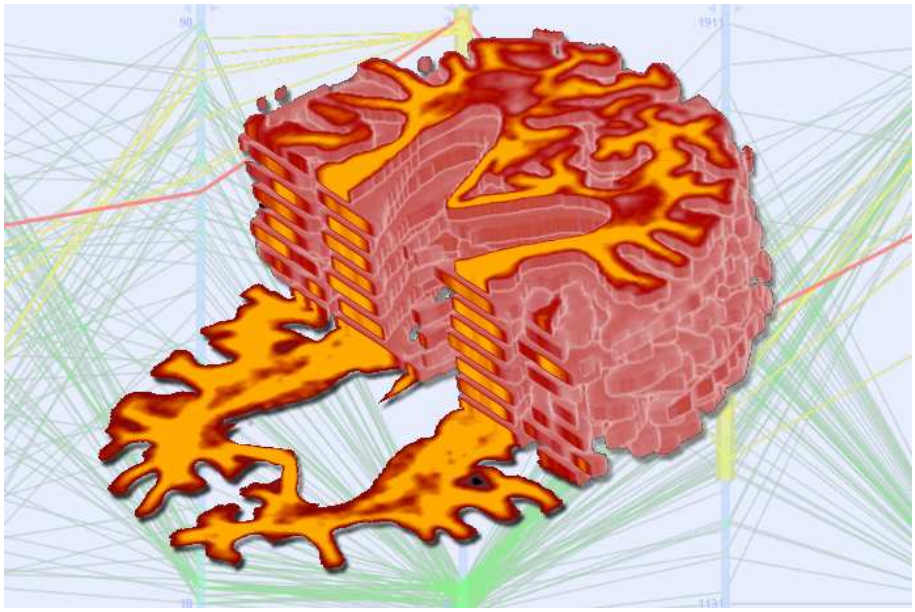
Multi-Volume Visualization and Exploration

Stian H. Eikeland
stian.eikeland@gmail.com

Aug 2010

Visualization Group
Department of Informatics
University of Bergen

Multi-Volume Visualization and Exploration



Stian H. Eikeland
stian.eikeland@gmail.com

Aug 2010

Supervised by Helwig Hauser

Visualization Group
Department of Informatics
University of Bergen

http://www.ii.uib.no/vis/teaching/thesis/2010-eikeland/

Abstract

The amount of available volumetric data for scientific data visualization has increased rapidly of the years, calling for new methods to visualize large multi-volume databases. A method for exploring and visualizing multi-volume datasets is presented in this thesis. The work consists of a linked three component setup featuring a metadata overview visualizing, a slice stack visualizing and a focus volume visualization. The system relies on externally provided and internally derived metadata to provide a good overview of multi-volume datasets using a parallel coordinates view. Through brushing and focus+context visualization it is possible to select interesting subsets of the data for further exploration using two linked direct volume rendering views, a slice stack visualizing corresponding slices from selected volumes and a focus volume providing a good detailed view of any selected volume. Through three-way interaction it is possible to efficiently explore and visualize large multi-volume datasets using this setup.

Acknowledgements

I would like to begin this thesis by thanking everyone who has helped to make it possible and who supported me along the way. I would like to thank the creator of the Volumeshop framework which my project is based on, namely Stefan Bruckner. I would also like to thank Arvid Lundervold for his help with the practical medical usecases. Thanks also go out to all of the people in the Visualization group here at the University of Bergen who have given me great tips and ideas for this project, and of course; my supervisor, Helwig Hauser, for his continued support and help through out these last two years. Last but not least, I would like to thank friends, family and my workplace for their support.

I've used data from the OASIS-project, which is supported by the following grants: P50 AG05681, P01 AG03991, R01 AG021910, P20 MH071616, U24 RR021382. I've also used muParser, a fast math parser library by Ingo Berg.

Contents

1 Introduction

The only reason people want to be masters of the future is to change the past.

- Milan Kundera

Barely forty years ago, Sir Godfrey Hounsfield sat in his lab and tinkered on a machine, a machine that was going to become one of major cornerstones of which diagnostic medicine is based on today and that also earned him one of the 1979 Nobel Prices [1]. By shooting a rotating x-ray beam through an object followed by hours of calculations on a large computer he was able to produce a sliced image of his object like figure 1.1(a). For the first time it was possible to not only look through the body as with X-ray images, but to look at it from the inside, making any virtual cut of the body as needed. The machine he had invented was the *Computed Tomography (CT)* scanner, which was commercialized only a few years later. Hounsfield's machine became incredibly popular and is today one of the most important tools for medical imaging, in 2007 over 70 million CT scans were performed in the United States alone [2].

