (b) MIDA, and (c) MIP. The template (gray) is depicted together with two average Gal4 data sets (red and blue).

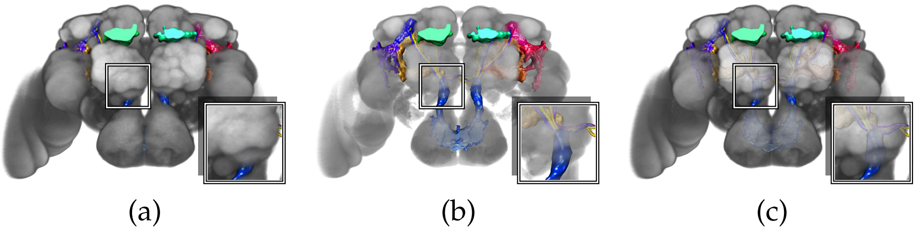

(a) (b) (c)

**Figure 5.4 –** Neural clusters in the fly brain depicted together with the template volume using (a) no see-through enhancement, (b) no see-through enhancement with an adjusted transfer function, and (c) see-through mode using the same transfer function as in the leftmost image.

neurons. MIDA, however, provides more anatomical context and spatial cues. We allow the user to smoothly transition between these three methods [19].

### 5.4.2 Geometry Enhancement

In the targeted application, geometric objects corresponding to segmented anatomical structures are displayed immersed in volumetric data. While the volume data is important as it provides the spatial context, it is undesirable that it fully occludes the geometry. Opacity could be adjusted to prevent occlusion, but it is cumbersome to tune transfer functions individually. Inspired by the MIDA approach to volume rendering, we employ a similar concept to enable the user to see-through the volume even if it would completely occlude intersecting objects. Based on the technique presented by Luft et al. [99], we apply an unsharp masking operation to the depth buffer established during rendering of the geometry. Their spatial importance function $\Delta D$ is defined as the difference between the low-pass filtered version of the depth buffer and the original depth buffer. This simple approach gives information about spatially important edges, e.g., areas containing large depth differences.

In our approach, we use $\Delta D$ to modulate the accumulated opacity and color along a viewing ray based on the absolute value of $\Delta D$. The result is then blended with the geometry's color contribution. Additionally, as proposed by Luft et al. [99], we can apply depth-enhancement by darkening and brightening the geometry color based on $\Delta D$ with no additional cost. The effect of this simple approach is that regions which feature depth discontinuities shine through the volume rendering most. Thus, while giving the user the ability to identify objects immersed in the volume, this methods still indicates occlusion relationships.

An example is shown in Figure 5.4 – the template brain tissue is depicted together with several neural clusters which are mostly occluded in Figure 5.4 (a) where no geometry enhancement is applied. In Figure 5.4 (b) the transfer function was adjusted to make the geometry more visible. Figure 5.4 (c) clearly depicts the geometry as well as the volume data while still indicating occlusion relationships using our see-through approach.

## 5.5 Visual Queries

While a traditional database browsing approach is useful for analyzing specific known structures, neurobiological research frequently requires access to the data based on spatial relationships. For example, the biologist may wish to identify neurons or other structures in the vicinity, in order to classify specific objects and to begin to reconstruct neural circuits. A specific case arises as new data is added to the database: the biologist wants to compare it to existing structures in order to decide whether it belongs to a known neuronal type. As there may be substantial variations in individual shapes, it is necessary to investigate all nearby objects to achieve a classification.

*BrainGazer* provides three basic types of visual queries: Semantic queries give access to related structures using information stored in the database. Object queries are based on the distance between whole objects. Path queries are the most flexible method. Through an intuitive freehand drawing interface, the user can search for proximal structures. These types of queries can be arbitrarily combined. Object and path queries can be used to amend or verify recorded semantic information stored in the database. The user can interact with these different query types through contextual hypertext labels which are displayed in-window.

### 5.5.1 Semantic Queries

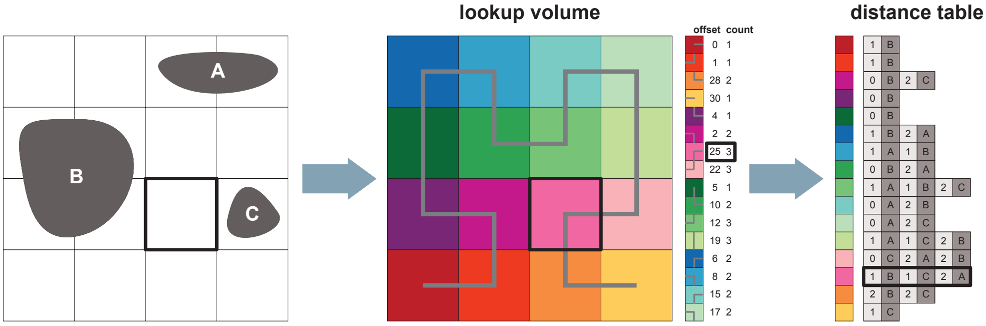
Semantic queries allow the user to quickly access contextual information and data for an object of interest. They are initiated by simply selecting an object in the visualization window through a mouse click. If multiple objects overlap in depth, subsequent clicks at the same position allow cycling through them. As soon as a new object has been picked, a contextual hypertext label appears on-screen and provides the information stored in the database such as the name of the structure and comments. References to other related objects are displayed as hyperlinks which can be used to access the associated structure. This includes geometric objects, e.g., other cell body locations, neural projections, or arborizations of the same neuronal type, as well as volumetric data such as the scan the object has been segmented from. This setup can be used to navigate through the data. For instance, an arborization may be part of one or several neuronal types. When selecting the arborization, the label shows all neuronal types linked to the arborization together with the cell body locations, neural projections, and other arborizations as hyperlinks. Selecting another arborization in one of these neuronal types will provide access to further structures considered connected to this arborization. Hovering over a hyperlink also highlights the corresponding objects if they already have been loaded. If the objects are not already visible, activating the link by clicking it will initiate a load operation. This simple approach allows the user to quickly navigate known neural circuits using a familiar interface.

### 5.5.2 Object Queries

In addition to providing access to semantic information already present in the database, our system allows users quick access to spatial proximity information in order to aid identification of new relationships.

For this purpose, we create a table which stores the minimum distances of an object to all other objects in a pre-processing step. We use signed distance volumes generated for all objects in the database. The minimal surface distance between two objects $i$ and $j$ is computed by sampling the distance volume of $j$ for every voxel along the surface of $i$ and vice versa. If the minimum is negative, we continue to compute the volume of intersection between the two objects and record it as a negative value. Table entries for each object are then sorted according to ascending distance values. This table is loaded into memory at startup and allows quick access to proximity information whenever an object is selected.

The result of an object query is displayed in conjunction with the semantic information in a contextual label when a picking operation occurs. The label will display hyperlinks for all objects within a certain range from the selected structure. A slider widget integrated with the label allows interactive filtering of the query results based on distance. When moving the slider, the label



| offset | count |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 28 | 2 |
| 30 | 1 |
| 4 | 1 |
| 2 | 2 |
| 25 | 3 |
| 22 | 3 |
| 5 | 1 |
| 10 | 2 |
| 12 | 3 |
| 19 | 3 |
| 6 | 2 |
| 8 | 2 |
| 15 | 2 |
| 17 | 2 |

| | | | | | |
|---|---|---|---|---|---|
| 1 | B | | | | |
| 1 | B | | | | |
| 0 | B | 2 | C | | |
| 0 | B | | | | |
| 0 | B | | | | |
| 1 | B | 2 | A | | |
| 1 | A | 1 | B | | |
| 0 | B | 2 | A | | |
| 1 | A | 1 | B | 2 | C |
| 0 | A | 2 | B | | |
| 0 | A | 2 | C | | |
| 1 | A | 1 | C | 2 | B |
| 0 | C | 2 | A | 2 | B |
| 1 | B | 1 | C | 2 | A |
| 2 | B | 2 | C | | |
| 1 | C | | | | |

**Figure 5.5 –** Lookup volume and distance table for a simple two-dimensional scene containing three objects – the city block distance metric is used and distances above 2 are ignored. The Hilbert curve used to arrange the distance table is overlayed in light gray. An example lookup is indicated with black outlines.

immediately updates. For each object type, the number of objects within the specified distance range is displayed followed by a list of their names (as retrieved from the database). Each name represents a hyperlink which can be used to load and highlight the object. These links are additionally color-coded to quickly identify the object's degree of spatial proximity.

### 5.5.3 Path Queries

Path queries are based on an intuitive freehand drawing interface: the user sketches an arbitrary path on top of the visualization and gets immediate feedback about nearby objects. The result of the query can then be loaded into view for further inspection. Sketches were chosen over more conventual selection tools such as rectangular or circular regions as they allow a more accurate characterization of the region of interest in the context of complex neural anatomy.

#### Index Generation

To facilitate fully interactive visual queries, we generate a spatial index which allows us to quickly retrieve the objects in the vicinity of a specific location. In a pre-processing step we create a lookup volume and a distance table. At runtime, the lookup volume is kept in memory while the distance table may be accessed out-of-core. The distance table grows with the number of objects in the database while the size of the lookup volume remains constant. This is important for scalability as significant growth in the number of annotated objects is expected. For each voxel, the lookup volume stores an offset into the distance table and the number $N$ of proximal objects found for the voxel position $P$. Each entry in the distance table corresponds to such a position in the volume and contains a list of $N$ <distance, identifier> pairs. The pairs in the list are sorted according to their ascending distance from $P$. Negative distances indicate that the point $P$ is located inside of the respective object. During pre-processing, the distances are determined using signed distance fields stored for each object. As we are not interested in objects located far from a queried point, all distances above a certain threshold are ignored and not stored. In practice, a maximum distance of 40 voxels has proven to be useful and is used in our current implementation.

Using these data structures, during interaction objects close to any voxel can be found by simply reading the offset and count from the corresponding location in the lookup volume and then retrieving the respective set of <distance, identifier> pairs from the distance table. In order to enable efficient caching for subsequent accesses to the distance table, it is advantageous to choose a locality-preserving storage scheme. Many out-of-core approaches employ space-filling curves for this purpose. In our current implementation, the entries of the distance table are arranged based on the three-dimensional Hilbert curve [56] which has been shown to have good locality-preserving properties [74]. Figure 5.5 illustrates lookup volume and distance table for the two-dimensional case.

Our concept also allows easy merging of distance tables and lookup volumes for disjoint sets of objects which is practically needed when new objects are inserted into the database. We merge the entries of the distance tables with a union operation and resort them. The offsets and counts in the lookup volumes can simply be added.

### Query Processing

Using the described data structures, we can efficiently determine objects in the vicinity of a voxel. For performing path queries, it is therefore necessary to identify a corresponding 3D object-space position for each 2D point along the path. Whenever a new point is added to the path, we read the depth buffer at the corresponding 2D location and use the inverse viewing transformation to transform it into object space. For geometry rendering, this results in the position on the surface of the object closest to the viewer. For volume rendering, however, several samples along a ray may contribute to a pixel. For each viewing ray, we therefore choose to write the depth of the sample which contributes most to the final pixel color. Particularly in conjunction with the volume rendering technique described in Section 5.4.1 this approach has proven useful – as stained tissues are presented visually more prominent, the respective depth will give access to objects in the vicinity of high-intensity volumetric structures. Using the depth buffer in this way also ensures that there is always a good correspondence between the selected query locations and the actual visualization when operations such as cropping have been applied or a slicing plane is displayed in the 3D visualization.

As path queries are used to find structures which are not visible, all currently displayed objects are ignored. During the query, we maintain a sorted list of <distance, identifier> pairs for all objects encountered along the path. When a new point is added, we retrieve its <distance, identifier> pairs from the distance table and merge them into this list. If an object has already been encountered along the path, the lower of the two distances is stored. Similar lists are kept separately for each object type. This information is then used to present the query results to the user.

### User Interaction

A path query can be initiated by the user by simply clicking on any position in the window and painting the desired path while keeping the mouse button pressed. A hypertext label pops up on the side of the window and is constantly updated with current information on the number and type of objects found. As soon as the user releases the mouse button, the contextual label moves to the center of the screen prompting the user to inspect the results. Activating a link by clicking it loads and highlights the corresponding objects. Query results can be discarded by right-clicking the label. The query results are displayed in the same way as for object queries using an integrated slider widget for interactive filtering.