In the years following the introduction of the CT scanner, several other imaging technologies became available. By using other methods than X-rays to peer into the body, such as using huge magnets to see how the body's water molecules align, it became possible to get different types of images. Some can clearly show structures such as bone (CT), others are able to differentiate between different types of tissue (*Magnetic Resonance Imaging (MRI)*), some can even monitor blood flow and tell us which parts of the brain that currently are in use (*Functional Magnetic Resonance Imaging (fMRI)*). Today a wide spectrum of different techniques are available, able to see many different aspects of the human body — from which cells are the most active (*Positron Emission Tomography (PET)*) to how the brain is wired (*Diffusion Tensor Imaging (DTI)*).

(a) First CT scan of a brain, as presented by Sir Godfrey Hounsfield in the 1979 Nobel Price lecture [3].

(b) An example of a modern day volume rendering of a CT scan, public domain image by Mugab[2].

Figure 1.1: The first CT scan of a human brain and a modern day volume rendering.

By capturing images at regular distances and stacking them it is possible to represent 3-dimensional volumes. In the beginning they were used and handled as sequences of images, but with advances in computer hardware and software it became possible to render these image stacks as 3D volumes. This field of research, Volume Rendering, was kickstarted in the late 1980s by the introduction of algorithms that could extract high resolution surfaces from volumetric data [4]. Just one year later, it was made possible to render 3D images from any angle without any intermediate steps by simply integrating over a stack of images [5]. Since then a wide array of tools and techniques have been developed to render, display, and explore volumetric datasets for many purposes. Spearheaded by the gaming industry, consumer computers have become so fast and powerful that even the computers in our homes can interactively display high resolution volumetric datasets. Rendering volumetric data has also moved outside of the medical realm, similar techniques and equipment is used in everything from quality control of industrial products to searching volumetric seismic datasets for traces of oil and gas, or to look out into the universe of captured astronomical data.

How to render and explore a single volumetric dataset interactively is by

[2]http://commons.wikimedia.org/wiki/File:High_Definition_Volume_Rendering.JPG

many considered a solved problem today, and it can give great insight into a single dataset. But given the incredible amount of volumetric dataset available, how can we explore a large population of these volumes, not just looking at them one at the time? Can we look at two medical scans simultaneous? Yes, that is — in a sense — very easy, for example by rendering them next to each other or even overlapping with different rendering styles. How about rendering five datasets next to each other, is it still possible make sense of it all? Probably, but now we're racing head on towards a wall of information overload — and five isn't even considered many datasets by today's standards, some of the medical databases used by the research community are much larger.

1.1 Open medical image repositories

There exist many large open repositories of medical imaging data being used by the research community today, some are made of simulated images from phantoms and MRI-simulators (BrainWeb [6]), while others are big collections of real medical scans. The purpose of these databases can be to support medical research of diseases such as Alzheimer's and Huntington's, or are even created for purposes like providing a gold standard for evaluating segmentation and image registration algorithms. Two of the most commonly used image repositories that are available to the general scientific community are the *Open Access Series of Imaging Studies (OASIS)* [7] project and also the *Alzheimer's Disease Neuroimaging Initiative's (ADNI)* [8] repository.

1.1.1 The Alzheimer's Disease Neuroimaging Initiative

The Alzheimer's Disease Neuroimaging Initiative is a large five year research project funded by the National Institutes of Health and partners [8]. ADNI is a study of cognitive aging, mild cognitive impairment (MCI) and Alzheimer's disease (AD) with the goal of gathering data that can be used to prevent neural degeneration in the elderly. The ADNI project has gathered MRI-images, PET-images, and Cerebral Spinal Fluid (CFS) samples from 400 volunteers with MCI, 200 volunteers with AD, and 200 healthy elderly as a control group. The ADNI image data is available both in raw format, and as several different post-processed varieties (sharpening, geometry distortion correction, etc). The image data is open to the general scientific community through the submission of an online application, though with several restrictions on it's use. With it's large collection of over 800 subjects, the ADNI

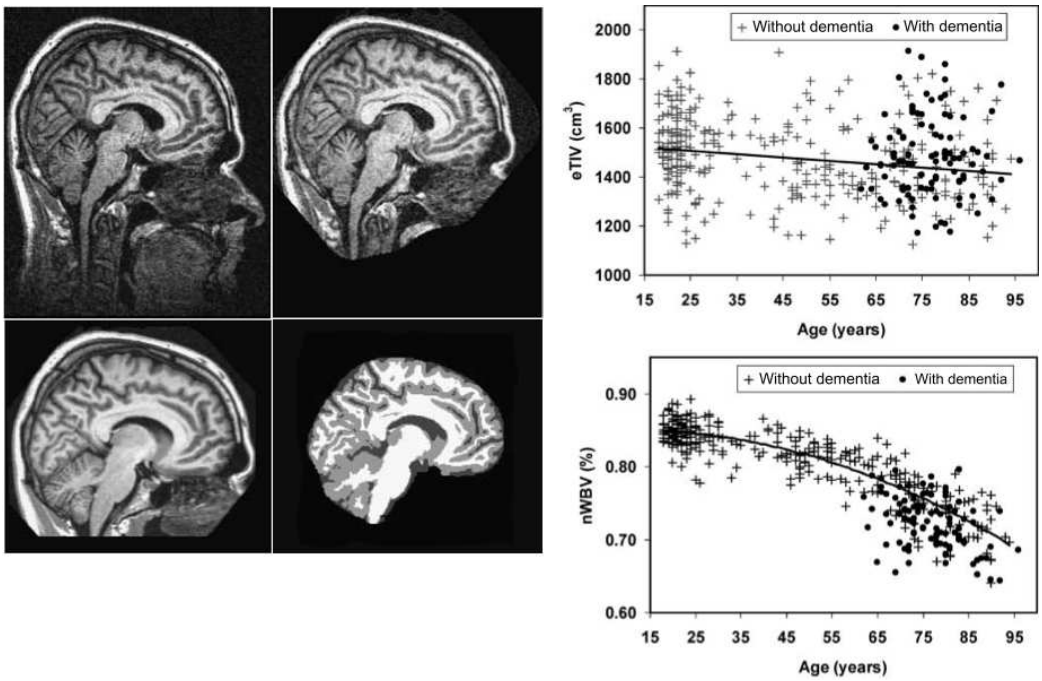


Figure 1.2: Data from the *OASIS* project. Images (left): Showing the original scan (top left), the defaced/anonymized scan (top right), the atlas registered scan (bottom left) and whitematter / graymatter segmentation (bottom right). Scatterplot (right): Showing the distribution of *Estimated Total Inter-cranial Volume (eTIV)* vs. age and distribution of *Normalized Whole Brain Volume (nWBV)* vs. age.

database is one of the largest open databases of it's kind today and is used in many studies of dementia.

1.1.2 The Open Access Series of Imaging Studies

The Open Access Series of Imaging Studies (OASIS) [7] is another large project with the goal of providing the scientific community with freely available MRI datasets, made available to facilitate discoveries in basic and clinical neuroscience. The OASIS-project consists of two larger studies with publicly available imaging data, *"Cross-sectional MRI Data in Young, Middle Aged, Non-demented, and Demented Older Adults"* [7] and *"Longitudinal MRI Data in Non-demented and Demented Older Adults"* [9]. The last one consists of imaging data of approximately 150 subjects between the ages of 60 and 96 — both men and women. Multiple MRI sessions have been conducted with

every subject, separated by at least one year. 64 of the subjects were clinically classified as demented at the time of their first session, including 51 individuals with a mild to moderate Alzheimer's disease. During the study, 14 subjects previously classified as non-demented, developed dementia. The first study is similar, but larger, containing imaging data of 416 subjects between the ages of 18 and 96. Here however, only one scanning session per individual is available. 100 of the subjects over the age of 60 in this study are clinically diagnosed with mild to moderate Alzheimer's disease.

The scans have been carefully screened and gone through several steps of quality control before being anonymized and post-processed. Some of the steps in this process include noise-filtering, defacing, atlas registration, and whitematter / greymatter segmentation, see figure 1.2. It is available for unrestricted use both on DVD, and for viewing and downloading from their webpage[3]. The total size of the datasets (including derived data) ranges in the size of 135 Gigabytes (compressed). Each dataset also contains a range of relevant metadata, such as the subject's age, sex, socioeconomic status, education level, *Clinical Dementia Rating (CDR)*, *Estimated Total Intercranial Volume (eTIV)*, *Normalized Whole Brain Volume (nWBV)* and the results of a *Mini-Mental State Examination (MMSE)*. The volumes are all available in the Analyze 7.5 format, including the three to four individual scans, an averaged atlas registered dataset (Talairach), a masked version of the atlas registered set, and a whitematter / greymatter segmentation mask. The datasets have also been segmented and labeled using the automated segmentation tool Freesurfer [10].