During the query, the specified path is overlayed with proximity clouds which provide an instant visual indication of close objects without having to load the geometry first. For this purpose, for every point of the path we draw a circle for each detected nearby object into an offscreen buffer. The radius of each circle corresponds to the recorded distance and each pixel inside the circle is set to this distance value – entries for objects which intersect the point are drawn using a default radius. Pixel values are combined using minimum blending. The result is a buffer which stores the closest distance to any object at each pixel. These values are then mapped to colors and opacities and displayed semi-transparently as shown in Figure 5.6 (b). After the query, when the user hovers over any of the hyperlinks the corresponding proximity overlay is shown.

## 5.6 Implementation

The presented system was implemented in C++ using OpenGL and GLSL. For the user interface, the Qt toolkit was used. The architecture is based on a flexible plug-in mechanism which allows independent modification or even replacement of system components. This modular concept has proven to be very useful as it allows rapid prototyping of new functionality. For instance, the traditional database browsing components were implemented and deployed first. The visual query module uses the same interface which greatly simplified integration and testing. As the system was developed within the scope of an ongoing project, we expect to add new features such as integration with additional databases using the same procedure. The application is designed to run on commodity PCs equipped with Shader Model 3.0 capable graphics hardware. The system is used on a number of different computers ranging from laptops to high-end visualization workstations.

## 5.7 Results and Discussion

Currently, the database contains several thousand individual confocal scans, several hundred average Gal4 volumes, as well as hundreds of geometric objects (cell body locations, neural projections, and arborizations) with new data items being added on a regular basis. The lookup volume is generated at a resolution of $384 \times 384 \times 82$ voxels. For each template (either brain or ventral nerve cord) the current distance table for a cutoff distance of 40 voxels requires approximately 200 MB of storage. The time required for adding a new object is approximately 5 minutes including distance field generation, computation of distance and object tables, and subsequent merging of these tables. These operations are performed in an offline batch process.

In order to gain estimates about the scalability of our approach, we performed experiments with higher cutoff distances which result in a larger number of entries per point. For a maximum distance of 256 voxels, the distance table requires 2 GB of storage thus approximately simulating a growth of one order of magnitude in the number of segmented structures. For the small distance table, the average times for lookup and retrieval of all entries for a single location from the hard disk are below 1 ms even without the use of an explicit caching mechanism. In the case of the large table, the time increases to approximately 3 ms indicating that the approach is prepared to handle a substantial increase in the number of objects.

A typical use-case of our visual query approach is illustrated in Figure 5.6. Since it is difficult to depict an interactive process using still images and because our user interface is designed for on-screen viewing, we refer to the accompanying video for a sample of an interaction session. Initially, in Figure 5.6 (a), the brain template is shown together with an average Gal4 volume which has been selected using the database browser. A path query is then specified in Figure 5.6 (b). The best match of the query – an arborization – is loaded and selected. In Figure 5.6 (c), the contextual label gives access to semantic query results: the arborization's neural cluster which contains one cell body location, two neural projections, and two further arborizations, is loaded. In Figure 5.6 (d), the object query information is used to load an additional intersecting neural projection. Finally, in Figure 5.6 (e) the neural projection's associated cluster is loaded using another semantic query resulting in Figure 5.6 (f). This simple example demonstrates how our approach –

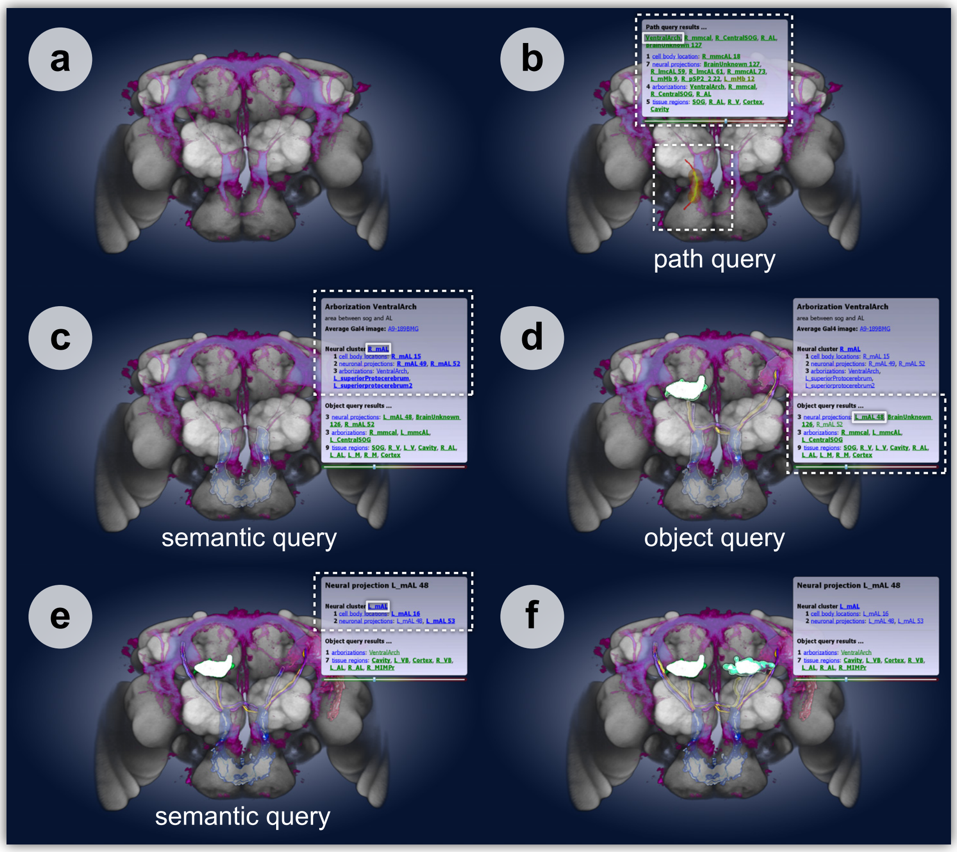


**Figure 5.6 –** A simple interaction session using visual queries. (a) Initial state. (b) Path query and selection of an arborization. (c) Selection of the arborization's neuronal type. (d) Object query for nearby neural projection. (e) Selection of the projection's neuronal type. (d) Final state.

using a combination of semantic queries, object queries, and path queries – allows intuitive navigation through complex data.

As *BrainGazer* was developed in an interdisciplinary effort together with domain experts, the described techniques benefitted from constant input by neurobiologists. The possibility of being able to quickly access semantically related or proximal objects was an important goal. We received very encouraging feedback on how the availability of such a system will ease future research. In particular, the concept of presenting query results directly in the visualization window using hypertext labels – as opposed to displaying them in a separate user-interface widget – was appreciated, as it allows the user to remain focused on the visualization. After initial demonstrations of our visual query approach several changes were made based on user comments. For instance, we integrated the distance slider with the contextual label to allow interactive filtering with immediate feedback. We also color-coded the object names in the query results to give a better indication of an object's placement within the query range. Another request was to leave the contextual label visible until explicitly discarded – initially, the label disappeared as soon as an object had been selected for loading. The new behavior allows users to inspect all likely matches sequentially before coming to a conclusion. As a neuronal type, for example, may contain several neural projections which – with slight variation – follow the same path, it is important to view them all when judging connectedness.

We are currently using the techniques presented in this paper to assemble a cellular atlas of the network of neurons that express the *fruitless* gene (*fru*), which have been functionally linked to male courtship behavior [39]. The *BrainGazer* system has proven invaluable in the digital reconstruction of this network, which now comprises over 90 distinct neuronal types. We are now working towards expanding this database to encompass an even broader range of neurons, while also further developing the database and visualization software. Our aim is to release these tools to the neuroscience research community in the near future, in the expectation that they will similarly facilitate the anatomical exploration of other neuronal circuits in the fly. Although this system has been developed for analysis of the Drosophila nervous system, the computational methods are equally applicable to any species that exhibits a high degree of stereotypy in the cellular architecture of its nervous system, including most other prominent model organisms in neurobiology research.

## 5.8 Conclusion

In this paper we presented a system for the interactive visualization, exploration, and analysis of neural circuits based on a neurobiological atlas. We discussed visualization techniques for the effective depiction of multi-channel confocal microscopy volume data in conjunction with segmented anatomical structures. An intuitive visual query approach for navigating through the available data based on semantic as well as spatial relationships was presented. The system was designed and implemented in collaboration with domain experts and is currently in use to assist their research.

In the future we want to extend the scope of this project. Our goal is to build a complete online atlas of neural anatomy. We plan to further develop the *BrainGazer* system so that it fully integrates with this atlas and make it freely available to researchers in the field. In particular, we aim to enable the interactive definition, modification, and annotation of semantic relationships between anatomical structures in the atlas based on visual queries. We envision such a system to facilitate large scale collaborative research in neuroscience.

Furthermore, we hope that making the system available to a larger user base will enable us to study and improve the effectiveness of the presented visualization and interaction techniques. One viable strategy could be the automatic gathering of anonymized usage logs together with evaluation forms directly in the application. Information derived from this data could be employed to optimize the workflow and to identify areas of future research.

**The following chapter was originally published as:**

S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.

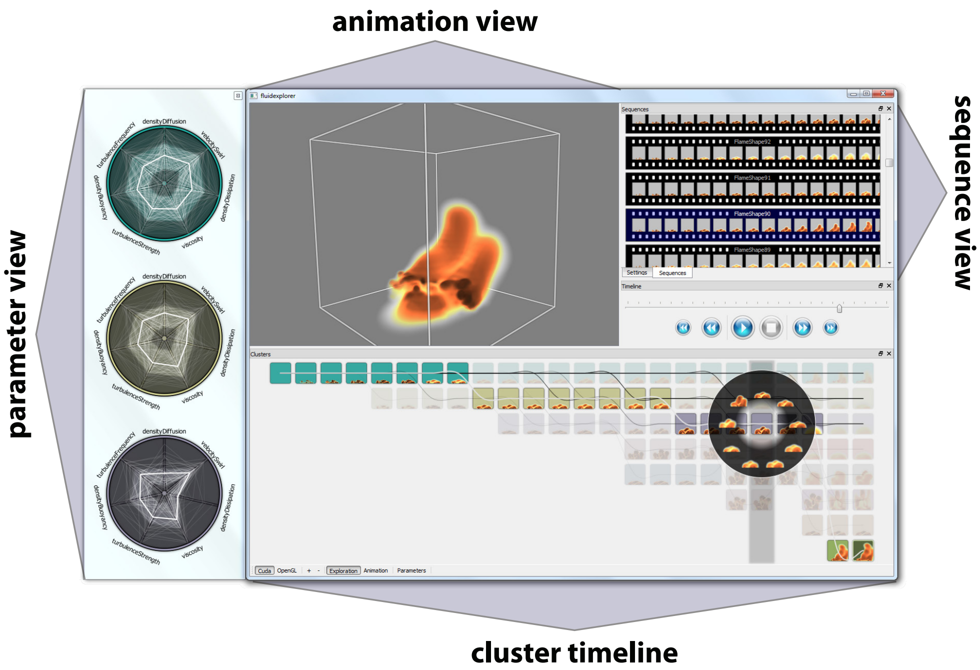


Figure 6.1 – Screenshot of our interactive exploration environment.

There's no system foolproof enough
to defeat a sufficiently great fool.
— *Edward Teller*

CHAPTER

# 6

# Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design

Graphics artists commonly employ physically-based simulation for the generation of effects such as smoke, explosions, and similar phenomena. The task of finding the correct parameters for a desired result, however, is difficult and time-consuming as current tools provide little to no guidance. In this paper, we present a new approach for the visual exploration of such parameter spaces. Given a three-dimensional scene description, we utilize sampling and spatio-temporal clustering techniques to generate a concise overview of the achievable variations and their temporal evolution. Our visualization system then allows the user to explore the simulation space in a goal-oriented manner. Animation sequences with a set of desired characteristics can be composed using a novel search-by-example approach and interactive direct volume rendering is employed to provide instant visual feedback. A user study was performed to evaluate the applicability of our system in production use.

## 6.1 Introduction

PHYSICALLY-BASED simulation is gaining increasing popularity for generating realistic animations of water, smoke, explosions, and related phenomena using computer graphics. Common modeling and animation software packages include built-in fluid dynamics simulators or offer this functionality via add-on modules. These existing tools frequently allow the user to modify the simulation parameters via standard controls such as sliders or numeric input fields. It is difficult, however, to predict the influence of changing one or several of these values. Depending on the exact scene setup, effects may be global or remain rather localized, both in space and time. Even small changes can dramatically affect the appearance of the resulting animation. Graphics artists, who aim to produce a particular visual result, therefore typically have to resort to a cumbersome and time-consuming trial-and-error approach. Moreover, as the simulation process is computationally expensive, interactive visual feedback is frequently not available. While recent advances in real-time fluid simulation help by reducing the simulation time [151, 173], the underlying problem remains: there is virtually no guidance in exploring a vast parameter space.

In this paper, we present a result-driven visual approach to navigate through this parameter space tailored to the requirements of graphics artists. Unlike scientists and engineers, who usually seek to understand and analyze the underlying physical phenomenon, these users are primarily interested in controlling the simulation in order to approximate a particular artistic vision. To facilitate this task, we sample the parameter space and apply clustering techniques in an effort to identify the characteristic spatio-temporal variations of the resulting simulations. The results of this process are presented to the user in an interactive visual exploration environment, which combines three-dimensional animated views with an abstracted representation of the identified spatio-temporal clusters. The user can interactively navigate through the space of simulations to find sequences with the desired characteristics using intuitive visual query facilities.

The main contributions of this paper can be summarized as follows: Firstly, we target an application area which, to the best of our knowledge, has not been explored before. Physically-based simulations have become a mainstay in the animation community and visualization tools designed to control the specification of their parameters can help to make the design process considerable less labor intensive. We also present a novel approach for clustering time-dependent volume data generated by sampling a high-dimensional parameter space. Furthermore, the paper describes new visualization and interaction techniques for volumetric time sequences designed to meet the requirements of graphics artists. Finally, we present a user study performed to evaluate the practical applicability of our approach to visual effects design.

## 6.2 Related Work

The visualization of general time-oriented data is an extensive field of research and Aigner et al. [2] as well as Andrienko et al. [6] provide comprehensive surveys. Our work focuses on the visualization of time-varying volume data, a topic which has been intensively studied in the context of science and engineering data [104]. In many cases, the user is interested in tracking certain features over time which can be difficult in animations of complex data. One approach is to consider the time series as a four-dimensional scalar field. Hanson and Heng [62] introduced general techniques for visualizing surfaces and volumes embedded in four dimensions and developed a 4D illumination model for this purpose. The HyperSlice method presented by van Wijk and van Liere [156] uses a matrix of orthogonal 2D slices as the basic visual representation of a multi-dimensional function. Woodring et al. [172] proposed an intuitive user interface for specifying arbitrary hyperplanes in 4D.

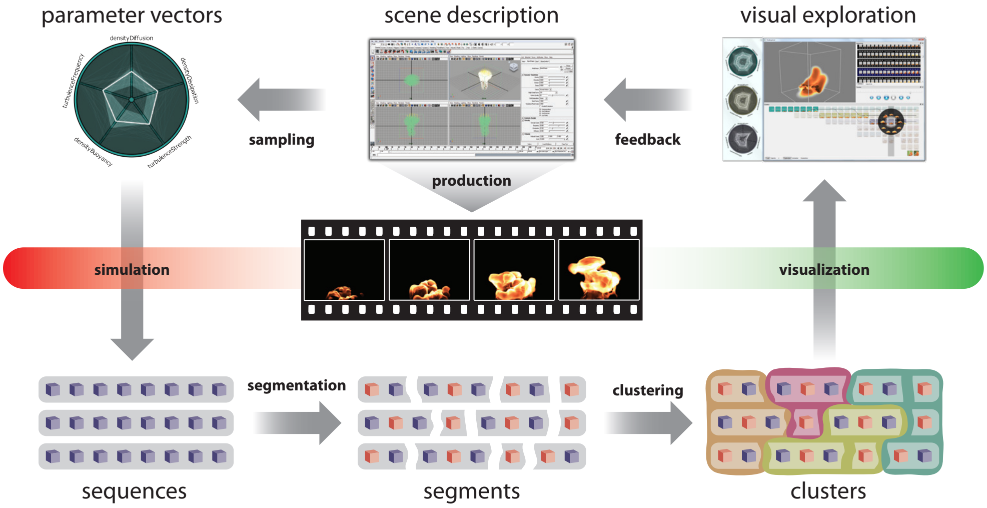


**Figure 6.2 –** Conceptual overview of our visualization system. The process starts from a scene description which defines the basic simulation scenario. Sampling generates a set of parameter vectors which are used to control the simulation process. The resulting sequences are then split into multiple short segments and clustering is applied to group these segments. The results can be interactively explored to find the desired parameter settings for the final animation.

After applying slicing or projection, the resulting volume can be displayed using standard techniques. Chronovolumes, presented by Woodring and Shen [169], use integration through time to produce a single volume that captures the essence of multiple time steps in a sequence. A further approach by Woodring and Shen [170] employs different operators to combine multiple volumes. While these methods are useful for detailed analysis and comparison, the resulting visualization can be quite abstract and difficult to grasp.

An alternative approach is to interpret the temporal progression of the data values at each point in space as a one-dimensional function referred to as a time-activity curve [48]. These curves can be used to identify spatial regions with certain properties. Muigg et al.[117] presented techniques for the visualization of a large set of these function graphs for applications such as breast tumor diagnosis. Woodring and Shen [171] applied clustering to time-activity curves to identify similar regions in space. The approach by Lee and Shen [94] attempts to identify temporal trends and models them as a state machine of trend sequence. A further approach for characterizing time-dependent volume data are time histograms which represent information about the frequency of occurrence for each data value and time. Akiba et al. [4] used time histograms to assist in the specification of transfer functions across multiple time steps. For the visualization of multi-variate time-dependent data, Akiba and Ma [3] also proposed the combination of time histograms and parallel coordinates. Our goal differs from these methods in that we do not attempt to track features over time or characterize the behavior of different regions. Instead, we want to globally investigate the similarities and characteristic variations between multiple volumetric time series.

Thus, while based on time-dependent volume data, our approach bears many similarities to methods from video processing and content retrieval. In order to overcome the sequential and time-consuming process of viewing video a noticeable amount of effort has been made to devise methods for analyzing and abstracting video data automatically [152]. A first step in many approaches is shot detection, i.e., partitioning the video into multiple series of interrelated consecutive frames. Hanjalic [60] provides a detailed overview of different methods employed for this purpose. A further step may involve clustering of these shots to extract a compact representation of the video in the form of representative key frames or preview sequences [61, 118]. Within the visualization community, Daniel and Chen [35] proposed the use of volume rendering to present summaries of video sequences. Our work draws inspiration from video analysis and abstraction methods and has many related goals such as the easy visual retrieval of data.

Dimensionality reduction and clustering are commonly employed for gaining insight into high-dimensional parameter spaces [7, 76]. Our input parameter space is also multi-dimensional, but we apply clustering to characterize the output space of simulations to extract information about visual variations over time. Furthermore, visualization and query techniques for interacting with complex sets of temporal data such as ThemeRiver [65], TimeSearcher [70], and PatternFinder [46] have inspired our work.

Finally, our approach is most closely related to techniques for design space exploration. Ma [103] introduced a visualization system which presents information on how parameter changes affect the result image as an image graph based on data generated during an interactive exploration process. Smith et al. [147] presented methods for navigating through a complex shape space of registered car models using an intuitive direct manipulation interface. The work of Monks et al. [114] discussed a system for acoustic design which applies visualization, simulation, and optimization in a goal-oriented manner. Marks et al. [108] introduced Design Galleries, a general concept for exploring parameter spaces. Our system is founded in their basic methodology of sampling the input space to generate a visual overview. In contrast to the methods presented in this paper, however, their work only discussed static output and did not address the complex issue of time-dependent data.

## 6.3 Overview

In order to distinguish our goal from that of typical simulation visualization approaches targeted at scientists and engineers, an analogy to biology can be drawn. The genome encodes the set of instructions for building a living organism. The term *genotype* refers to an organism's full hereditary information, even if not expressed. The term *phenotype*, on the other hand, refers to an organism's actual observed properties, such as morphology or behavior. Different genotypes may result in similar phenotypical characteristics during different stages of development. *Cladistics* is the systematic study of organisms based on their genetic relationships, while *phenology* attempts to classify organisms based on overall similarity regardless of their evolutionary relation. Even though most of today's evolutionary biologists favor cladistics, phenetic approaches can prove useful when studying diverse groups of closely-related organisms. Similarly, we want to provide visualization tools to explore the *simulation space*, i.e., our main focus lies in visualizing the variability in observable characteristics of a set of simulations. In this sense, our approach can be considered deliberately phenetic. In contrast, if the primary goal is to analyze the underlying *parameter space*, a cladistic approach is usually more suitable.

A conceptual overview of our system is depicted in Figure 6.2. We start from a *scene description* generated in a standard modeling/animation software package. It consists of the basic simulation settings, such as duration, geometric setup, and emitter specification. While many artists have developed an intuition which general parameters need to be tuned in order to achieve a certain result, the actual parameter values are highly dependent on the specific nature of the scene. To provide visual guidance in this selection process, our approach begins by randomly sampling a manually selected subset of the parameter space. The sampling process generates a set of *parameter vectors*. For each of these combinations of parameter values, a simulation consisting of multiple time steps in the form of volumetric grids is produced. These *sequences* may exhibit different characteristics at different points in their temporal evolution. For instance, they share the same initial state and, depending on the parameters, can diverge at varying rates. Likewise, multiple simulation sequences may start to converge to similar states as they progress. As a simple example, consider a smoke simulation: Initially smoke will rise, but, depending on the temperature, gravity will cause the smoke particles to fall again at a certain point in time. In order to capture these kinds of characteristic variations, we evaluate the spatio-temporal similarity of the generated simulations. First, a segmentation step decomposes each simulation sequence into multiple continuous *segments*. A density-based clustering algorithm is then applied to group multiple similar segments into visually distinct phases. The results of this classification process are presented in an easily-understandable layout for interactive *visual exploration*. The user can inspect the different variations and use intuitive interaction tools to find sequences which exhibit the desired spatial and temporal characteristics. The corresponding parameters, or the already generated simulation, can then be used for production of the final animation.

The remainder of this paper details these individual steps and components. Section 6.4 is devoted to sampling, segmentation, and clustering, while Sec-

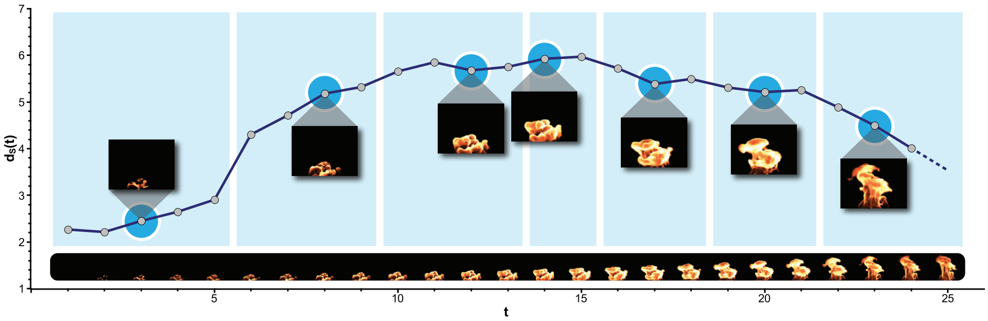


**Figure 6.3 –** Sequence segmentation. A sequence of 25 time steps split into 7 segments using our algorithm is shown. The graph depicts the dissimilarity $d_S(t)$ between two subsequent time steps $t$ and $t+1$. The highlighted points indicate the selected representative time steps for each segment.

tion 6.5 focuses on visualization and interaction techniques. Implementation details are discussed in Section 6.6. Section 6.7 presents the results of a user study performed to evaluate the suitability of our system for production use. The paper is concluded in Section 6.8.

## 6.4 Sampling and Clustering

In this section, we describe the individual processing steps which form the basis of our approach.

### 6.4.1 Sample Generation

The high computational costs of fluid simulation severely constrain interactive exploration within the authoring environment. In an effort to eliminate the cumbersome trial-and-error process of changing a parameter value, waiting for the result to compute, and then deciding whether the desirable effect has been achieved, we generate random simulation samples in an offline process. While this may seem costly, both in terms of processing time and storage demands, it has the considerable advantage that it can be performed without requiring user intervention, e.g., overnight. Animation studios are typically equipped with render farms, so this setup fits well into the environment of our intended users. Initially, the user chooses a set of $M$ simulation parameters:

$$P = \{p_1, p_2, \ldots, p_M\} \tag{6.1}$$

where each parameter $p_i \in P$ has an associated range of interest $R_i = [a_i, b_i] \subset \mathbb{R}$. This choice is mostly influenced by the desired effect and the physical interpretation of these parameters. A selected number of $N$ samples of this $M$-dimensional parameter space will be generated. We refer to each combination of simulation parameter values as a *parameter vector* $\mathbf{x} \in \mathbb{R}^M$ :

$$\mathbf{x} = (x_1, x_2, \ldots, x_M) \tag{6.2}$$

with $x_i \in R_i$. For each parameter vector $\mathbf{x}$, the simulation module then generates a *simulation sequence* $S(\mathbf{x})$ written as a set of $T$ time steps:

$$S(\mathbf{x}) = \{s_1, s_2, \ldots, s_T\} \tag{6.3}$$

where each $s_i$ is a volumetric grid. Depending on the type of simulation, each grid point may store multiple attributes, such as density, temperature, pressure, etc. For simplicity, the remainder of this paper will focus only on scalar output, but our methods equally apply to multi-channel data. For most common effects density and temperature are simulated and typically mapped to, respectively, opacity and color. The sampling process generates $N$ sequences consisting of $T$ volumes. For most visual effects, simulations will be rather short with $T$ ranging from tens to a few hundred frames. Grid sizes vary depending on the specific effect, but are typically smaller than for other common types of volume data such as medical scans. $N$ should be chosen according to the number of parameters, but is constrained by the simulation cost in terms of processing time and disk space requirements.

In our current implementation, we use unconstrained random sampling as it permits the easy addition of further samples as well as termination of the sampling process at any time. However, to ensure a more uniform coverage of the parameter space alternative schemes such as Latin hypercube sampling may be used instead. One major practical advantage of random sampling is that the exploration of intermediate results is easily possible.

### 6.4.2 Sequence Segmentation

To facilitate robust clustering as well as to reduce the computational load of the subsequent processing steps, our approach first splits each volumetric time sequence $S$ into multiple short *segments* $S' \subseteq S$ of varying length. It is important to note that, at this point, we are not concerned with identifying overall similarity. Rather, we want to divide each simulation sequence into a smaller number of manageable units which exhibit high similarity and are continuous in time. Here, we draw inspiration from the field of video processing. Many methods for generating an overview of a video clip start by dividing the input into multiple shots by detecting discontinuities [60]. In contrast to these methods, however, our approach groups neighboring time steps as a simulation will in general not exhibit distinct boundaries.

For each sequence $S$, we compute a dissimilarity measure between neighboring time steps of a sequence using the sum of squared intensity differences over all grid points of the corresponding volumes:

$$d_S(t) = \sum_{\mathbf{u}} (s_{t+1}(\mathbf{u}) - s_t(\mathbf{u}))^2 \tag{6.4}$$

where $s_t(\mathbf{u})$, $s_{t+1}(\mathbf{u})$ are the data values at the three-dimensional grid position $\mathbf{u}$ of two subsequent time steps with $t \in [1, T-1]$. We then use a simple greedy algorithm which merges neighboring time steps based on their dissimilarity. Initially, each time step forms its own segment. The cost value associated with each segment is initialized to zero. We then iteratively merge two neighboring time steps if the dissimilarity at their boundaries added to their individual costs is minimal. The cost of the resulting segment is updated to this sum. This process proceeds at least as long as the number of segments is larger than a specified value for the maximum segment count. After that, the algorithm terminates when the minimum cost exceeds a threshold. The first parameter, the maximum number of segments, is set according to the available computational resources – a larger number of segments will increase the time required for the subsequent clustering step. For the cost threshold, we use the average dissimilarity of the sequence.

The result of this algorithm is a varying number of segments for each simulation sequence. The representative time step $r(S')$ for a segment $S'$ is chosen such that it minimizes the absolute difference between the cumulative dissimilarity of its predecessors and successors within the segment:

$$r(S') = \operatorname*{arg\,min}_{s_i \in S'} \left| \sum_{s_j \in S', j<i} d_S(j) - \sum_{s_j \in S', j>i} d_S(j-1) \right| \tag{6.5}$$

For segments with only two members, the lower time step is chosen. In the subsequent clustering step, all of the individual time steps represented by one such segment are treated as a unit and the representative time step is used in their place.

Figure 6.3 depicts an example of the sequence segmentation process. A short sequence of 25 time steps is split into 7 segments. The graph shows the dissimilarity $d_S(t)$ for $t \in [1, 24]$ – note that for the last time step $t = 25$ of the sequence, this function is undefined. The highlighted points indicate the chosen representative time steps and the images show renderings of the corresponding volumes. No minimum number of segments was specified and the cost threshold was set to the average dissimilarity.

### 6.4.3 Density-based Clustering

Having split each simulation sequence into a number of representative segments, we now aim to compare the simulation space on a global level, i.e., we want to identify similar phases or states which may occur at different points within the temporal evolution of each simulation. For this purpose, we employ a density-based clustering approach. In contrast to partitional and hierarchical approaches, density-based clustering uses a local cluster criterion, in which clusters are defined as regions in the data space where the data points are dense, separated from one another by low-density regions. In particular, we employ a variation of the DBSCAN algorithm [43], as it has the ability of discovering clusters with arbitrary shape and does not require the predetermination of the number of clusters. This is advantageous, as it allows us to make minimal assumptions about the similarity relationships in simulation space.

DBSCAN requires two parameters: $\varepsilon$, which defines the maximum distance between two points considered to be neighbors, and $p_{\min}$, the minimum number of points required to form a cluster. The algorithm starts with an arbitrary point that has not been visited. This point's $\varepsilon$-neighborhood is retrieved, and, if it contains sufficiently many points, a cluster is started. Otherwise, the point is marked as noise. This point may later be found to be in a sufficiently-sized $\varepsilon$-environment of a different point and hence still become part of a cluster. If a point is found to be part of a cluster, its $\varepsilon$-neighborhood is also part of that cluster. Thus, all points that are found within the $\varepsilon$-neighborhood are added to its cluster, as is their own $\varepsilon$-neighborhood. This process continues until no further points can be found. Then, a new unvisited point is processed.

In our case each segment, identified by its representative time step, corresponds to one point. For the neighborhood size, which allows DBSCAN to judge local density, however, the point contributes with the number of members of the corresponding segment. This enables the information gathered during sequence segmentation to influence the clustering process. An important choice is the dissimilarity measure employed in the clustering algorithm – as motivated in Section 6.3, we are primarily interested in visualizing the simulation data in terms of their observable characteristics. Feature-based distance metrics have shown to have many advantages for various clustering tasks and much work has been devoted to developing techniques for extracting features in scalar- as well as vector-valued volume data [146]. However, most of these methods require several parameters and are tailored to specific tasks. Since we want to compare hundreds of volumes, manual parameter selection is not an option (indeed, the main motivation of our work is to simplify parameter specification). Moreover, it would be difficult to define a feature vector which provides a robust basis for comparing simulation time steps generated across the full range of the parameter space. Thus, instead of attempting to extract explicit features, we use a rather simplistic dissimilarity measure based on the sum of squared intensity differences between two volumes $v_1$ and $v_2$:

$$d(v_1, v_2) = w(v_1, v_2) \sum_{\mathbf{u}} (v_1(\mathbf{u}) - v_2(\mathbf{u}))^2 \tag{6.6}$$

with

$$w(v_1, v_2) = 1 + \begin{cases} \sqrt{|\mathrm{ti}(v_1) - \mathrm{ti}(v_2)|} & \text{if } \mathrm{si}(v_1) = \mathrm{si}(v_2) \\ 0 & \text{otherwise} \end{cases} \tag{6.7}$$

where $v_1(\mathbf{u})$ and $v_2(\mathbf{u})$ are the data values at the three-dimensional grid position $\mathbf{u}$, $\mathrm{ti}(v_1)$, $\mathrm{ti}(v_2)$ are the time step indices, and $\mathrm{si}(v_1)$, $\mathrm{si}(v_2)$ are the sequence identifiers of, respectively, the volumes $v_1$ and $v_2$. The additional weight increases the dissimilarity of time steps within one sequence based on their temporal differences. This allows the clustering algorithm to group similar temporal progressions between different sequences even if they are not entirely synchronous. The approach is related to the measures proposed by Birant and Kut [16] even though they focus on geographic time series data. The choice of the sum of squared differences as the basis for our measure is motivated by its frequent use in image registration tasks [164]. While other measures, such as mutual information, may perform better they also come at significantly higher computational costs.

The clustering step then proceeds as follows: First, we compute the dissimilarity matrix by comparing each pair of segment representatives. Next, for each segment representative, a neighborhood index is generated by sorting the dissimilarity values in ascending order. The DBSCAN algorithm is then executed resulting in a number of clusters. If a segment representative is part of a cluster, all members of the corresponding segment are assumed to share this association.

For specifying the parameters of the algorithm, we use a simple heuristic [43]. The minimum number of points $p_{\min}$ is set to:

$$p_{\min} = \ln\left(\frac{1}{\overline{|S'|}} NT\right) \tag{6.8}$$

where $\overline{|S'|}$ is the average number of members in a segment, $N$ is the number of sequences, and $T$ is the number of time steps per sequence. We set $\varepsilon$ to the average dissimilarity between all segments. In our experiments these values have shown to be robust defaults. However, as the time required for executing the actual clustering algorithm once the dissimilarity matrix has been computed is negligible, these values can also be adjusted easily if an unusually low or high number of clusters is detected.

In addition to references to its members, each cluster stores the following additional information computed after the clustering algorithm has completed:

**Cluster medoid** – the cluster member which minimizes the average dissimilarity to all other members in the cluster.

**Sequence range** – the set of sequence identifiers which have at least one member in the cluster.

**Temporal range** – the set of time step indices covered by the members of the cluster.

**Temporal medoids** – for each distinct time step index contained in the cluster, the member which minimizes the average dissimilarity to all cluster members of the same time step.

In the following section we discuss how this information is used to generate a compact representation of the simulation space's temporal evolution.

## 6.5 Interactive Exploration

Having identified spatio-temporal clusters in the set of sampled simulations, we want to present this information, together with the original simulation sequences, in an easily-understandable manner. The general layout of our interactive visualization system is shown in Figure 6.1. The user interface consists of several different linked views. The *animation view* shows a volume rendering of the currently selected sequence and is controlled by a standard time slider. The *sequence view* allows the user to browse through all available simulation sequences. The *cluster timeline* is the main element for our application – it gives an overview of the visual variations across the simulated sequences over their temporal range and allows the user to search for sequences with particular characteristics. Finally, the *parameter view* provides a visualization of the parameter space variations for a selected set of clusters. All views are linked, so selecting a particular sequence in the sequence view, for example, will update the animation view and highlight the corresponding elements in the other views.

### 6.5.1 Animation View

As the final result of the process assisted by our visualization system is an animation sequence, it is important to provide an interactive preview. The animation view depicts a volume rendering of the currently selected sequence. It allows viewpoint manipulation and playback using a standard time slider and animation controls. Rendering of time-dependent volume data is an active area of research and many powerful techniques capable of dealing with large data sets have been presented [104]. As this is not the focus of our paper, we will only briefly describe our setup. Our system features two different volume renderers: An emission/absorption ray caster implemented in CUDA, and a slice-based renderer which supports self-shadowing and scattering approximations based on a conical phase function. The latter renderer uses OpenGL since the slice-by-slice processing scheme required by its illumination model performs significantly better in OpenGL than a comparable CUDA implementation as it can exploit fixed-function GPU operations. The CUDA renderer offers better overall performance, while the superior optical model of the OpenGL renderer provides higher fidelity. The user can switch between these two renderers at runtime.

### 6.5.2 Sequence View

The sequence view provides a simple overview of all simulated sequences by depicting a ″film strip″ of their time steps. One of its purposes is to allow the user to establish a mental model of the visualization process. The sequence view displays all simulation sequences. Hence, the animation view which only displays a single time step at a time, and the cluster timeline, which provides a summarized and abstracted view of the sequences' temporal progression, can be interpreted as filtered representations of the data. The sequence view is linked to the other views, so whenever the current sequence is changed it is scrolled into view and highlighted. While it would also be possible to use the results of sequence segmentation for selecting the depicted images, we instead choose to uniformly divide the time range so the chosen time steps are the same for all sequences resulting in a more traditional presentation familiar from common animation and video processing software. Depending on the width of the view, the number of images is adjusted to fill the available space. The displayed images are live thumbnails, i.e., they are updated whenever settings such as the current viewpoint change. Image generation is performed in the background using our CUDA volume renderer and the resulting thumbnails are cached in main memory.

### 6.5.3 Cluster Timeline

The cluster timeline provides a concise overview of the visual variations over the duration of the simulation while summarizing the similarities between different sequences and represents the key component of our interface. The general idea behind this visualization is to consider the identified clusters as distinct phases in the temporal evolution of a simulation. Multiple sequences may enter a particular phase at different points in time as they progress. Since, depending on the nature of the simulation, potentially many of these phases exist, the generated layout should be compact. Our dissimilarity measure is successful in grouping similar segments, but since it is not feature-based the inter-cluster distance is less informative and we choose not to visually encode it which provides more freedom in designing a compact layout. While our experiments have shown that the identified clusters tend to cover continuous time ranges, we do not explicitly enforce temporal continuity so the visualization algorithm needs to be capable of handling discontinuous clusters as well. Finally, as artists are used to dealing with linear time scales in their standard tools, we choose not to distort the time axis.

Based on these general guidelines, we developed a simple layout algorithm which visualizes the temporal distribution of clusters and their membership relationships. The cluster layout is generated in the following manner: Initially, all clusters $C$ are assigned a global rank $r_G(C)$ defined as the product of the total number of time steps covered by the cluster and the number of distinct

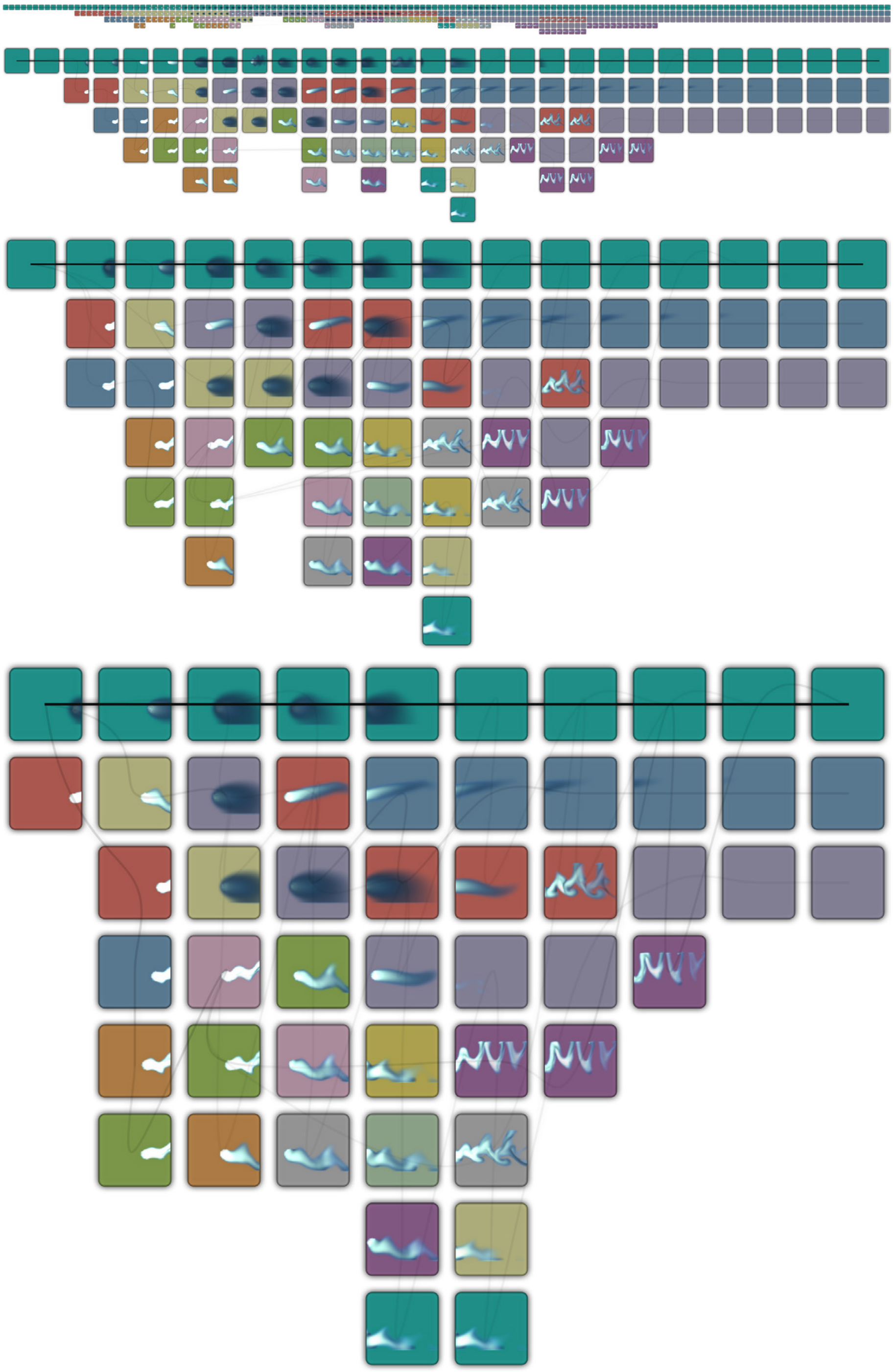


**Figure 6.4 –** Cluster timeline at different temporal compression levels. The depicted time interval sizes are, from top to bottom: 1, 5, 10, and 15. The sequences in this set of 128 simulations of a bullet passing through a medium have 150 time steps each.

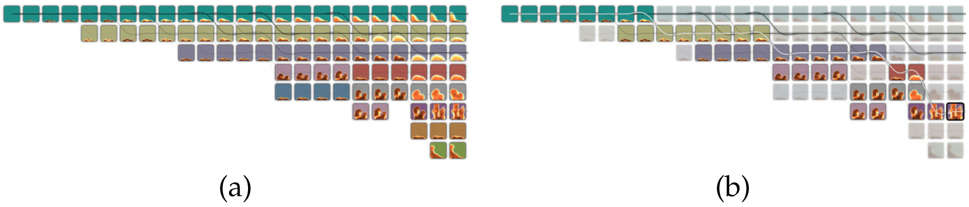


**Figure 6.5 –** Interaction with the cluster timeline. (a) No selection has been made, so all cluster items remain active. (b) A cluster item has been selected, the corresponding sequence path is highlighted, and unconnected items are dimmed. The depicted data set is a fire effect consisting of 128 sequences, each with 25 time steps.

sequences it contains. This means that clusters which cover a large temporal range and/or include many different sequences will be ranked higher. Each cluster is assigned a color using one of the ColorBrewer's [63] qualitative color schemes. As it is generally advised against attempting to visually encode too many classes using color, the assigned cluster colors may not be unique. If the number of total clusters exceeds the number of colors in the scheme (the maximum is 12), we maximize the temporal difference between clusters which are assigned the same color.

*Cluster items* are then positioned on the canvas by traversing the temporal simulation range. One cluster item represents a subset of the cluster's members which share a common time interval. Thus, the number of associated cluster items varies depending on the temporal extent of a cluster. For every non-overlapping time interval $[t_s, t_e)$ and every cluster $C$, the number $r_T(C, t_s, t_e)$ of cluster members which are contained in the interval is determined. All clusters where this number is greater than zero are sorted in descending order according to the product of $r_G(C)$ and $r_T(C, t_s, t_e)$ and one item is created for each of these clusters. The horizontal position of the item is determined by the current interval $[t_s, t_e)$ while the vertical position (from top to bottom) corresponds to the sorting order. The visual representation of a cluster item consists of a background rectangle in the cluster color and a live thumbnail image depicting a rendering of the cluster's temporal medoid for the current interval, i.e., the cluster member which minimizes the average dissimilarity to all other members within the same time interval.

To provide an overview of the temporal progression of the individual simulation sequences, a *sequence path* is generated for every sequence. The path represents the progression of cluster memberships of a sequence over time and is displayed as a cubic spline connecting the centers of all cluster items the sequence is a member of. It is drawn using an opacity based on the total number of sequences which enables the identification of membership patterns that occur more frequently. The path of the current sequence, i.e., the one that is displayed in the animation view and selected in the sequence view, is emphasized and drawn with full opacity on top of all other paths.

The resulting layout depicts, for every time interval, the possible variations identified in the clustering process. Due to the influence of temporal range on the sorting order, clusters covering many time steps appear first. The user can control the size of the interval interactively using a slider. When changing the interval size, the size of the cluster items on screen remains the same, but the time axis gets compressed. Cluster items which represent the same cluster merge, while those of different clusters stack on top of each other according to their rank. At the highest zoom level, i.e., the interval spans the entire vertical range, all clusters are listed vertically according to their rank. This sort of temporal compression enables viewing of long sets of sequences without scrolling, while preserving their salient features and variations. As a sequence path may connect several cluster items at the same horizontal position for large intervals, we only connect them to the highest ranked cluster item in these cases.

Figure 6.4 shows an example of the cluster timeline at different temporal compression levels. The sequences in this set of 128 simulations have 150 time steps each. The depicted time interval sizes are, from top to bottom: 1, 5, 10, and 15. Note that on screen the cluster items always have the same size and the view scrolls horizontally.

### Search-By-Example

To enable result-driven exploration, the user can interact with the cluster timeline. Selecting a cluster item will highlight all other cluster items which have members that connect to it, i.e., there is a sequence which is a member of both clusters within the time interval of the item. Multiple items can be selected thereby further filtering the view. Whenever such a selection is made, the current sequence displayed in the animation and sequence views is instantly updated to the best match of the query and the corresponding sequence path is highlighted in the cluster timeline. When the selection is modified, candidate sequences, i.e., those which connect to all selected cluster items, are ranked by the number of their time steps which are members of the corresponding clusters. Of those sequences with the maximum number of overlapping time steps, the one which minimizes the dissimilarity to the temporal medoids of all selected cluster items is chosen. This strategy prefers sequences with longer membership times in the clusters corresponding to the selected items, i.e., their sequence paths will tend to be more straight.

An example for this type of interaction with the cluster timeline is shown in Figure 6.5. The cluster timeline for a flame effect simulation is depicted in Figure 6.5 (a). When a cluster item is selected, as shown in Figure 6.5 (b), all items which share no connection with the selected item are dimmed. The sequence path of the best query match is emphasized. The cluster items which remain active indicate the possible variations which share a similar end state.

This intuitive visual query metaphor enables quick identification of sequences with the desired spatial and temporal properties, or alternatively, that there are no such sequences. The linked animation view additionally adds to the flexibility of this approach for finding desired simulation characteristics. When selecting a cluster item, the sequence depicted in the animation view changes according to the best match of the query, but the time position remains unchanged and can be controlled independently by the time slider. Only when an item is double-clicked, the time slider is moved to the start of its interval. This enables the user to, for instance, quickly switch between different variations early in the temporal progression and get a three-dimensional view of their evolution at a later point in time.

To explore the variations within a cluster, a radial context menu can be opened by right-clicking a cluster item. The menu depicts the nearest neighbors of the cluster item's temporal medoid which are part of the same cluster, but not necessarily at the same time step. By clicking on one of the displayed thumbnails, the corresponding sequence is selected. This enables navigation within a cluster. The context menu is visible in the screenshot shown in Figure 6.1.

### Sequence Blending

Sometimes, a particular desired temporal progression may not occur in the set of simulations. This can be due to the limited number of samples, but it may also be the case that it is physically impossible given the chosen set of simulation parameters. Nonetheless, artists will often sacrifice physical plausibility for achieving a desired result. Animation packages usually include functionality to combine different simulation runs. In our system, we allow the user the possibility to specify blending between sequences directly in the cluster timeline. A cluster item can be marked as a key frame, i.e., the best matching sequence for the item, as described previously, will be displayed until the end of the cluster item's time interval. If a further key frame is selected, a transition between the two sequences will occur in the animation view within the time interval between the two cluster items. Blending is performed in volume space using on-the-fly interpolation between the corresponding time steps of both sequences based on a user-specified easing curve. An extension of these simple animation facilities using approaches such as those presented by Wohlfart and Hauser [168] or recent work by Akiba et al. [5] could also be an interesting direction for further research.

## 6.5.4 Parameter View

While, as initially stated, it is not our primary goal to facilitate detailed analysis of the parameter space, it is still useful to provide an overview of the parameter variations within a cluster. For this purpose, we employ circular parallel-coordinate plots inspired by DataRoses as proposed by Elmqvist et al. [42]. For each selected cluster, the parameter view shows a star plot layout where each of the simulation parameters corresponds to one of the equiangular axes. As it is common to facilitate side-by-side comparisons, all star plots use the same scaling with the minimum of the parameter range at the center and the maximum located at the radius of the circle for each axis. Since we want to visualize the parameter distribution in relation to its visual manifestation, the depicted parameter vectors are weighted indirectly proportional to the dissimilarity of the corresponding segment to the cluster medoid. The weights are scaled such that the medoid is assigned a value of one and the member with the highest dissimilarity to the medoid receives a weight of $|C|^{-1}$ where $|C|$ is the number of cluster members. The parameter vector for each cluster member is then depicted as a polygon with an opacity proportional to its weight. The weighted mean of parameter vectors within the cluster is shown as thick white polygon with full opacity. Additionally, the region enclosed by the first and third weighted quartile is highlighted.

Figure 6.6 depicts the parameter view for three clusters of a flame effect together with a rendering of the final time step of a cluster member. Note how a correspondence between lower dissipation and a more typical fire-like appearance is indicated when observing the distribution of parameter vectors within each star plot. The corresponding cluster timeline is shown in Figure 6.5.

## 6.6 Implementation

Our system was implemented in C++ using the Qt cross-platform application framework and consists of two basic parts: the stand-alone visualization application and a processing module. The processing module was implemented as a plugin-in for Autodesk Maya. The sampling process can be initiated using a command in Maya's scripting language MEL or can be bound to a user-interface element. Both modules communicate via sockets, and can therefore be used over the network.

In order to provide a fully responsive interactive experience without delays, we heavily rely on parallelism at several different levels. Even if individual volume dimensions may be comparably small (they usually do not exceed dimensions of $128^3$), the typical amount of data still consists of several hundreds of these volumes and cannot be held in main memory. We employ NVidia's CUDA GPU computing platform, which allows synchronous execution of GPU processing tasks and memory transfers. Both can also be performed concurrently with CPU processing using CUDA's stream concept. This high degree of parallelism allows for excellent latency hiding and a fully controllable memory footprint. In our architecture, each individual volume – referred to as a data item – is identified by a unique index. A component

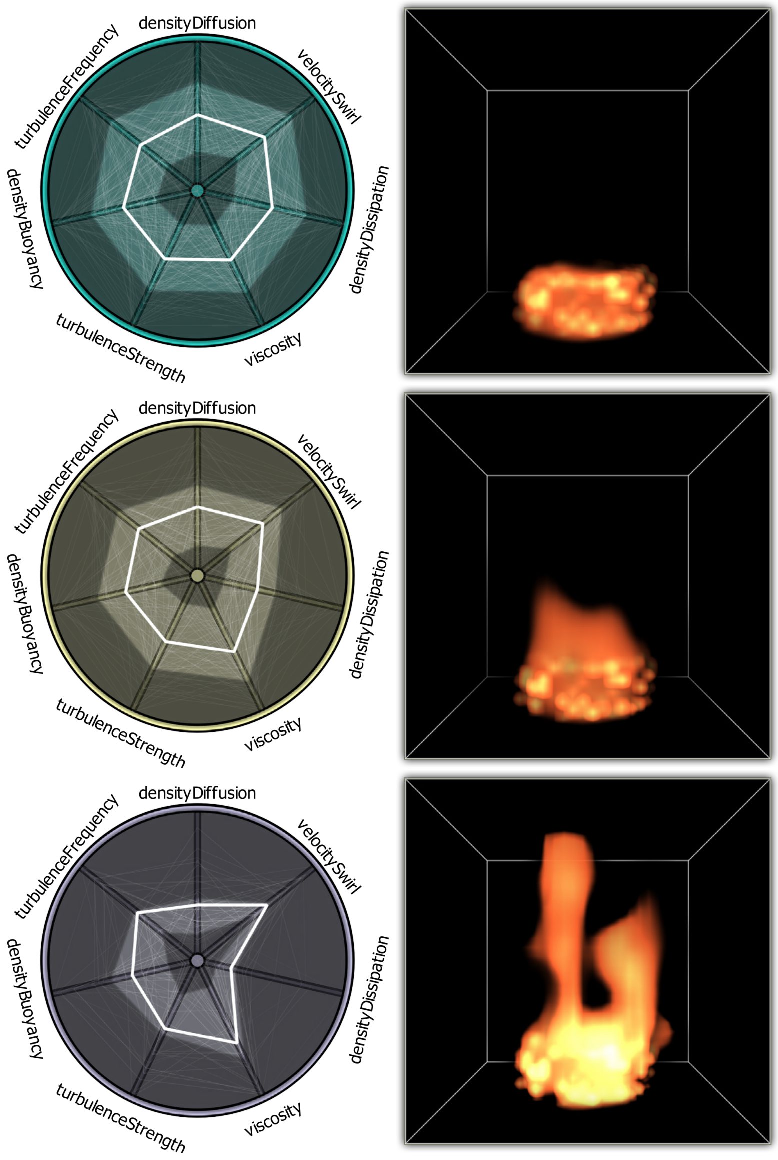


**Figure 6.6 –** Parameter view for three clusters of a set of flame simulations together with renderings of representative cluster members.

which requires access to one or multiple data items places an asynchronous request for the desired index range and continues operation. The data access component collects and schedules these requests according to priority and access pattern. Two memory pools are used to cache data: one in main memory and one on the GPU. When placing a data request, the caller can indicate whether it wants to access the data on the CPU, the GPU, or both and cache replacement is performed accordingly. For example, if the data is only

**Table 6.1 –** Statistics and performance numbers for the processing of two data sets. Timings are given for simulation, segmentation of the simulation sequences, computation of the dissimilarity matrix, and clustering. System configuration: Intel Core 2 Duo 2.53 GHz CPU, 4 GB RAM, NVidia GeForce 9600M GT GPU.

| **Data set** | **Flame** | **Bullet** |
|---|---|---|
| Sequences | 128 | 128 |
| Time steps | 25 | 150 |
| Resolution | $30 \times 30 \times 30$ | $100 \times 40 \times 30$ |
| Disk space | 230 MB | 8.8 GB |
| Segments | 899 | 3183 |
| Clusters | 8 | 22 |
| Simulation | 23 min | 448 min |
| Segmentation | 1 min | 7 min |
| Dissimilarity | 2 min | 76 min |
| Clustering | $< 1$ s | 8 s |

required on the GPU, e.g., for volume rendering, the corresponding slot in the main memory pool can be marked as available for replacement immediately after transfer to GPU memory has been completed. A background thread is responsible for loading data items from hard disk into main memory and, if requested, initiates transfer to GPU memory. For each data item in the requested index range a notification is sent when it is available in the indicated memory pool. Before the requesting component is notified, the data item is marked as locked to prevent cache replacement until processing has finished. The requesting component is responsible for releasing the item as soon as the data is no longer required. Both the CPU and the GPU memory pool use a Least-Recently Used (LRU) cache replacement policy. Requests can also be marked as optional, i.e., they are scheduled whenever no other items are being transferred. In this case, no locking occurs and the caller is not notified of availability. This is useful for prefetching data during animation rendering.

## 6.7 Evaluation

While we described the capabilities of our system and attempted to illustrate them in static images, it is difficult to fully capture interactive processes in this manner. We therefore refer the reader to the accompanying video for a live demonstration. Table 6.1 lists information on the depicted data sets as well as simulation and preprocessing times.

The research presented in this paper was motivated by animation professionals who deal with the problem of missing visual guidance in choosing simulation parameters on a daily basis. Their requirements guided our development process. In order to evaluate the functionality, usability, and potential practical impact of our system, as well as to identify areas which require further research, we performed a user study. Based on previous demonstrations to practitioners and our own experience with the system, we had the following hypotheses:

1. Our general approach for visualizing fluid simulations for visual effects design will be considered useful and valuable.
2. The cluster timeline will be considered helpful in exploring the variations in the set of simulations.
3. There will be difficulties in understanding the parameter view and it will be considered less helpful.

The study was performed one participant at a time using the following protocol: Each participant was first asked to fill out a background questionnaire and then received a general verbal introduction into the concepts behind our approach, followed by a live tutorial on how to use the software. Each participant also received a one-page summary of the mouse and keyboard mappings in the application. Next, the participant was asked to perform a list of simple tasks. We employed the think-aloud protocol, i.e., the participants were asked to verbalize their thoughts and actions. We specifically designed the tasks to be open-ended and to rely on the subjective judgement of the user, for example *"Find the simulation sequence that provides, in your judgement, the most realistic appearance"*. No time limit was given and the study participants were encouraged to freely explore all aspects of the application. Screen capture and audio recordings were made to document the user interaction with the system. After completing the tasks, the participants were asked to fill out a post-questionnaire in which they had to rate 25 statements on a 5-point attitude Likert scale. The questionnaire covered general application functionality, suitability and difficulty of tasks, as well as the assessment of individual components (e.g., *"I found the cluster timeline was more helpful in completing the tasks than the sequence view"*). Additionally, it also included the ten items of the System Usability Scale (SUS). This evaluation technique provides a global assessment of overall usability and user satisfaction on a scale ranging from 0 to 100 and has been shown to yield reliable results even for small user groups [10]. Finally, a semi-structured interview consisting of questions on the overall impression as well as several specific topic areas was performed.

Using this protocol, the study was performed on a total of 12 subjects (9 male, 3 female) divided into two groups. Group A consisted of 7 (5 male, 2 female) interactive arts students with moderate to expert knowledge in 3D modeling and animation, but generally less experience with fluid simulation. Group B consisted of 5 (4 male, 1 female) visual effects professionals with several years of expertise in the subject matter who routinely employ fluid simulation in their work. While the targeted duration of an evaluation session was approximately one hour per person, there were considerable variations with some subjects choosing to spend more time on exploring different aspects of the system and/or making extensive comments during the interview.

All participants agreed that the functionality provided by the system was useful. Most subjects found that the cluster timeline provided a good summary of the variations within the set of simulations (10 agree, 2 disagree). Interestingly, while all subjects from group B found that the cluster timeline was useful in completing the tasks, the corresponding scores of group A showed more variability (4 agree, 1 disagree, 2 neutral). Similarly, most subjects from group B (4 agree, 1 disagree) thought that the cluster timeline was more helpful than the sequence view in completing the tasks, while the response of group A was less uniform (3 agree, 2 disagree, 2 neural). Only a minority of the subjects found the parameter view useful (5 agree, 3 disagree, 4 neutral). In general, the selected tasks were considered to be easy to understand (10 agree, 2 disagree), appropriate for assessing the functionality of the system (8 agree, 1 disagree, 3 neutral), and relatively easy to complete (8 agree, 1 disagree, 3 neutral).

Most subjects found that the system was generally easy to use (10 agree, 1 disagree, 1 neutral). However, the average SUS score of 62.7 with a standard deviation of 13.1 also indicates that our current prototype needs additional work on user interface and interaction design. According to the work of Bangor et al. [11], this corresponds to an adjective rating between *"OK"* (50.9) and *"Good"* (71.4). During the interviews, we identified several issues that negatively affected the usability assessment. For example, almost all participants found the highlighting of the currently selected sequence path too subtle and therefore had difficulties identifying it. Furthermore, many participants found it hard to understand the relationship between individual simulation sequences and clusters. The behavior when selecting multiple items in the cluster timeline was also considered to be confusing by several subjects. The most commonly requested features were a way to easily mark certain simulation sequences as favorites and the ability to compare them side-by-side. As we had anticipated, many participants did not find the parameter view particularly useful. However, several subjects indicated that the integration of additional interaction functionality such as the filtering of parameter values would make this component more valuable.

In general, the response of the visual effects specialists was particularly encouraging. All of them indicated that they would like to use our system in production as soon as some minor quirks are resolved and some of them were interested in having the current version of the software installed on their workstations immediately. The ability to visually explore parameter variations was highly appreciated, and comments such as *"This can save me many hours of work"* and *"Instead of spending my time doing guesswork, this system* *does the guesswork for me"* left us with the impression that our approach can provide a valuable addition to the production process. A common use-case the professionals found particularly exciting was the ability to easily generate multiple variations of a particular effect for review by supervisors and clients. Additionally, there were many requests about applying our general concept to other common tasks such as cloth and crowd simulation. Based on the overwhelmingly positive feedback we received, we are currently working on addressing the main usability issues and integrating feature suggestions. Within the upcoming months, we plan to provide an updated version of the software to the visual effects artists for beta testing in production use.

## 6.8 Conclusion

In this paper, we presented a visualization system for the exploration of simulation parameter settings in visual effects design. Current software tools provide no visual guidance in the parameter selection process and artists have to resort to a time-consuming and cumbersome trial-and-error strategy. Our system samples the parameter space and employs a novel approach for clustering the resulting volumetric time sequences in order to discover characteristic variations in relation to their temporal evolution. Our novel visual representation of the clustering results enables exploration of the simulation space at different temporal levels-of-detail and provides an instant overview. Using our result-driven interactive visual exploration environment gives users the ability to find simulation sequences based on a particular artistic vision.

The main contribution of this paper is the general concept of a visualization system for the phenological exploration of simulation data. To the best of our knowledge, our system is the first that specifically attempts to make modeling of difficult natural phenomena accessible to the non-technical practitioner. Our approach for segmenting and clustering volumetric time series deliberately makes minimal assumptions and attempts to classify the data based on their observable characteristics. Although our methods were developed with a specific application in mind, the proposed techniques may also be useful in other scenarios. In particular, cloth modeling and other animation effects that are based on intrinsically computationally expensive or mathematically complex models may benefit from the described methods. Furthermore, the investigation of the parameter distribution in relation to clusters solely based on the characteristics of the output data may be an interesting alternative approach to conventional analysis methods for general simulation data. A further direction for future work involves the integration of computational steering into our system. Ideally, one would like to be able to derive the parameters for a set of desired output characteristics from a sparse set of samples. One approach could be to employ key frame information specified using our blending mechanism directly as the basis for an optimization strategy based on a multi-objective evolutionary algorithm [120].

# Bibliography

[1] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of ACM CHI*, pages 619–626, 1992.

[2] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data - a systematic view. *Computers & Graphics*, 31 (3):401–409, 2007.

[3] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of EuroVis 2007*, pages 115–122, 2007.

[4] H. Akiba, K.-L. Ma, and N. Fout. Simultaneous classification of time-varying volume data based on the time histogram. In *Proceedings of EuroVis 2006*, pages 1–8, 2006.

[5] H. Akiba, C. Wang, and K.-L. Ma. AniViz: A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications*, 30(5):61–71, 2010.

[6] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541, 2003.

[7] M. Ankerst, S. Berchtold, and D. A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of IEEE InfoVis 1998*, pages 52–60, 1998.

[8] R. S. Avila, L. M. Sobierajski, and A. E. Kaufman. Visualizing nerve cells. *IEEE Computer Graphics and Applications*, 14(5):11–13, 1994.

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

[10] A. Bangor, P. Kortum, and J. Miller. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, 24(6):574–594, 2008.

[11] A. Bangor, P. Kortum, and J. Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4(3):114–123, 2009.

[12] J. F. Barrett and K. Nicholas. Artifacts in CT: Recognition and avoidance. *Radiographics*, 24(6):1679–1691, 2004.

[13] L. D. Bergman, B. E. Rogowitz, and L. A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings of IEEE Visualization 1995*, pages 118–125, 1995.

[14] L. Bertrand and J. Nissanov. The neuroterrain 3D mouse brain atlas. *Frontiers in Neuroinformatics*, 2:3, 2008.

[15] G. Bezgin, A. T. Reid, D. Schubert, and R. Kötter. Matching spatial with ontological brain regions using java tools for visualization, database access, and integrated data analysis. *Neuroinformatics*, 7(1):7–22, 2009.

[16] D. Birant and A. Kut. ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

[17] J. G. Bjaalie. Localization in the brain: New solutions emerging. *Nature Reviews Neuroscience*, 3:322–325, 2002.

[18] A. H. Brand and N. Perrimon. Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. *Development*, 118(2):401–415, 1993.

[19] S. Bruckner and M. E. Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum*, 28(3):775–782, 2009.

[20] S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010.

[21] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.

[22] S. Bruckner, S. Grimm, A. Kanitsar, and M. E. Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 2006.

[23] S. Bruckner, V. Šoltészová, M.E. Gröller, J. Hladůvka, K. Bühler, J. Y. Yu, and B. J. Dickson. BrainGazer – visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6): 1497–1504, 2009.

[24] G. A. P. C. Burns, W.-C. Cheng, R. H. Thompson, and L. W. Swanson. The NeuARt II system: A viewing tool for neuroanatomical data based on published neuroanatomical atlases. *BMC Bioinformatics*, 7:531–549, 2006.

[25] W. Cai and G. Sakas. Data intermixing and multi-volume rendering. *Computer Graphics Forum*, 18(3):359–368, 1999.

[26] H. Carr, B. Duffy, and B. Denby. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1259–1265, 2006.

[27] H. Carr, J. Snoeyink, and M. van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry: Theory and Applications*, 43(1):42–58, 2010.

[28] M. Chen and H. Jänicke. An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6): 1206–1215, 2010.

[29] M. Chen and J. V. Tucker. Constructive volume geometry. *Computer Graphics Forum*, 19:281–293, 2000.

[30] M. Chicurel. Databasing the brain. *Nature*, 406:822–825, 2000.

[31] C. D. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, 2008. ISSN 1077-2626.

[32] C. D. Correa and K.-L. Ma. Visibility-driven transfer functions. In *Proceedings of PacificVis 2009*, pages 177–184, 2009.

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

[34] B. Csébfalvi, L. Mroz, H. Hauser, A. König, and M. E. Gröller. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum*, 20(3):452–460, 2001.

[35] G. Daniel and M. Chen. Video visualization. In *Proceedings of IEEE Visualization 2003*, pages 409–416, 2003.

[36] P. de Heras Ciechomski, R. Mange, and A. Peternier. Two-phased real-time rendering of large neuron databases. In *Proceedings of International Conference on Innovations in Information Technology 2008*, pages 712–716, 2008.

[37] W. de Leeuw, P. J. Verschure, and R. van Liere. Visualization and analysis of large data collections: A case study applied to confocal microscopy data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5): 1251–1258, 2006.

[38] M. Derthick, J. Kolojejchick, and S. F. Roth. An interactive visual query environment for exploring data. In *Proceedings of ACM UIST*, pages 189–198, 1997.

[39] B. J. Dickson. Wired for sex: the neurobiology of drosophila mating decisions. *Science*, 322(5903):904–909, 2008.

[40] H.-U. Dodt, U. Leischner, A. Schierloh, N. Jährling, C. P. Mauch, K. Deininger, J. M. Deussing, M. Eder, W. Zieglgänsberger, and K. Becker. Ultramicroscopy: Three-dimensional visualization of neuronal networks in the whole mouse brain. *Nature Methods*, 4:331–336, 2007.

[41] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *Proceedings of ACM SIGGRAPH 1988*, pages 65–74, 1988.

[42] N. Elmqvist, J. Stasko, and P. Tsigas. Datameadow: A visual canvas for analysis of large-scale multivariate data. In *Proceedings of VAST 2007*, pages 187–194, 2007.

[43] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Knowledge Discovery and Data Mining 1996*, pages 226–231, 1996.

[44] C. Eusemann, D. R. Holmes III, B. Schmidt, T. G. Flohr, R. Robb, C. McCollough, D. M. Hough, J. E. Huprich, M. Wittmer, H. Siddiki, and J. G. Fletcher. Dual energy CT: How to best blend both energies in one fused image? In *Proceedings of SPIE Medical Imaging 2008*, pages 1–8, 2008.

[45] A. C. Evans, S. Marrett, J. Torrescorzo, S. Ku, and L. Collins. MRI-PET correlation in three dimensions using a volume-of-interest (VOI) atlas. *Journal of Cerebral Blood Flow and Metabolism*, 11(2):A69–A78, 1991.

[46] J. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Proceedings of VAST 2006*, pages 167–174, 2006.

[47] S. Fang and R. Srinivasan. Volumetric-CSG - a model-based volume visualization approach. In *Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization*, pages 88–95, 1998.

[48] 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*, pages 281–288, 2007.

[49] M. Feixas, M. Sbert, and F. González. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception*, 6(1):1–23, 2009.

[50] M. Ferre, A. Puig, and D. Tost. A framework for fusion methods and rendering techniques of multimodal volume data: Research articles. *Computer Animation and Virtual Worlds*, 15:63–77, 2004.

[51] E. K. Fishman, D. R. Ney, D. G. Heath, F. M. Corl, K. M. Horton, and P. T. Johnson. Volume rendering versus maximum intensity projection in CT angiography: What works best, when, and why. *Radiographics*, 26 (3):905–922, 2006.

[52] J. Fredriksson. Design of an internet accessible visual human brain database system. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 469–474, 1999.

[53] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.

[54] I. Fujishiro, Y. Takeshima, T. Azuma, and S. Takahashi. Volume data mining using 3D field topology analysis. *IEEE Computer Graphics and Applications*, 20(5):46–51, 2000.

[55] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[56] W. J. Gilbert. A cube-filling hilbert curve. *The Mathematical Intelligencer*, 6(3):78, 1984.

[57] M. Haidacher, S. Bruckner, A. Kanitsar, and M. E. Gröller. Information-based transfer functions for multimodal visualization. In *Proceedings of Visual Computing for Biomedicine 2008*, pages 101–108, 2008.

[58] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller. Volume visualization based on statistical transfer-function spaces. In *Proceedings of IEEE Pacific Visualization 2010*, pages 17–24, 2010.

[59] M. Haidacher, S. Bruckner, and M. E. Gröller. Volume analysis using multimodal surface similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, 2011.

[60] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, 2002.

[61] A. Hanjalic and H. Zhang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8):1280–1289, 1999.

[62] A. J. Hanson and P. A. Heng. Four-dimensional views of 3D scalar fields. In *Proceedings of IEEE Visualization 1992*, pages 84–91, 1992.

[63] M. Harrower and C. A. Brewer. ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[64] H. Hauser, L. Mroz, G.-I. Bischi, and M. E. Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3): 242–252, 2001.

[65] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[66] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search techniques. In *Proceedings of IEEE Visualization 1996*, pages 227–234, 1996.

[67] W. Heidrich, M. McCool, and J. Stevens. Interactive maximum projection volume rendering. In *Proceedings of IEEE Visualization 1995*, pages 11–18, 1995.

[68] C. Heinzl, J. Kastner, and M. E. Gröller. Surface extraction from multi-material components for metrology using dual energy CT. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1520–1527, 2007.

[69] J. Hladůvka, A. König, and M. E. Gröller. Curvature-based transfer functions for direct volume rendering. In *Proceedings of the Spring Conference on Computer Graphics*, pages 58–65, 2000.

[70] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[71] H. Hong, J. Bae, H. Kye, and Y.-G. Shin. Efficient multimodality volume fusion using graphics hardware. In *Proceedings of the International Conference on Computational Science 2005*, pages 842–845, 2005.

[72] J. Hsieh. *Computed Tomography: Principles, Design, Artifacts and Recent Advances*. SPIE Press, 2003.

[73] X. Huang, N. Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1303–1318, 2006.

[74] H. V. Jagadish. Linear clustering of objects with multiple attributes. *Proceedings of ACM SIGMOD 1990*, 19(2):332–342, 1990.

[75] A. Jenett, J. E. Schindelin, and M. Heisenberg. The virtual insect brain protocol: Creating and comparing standardized neuroanatomy. *BMC Bioinformatics*, 7(1):544–555, 2006.

[76] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009.

[77] M. Jones, J. Baerentzen, and M. Šrámek. 3D distance fields: a survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):581–599, 2006.

[78] A. E. Kaufman, R. Yagel, R. Bakalash, and I. Spector. Volume visualization in cell biology. In *Proceedings of IEEE Visualization 1990*, pages 160–167, 1990.

[79] J. Kehrer, F. Ladstädter, P. Muigg, H. Doleisch, A. Steiner, and H. Hauser. Hypothesis generation in climate research with interactive visual data exploration. *IEEE Transactions on Visualization and Computer*, 14(6):1579–1586, 2008.

[80] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):100–107, 2002.

[81] M. Khoury and R. Wenger. On the fractal dimension of isosurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1198–1205, 2010.

[82] J. Kim, S. Eberl, and D. Feng. Visualizing dual-modality rendered volumes using a dual-lookup table transfer function. *Computing in Science and Engineering*, 9(1):20–25, 2007.

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

[84] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, 2003.

[85] J. Kniss, C. Hansen, M. Grenier, and T. Robinson. Volume rendering multivariate data to visualize meteorological simulations: a case study. In *Proceedings of VisSym 2002*, pages 189–195, 2002.

[86] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8:270–285, 2002.

[87] 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*, 9(2):150–162, 2003.

[88] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization 2003*, pages 497–504, 2003.

[89] J. Kniss, J. P. Schulze, U. Wössner, P. Winkler, U. Lang, and C. Hansen. Medical applications of multi-field volume rendering and VR techniques. In *Proceedings of VisSym 2004*, pages 249–254, 2004.

[90] S. H. Koslow and S. Subramaniam, editors. *Databasing the Brain: From Data to Knowledge (Neuroinformatics)*. Wiley, 2002.

[91] A. Kuß, S. Prohaska, B. Meyer, J. Rybak, and H.-C. Hege. Ontology-based visualization of hierarchical neuroanatomical structures. In *Proceedings of Visual Computing for Biomedicine 2008*, pages 177–184, 2008.

[92] T. Kvålseth. Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):517–519, 1987.

[93] C. Lau, L. Ng, C. Thompson, S. Pathak, L. Kuan, A. Jones, and M. Hawrylycz. Exploration and visualization of gene expression with neuroanatomy in the adult mouse brain. *BMC Bioinformatics*, 9(1):153–163, 2008.

[94] 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*, 15(6):1359–1366, 2009.

[95] D. N. Levin, X. P. Hu, K. K. Tan, S. Galhotra, C. A. Pelizzari, G. T. Chen, R. N. Beck, C. T. Chen, M. D. Cooper, and J. F. Mullan. The brain: integrated three-dimensional display of MR and PET images. *Radiology*, 172:783–789, 1989.

[96] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[97] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[98] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21: 163–169, 1987.

[99] T. Luft, C. Colditz, and O. Deussen. Image enhancement by unsharp masking the depth buffer. *ACM Transactions on Graphics*, 25(3):1206–1213, 2006.

[100] E. B. Lum and K.-L. Ma. Lighting transfer functions using gradient aligned sampling. In *Proceedings of IEEE Visualization 2004*, pages 289–296, 2004.

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

[102] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman. Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1648–1655, 2007.

[103] K.-L. Ma. Image graphs - a novel approach to visual data exploration. In *Proceedings of IEEE Visualization 1999*, pages 81–88, 1999.

[104] K.-L. Ma. Visualizing time-varying volume data. *Computing in Science & Engineering*, 5(2):34–42, 2003.

[105] R. Maciejewski, S. Rudolph, R. Hafen, A. Abusalah, M. Yakout, M. Ouzzani, W. S. Cleveland, S. J. Grannis, M. Wade, and D. S. Ebert. Understanding syndromic hotspots - a visual analytics approach. In *Proceedings of VAST 2008*, pages 35–42, 2008.

[106] M. M. Malik, T. Möller, and M. E. Gröller. Feature peeling. In *Proceedings of Graphics Interface 2007*, pages 273–280, 2007.

[107] S. Marchesin, J.-M. Dischler, and C. Mongenet. Feature enhancement using locally adaptive volume rendering. In *Proceedings of the International Symposium on Volume Graphics 2007*, pages 41–48, 2007.

[108] 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 ACM SIGGRAPH 1997*, pages 389–400, 1997.

[109] A. Martin and M. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of IEEE Visualization 1995*, pages 271–278, 1995.

[110] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[111] N. L. Max. Computer rendering of lobster neurons. In *Proceedings of ACM SIGGRAPH 1976*, pages 241–245, 1976.

[112] A. Maye, T. H. Wenckebach, and H.-C. Hege. Visualization, reconstruction, and integration of neuronal structures in digital brain atlases. *International Journal of Neuroscience*, 116(4):431–459, 2006.

[113] Z. Melek, D. Mayerich, C. Yuksel, and J. Keyser. Visualization of fibrous and thread-like data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1165–1172, 2006.

[114] M. Monks, B.M. Oh, and J. Dorsey. Audioptimization: Goal-based acoustic design. *IEEE Computer Graphics and Applications*, 20(3):76–91, 2000.

[115] B. Mora and D. S. Ebert. Instant volumetric understanding with order-independent volume rendering. *Computer Graphics Forum*, 23(9):489–497, 2004.

[116] L. Mroz, A. König, and M. E. Gröller. Maximum intensity projection at warp speed. *Computers & Graphics*, 24(3):343–352, 2000.

[117] P. Muigg, J. Kehrer, S. Oeltze, H. Piringer, H. Doleisch, B. Preim, and H. Hauser. A four-level focus+context approach to interactive visual analysis of temporal features in large scientific data. *Computer Graphics Forum*, 27(3):775–782, 2008.

[118] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang. On clustering and retrieval of video shots. In *Proceedings of Multimedia 2001*, pages 51–60, 2001.

[119] M. E. Noz, G. Q. Maguire, M. P. Zeleznik, E. L. Kramer, F. Mahmoud, and J. Crafoord. A versatile functional/anatomic image fusion method for volume data sets. *Journal of Medical Systems*, 25(5):297–307, 2001.

[120] S. Obayashi. *Evolutionary Multi-Objective Optimization and Visualization*, chapter 16, pages 175–185. Springer, 2005.

[121] S. Oeltze and B. Preim. Visualization of vasculature with convolution surfaces: method, validation and evaluation. *IEEE Transactions on Medical Imaging*, 24(4):540–548, 2005.

[122] S. R. Olsen and R. I. Wilson. Cracking neural circuits in a tiny brain: new approaches for understanding the neural circuitry of drosophila. *Trends in Neurosciences*, 31(10):512–520, 2008.

[123] V. Pekar, R. Wiemker, and D. Hempel. Fast detection of meaningful isosurfaces for volume data visualization. In *Proceedings of IEEE Visualization 2001*, pages 223–230, 2001.

[124] W. Pereanu and V. Hartenstein. Neural lineages of the drosophila brain: A three-dimensional digital atlas of the pattern of lineage location and projection at the late larval stage. *The Journal of Neuroscience*, 26(20): 5534–5553, 2006.

[125] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. S. Avila, K. M. Raghu, R. Machiraju, and J. Lee. The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–22, 2001.

[126] F. Pinto and C. Freitas. Design of multi-dimensional transfer functions using dimensional reduction. In *Proceedings of EuroVis 2008*, pages 130–137, 2007.

[127] W. A. Press, B. A. Olshausen, and D. C. van Essen. A graphical anatomical database of neural connectivity. *Philosophical Transactions of the Royal Society*, 356:1147–1157, 2001.

[128] A. Rajwade, A. Banerjee, and A. Rangarajan. Probability density estimation using isocontours and isosurfaces: applications to information-theoretic image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):475–491, 2009.

[129] P. Rautek, S. Bruckner, and M. E. Gröller. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1336–1343, 2007.

[130] P. Rautek, S. Bruckner, and M. E. Gröller. Interaction-dependent semantics for illustrative volume rendering. *Computer Graphics Forum*, 27(3): 847–854, 2008.

[131] C. Rezk-Salama and A. Kolb. Opacity peeling for direct volume rendering. *Computer Graphics Forum*, 25(3):597–606, 2006.

[132] P. Rheingans and D. S. Ebert. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):253–264, 2001.

[133] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the International Conference*

*on Coordinated & Multiple Views in Exploratory Visualization 2007*, pages 61–71, 2007.

[134] S. Roettger, M. Bauer, and M. Stamminger. Spatialized transfer functions. In *Proceedings of EuroVis 2005*, pages 271–278, 2005.

[135] T. Rohlfing and C. R. Maurer, Jr. Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. *IEEE Transactions on Information Technology in Biomedicine*, 7(1):16–25, 2003.

[136] G. Sakas, M. G. Vicker, and P. J. Plath. Visualization of laser confocal microscopy datasets. In *Proceedings of IEEE Visualization 1996*, pages 375–379, 1996.

[137] Y. Sato, N. Shiraga, S. Nakajima, S. Tamura, and R. Kikinis. Local maximum intensity projection (LMIP): A new rendering method for vascular visualization. *Journal of Computer Assisted Tomography*, 22(6): 912–917, 1998.

[138] Y. Sato, C. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):160–180, 2000.

[139] C. E. Scheidegger, J. M. Schreiner, B. Duffy, H. Carr, and C. T. Silva. Revisiting histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1659–1666, 2008.

[140] S. Schmitt, J. F. Evers, C. Duch, M. Scholz, and K. Obermayer. New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. *NeuroImage*, 23(4):1283–1298, 2004.

[141] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum*, 28:855–862, 2009.

[142] P. Šereda, A. Vilanova, and F. A. Gerritsen. Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In *Proceedings of EuroVis 2006*, pages 243–350, 2006.

[143] R. Shams and N. Barnes. Speeding up mutual information computation using NVIDIA CUDA hardware. In *Proceedings of Digital Image Computing: Techniques and Applications 2007*, pages 555–560, 2007.

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

[145] A. Sherbondy, D. Akers, R. Mackenzie, R. Dougherty, and B. Wandell. Exploring connectivity of the brain's white matter with dynamic queries. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):419–430, 2005.

[146] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, 1997.

[147] R. Smith, R. Pawlicki, I. Kókai, J. Finger, and T. Vetter. Navigating in a shape space of registered models. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1552–1559, 2007. ISSN 1077-2626.

[148] M. Straka, M. Cervenansky, A. La Cruz, A. Köchl, M. Šrámek, M. E. Gröller, and D. Fleischmann. The VesselGlyph: Focus & context visualization in CT-angiography. In *Proceedings of IEEE Visualization 2004*, pages 385–392, 2004.

[149] S. Takahashi, Y. Takeshima, I. Fujishiro, and G. M. Nielson. Emphasizing isosurface embeddings in direct volume rendering. In G. P. Bonneau, T. Ertl, and G. M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 185–206. Springer, 2006.

[150] S. Tenginakai, J. Lee, and R. Machiraju. Salient iso-surface detection with model-independent statistical signatures. In *Proceedings of IEEE Visualization 2001*, pages 231–238, 2001.

[151] A. Treuille, A. Lewis, and Z. Popović. Model reduction for real-time fluids. *ACM Transactions on Graphics*, 25(3):826–834, 2006.

[152] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(1):1–37, 2007.

[153] J. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[154] F.-Y. Tzeng and K.-L. Ma. A cluster-space visual interface for arbitrary dimensional classification of volume data. In *Proceedings of VisSym 2004*, pages 17–24, 2004.

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

[156] J. J. van Wijk and R. van Liere. Hyperslice: visualization of scalar functions of many variables. In *Proceedings of IEEE Visualization 1993*, pages 119–125, 1993.

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

[158] I. Viola, M. Feixas, M. Sbert, and M. E. Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12 (5):933–940, 2006.

[159] P. Šereda, A. Vilanova Bartrolí, I. W. O. Serlie, and F. A. Gerritsen. Visualization of boundaries in volumetric data sets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):208–218, 2006.

[160] V. Šoltészová, D. Patel, S. Bruckner, and I. Viola. A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum*, 29(3):883–891, 2010.

[161] J. W. Wallis, T. R. Miller, C. A.Lerner, and E. C. Kleerup. Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging*, 8(4):297–303, 1989.

[162] T. Walter, D. W. Shattuck, R. Baldock, M. E. Bastin, A. E. Carpenter, S. Duce, J. Ellenberg, A. Fraser, N. Hamilton, S. Pieper, M. A. Ragan, J. E. Schneider, P. Tomancak, and J-K. Hériché. Visualization of image data from cells to organisms. *Nature Methods*, 7(3S):S26–S41, 2010.

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

[164] L. Wang, Y. Zhang, and J. Feng. On the Euclidean distance of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1334–1339, 2005.

[165] S. W. Wang and A. E. Kaufman. Volume-sampled 3D modeling. *IEEE Computer Graphics and Applications*, 14(5):26–32, 1994.

[166] Z. Wang, D. Ziou, C. Armenakis, D. Li, and Q. Li. A comparative analysis of image fusion methods. *IEEE Transactions on Geoscience and Remote Sensing*, 43:1391–1402, 2005.

[167] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical image analysis*, 1(1):35–51, 1996.

[168] M. Wohlfart and H. Hauser. Story telling for presentation in volume visualization. In *Proceedings of EuroVis 2007*, pages 91–98, 2007.

[169] J. Woodring and H.-W. Shen. Chronovolumes: a direct rendering technique for visualizing time-varying data. In *Proceedings of Volume Graphics 2003*, pages 27–34, 2003.

[170] J. Woodring and H.-W. Shen. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.

[171] J. Woodring and H.-W. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–137, 2009.

[172] J. Woodring, C. Wang, and H.-W. Shen. High dimensional direct rendering of time-varying volumetric data. In *Proceedings of IEEE Visualization 2003*, pages 417–424, 2003.

[173] E. Wu, Y. Liu, and X. Liu. An improved study of real-time fluid simulation on GPU. *Computer Animation and Virtual Worlds*, 15(34):139–146, 2004.

[174] L. Xu, T.-Y. Lee, and H.-W. Shen. An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1216–1224, 2010.

[175] J. Yang, M. O. Ward, E. A. Rundensteiner, and S. Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of VisSym 2003*, pages 19–28, 2003.

[176] Y. Y. Yao. Information-theoretic measures for knowledge discovery and data mining. In Karmeshu, editor, *Entropy Measures, Maximum Entropy Principle and Emerging Application*, pages 115–136. Springer, 2003.

[177] Z. Ying, R. Naidu, K. Guilbert, D. Schafer, and C. R. Crawford. Dual energy volumetric x-ray tomographic sensor for luggage screening. In *Proceedings of the IEEE Sensors Applications Symposium 2007*, pages 1–6, 2007.

[178] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[179] K. J. Zuiderveld and M. A. Viergever. Multi-modal volume visualization using object-oriented methods. In *Proceedings of the IEEE Symposium on Volume Visualization 1994*, pages 59–66, 1994.