1.2 Exploring large multi-volume datasets

When researchers work with large datasets like the ones mentioned above, they typically start by forming a hypothesis. They derive the needed data from the volumetric images to test their hypothesis, and then use this derived metadata in statistical software such as SPSS[4], SAS[5], Matlab[6], or the open source project R[7]. Correlations between attributes or interesting outliers can be found by performing different types of statistical analysis. This is a method that scales quite well – many of these statistical analysis tools

[3]The Oasis Project — http://www.oasis-brains.org/
[4]IBM SPSS – http://www.spss.com/
[5]Statistical Analysis Software – http://www.sas.com/
[6]Matlab by MathWorks – http://www.mathworks.com/
[7]The R Project – http://www.r-project.org/

can deal with large amounts of data. While this is very good for hypothesis testing, it isn't necessarily that good for exploration. The user ends up being decoupled from the original volumetric data when analyzing only derived metadata. This means that every time a need for new derived data emerges or when it becomes necessary to interact directly with the volumetric data, the user is forced to break break out of the analyzation loop. For interactive exploration it is better with a tightly linked system, featuring interactive data derivation, visualization and interaction of both derived data and multiple volumes in linked views.

There is no question that dealing with large amounts of volumetric data can be hard. Not only does one need to handle large amounts of data (possibly many Gigabytes of volumetric data), but it is also a challenge of handling increased dimensionality. Visualizing a single 3D-volume is considered computationally easy and can be done without overwhelming the user with information, but with additional dimensions such as multiple subjects or timesteps it can quickly become a complicated task because of the large amounts of information that now needs to be presented to the user. It is a challenge of obfuscation, large data sizes, and dimensionality.

The questions this thesis deal with is how to interactively explore large databases of volumetric data like the databases presented above (ADNI and OASIS). What methods can be used to get an overview of large amounts of volumetric data from multiple individuals or timesteps without succumbing to information overload? What methods can be used to break down the dimensionality of such large datasets with the purpose of easy interaction and exploration of volumetric data? It is easy to imagine that presenting hundreds of volumes in a single view is not a good way to provide an overview, and it is quite obvious that there is a need for abstraction. Luckily, datasets such as the ones from the ADNI and the OASIS project are usually accompanied by a wide range of metadata, extra pieces of information telling us something about the data. In the domain of medical imaging this would typically mean external attributes like age, sex, or results from clinical tests, but it can also mean values it is possible to derive from the volumetric data itself; the size, shape or thickness of a segment, the frequency-distribution in a particular slice and similar. The methods presented in this thesis utilizes this derived data to provide a good overview – an overview capable of showing outliers and trends, to interactively derive new metadata, and to drill down on details or groups of volumes that seems worthy of further exploration. Methods from a toolbox of dimensionality reduction techniques are used for creating multiple linked views that together visualize multi-individual volu-

metric data efficiently and helps with interaction. These methods allow an user to explore and interact with the data – all the way from an overview of multiple volumes, to derive new metadata as needed, to drill down into interesting groups of volumes, to compare and look at the detailed view of a single volume, all as illustrated in this thesis.

1.3 Overview of the thesis

The main goal of this thesis was to create an interactive system enabling users to get a good overview of large sets of volumetric data using metadata visualization, and to drill down and interact with the data through the concepts of *slice stacks* and *focus volumes*. It starts out by discussing techniques for visualizing multiple volumes in chapter 2, followed by a description of this thesis' proposed solution in chapter 3. These techniques have been implemented in an already existing framework for volumetric rendering called Volumeshop, the details of this integration are discussed in chapter 5 and demonstrated in the context of the already introduced OASIS database in chapter 4. In the end there is a short summary and a conclusion in chapter 6.

2
State of the art

The mind, once expanded to the dimensions of larger ideas, never returns to its original size.

- Oliver Wendell Holmes

The task of visualizing multiple volumes is something that several different branches of the scientific visualization community have worked on. Objects can be captured in multiple modalities and rendered as one image that can provide more or different information than what is possible to get by only looking at a single modality. An example is astronomical data, often gathered using different imaging technologies – it is possible to capture images of the visible light spectrum using one camera and images of the X-ray or infrared spectrum using other cameras. All of these images are telling us something about the same subject, and being able to look at more than one at the same time can often have a synergetic effect. Another example is the acquisition of multiple medical scans of a patient to track the progression of a disease. In this case we have multiple volumes of the same subject captured at different time intervals, data which can be used to discover changes that have occurred and when this happened.

Scenarios like these can be visualized with techniques for intra-subject visualization which is discussed in section 2.1. In section 2.2, we move outside of the single subject case to see what is possible when dealing with datasets of the same modality captured from multiple subjects – inter-subject visualization. Realizing that these examples are all specific cases of higher-dimensional datasets (meaning higher than the usual three spatial dimensions) – the volumes themselves represent 3D space, but when including multiple time-steps, modalities or individuals, we can consider such scenarios as four or higher-dimensional. The chapter therefore includes methods for visualizing and exploring higher-dimensional data in general in section 2.3.

2.1 Intra-subject multi-volume visualization

The techniques for intra-subject multi-volume visualization can roughly be divided into two main groups, one dealing with visualization of multiple modalities and one dealing with time-varying data. Common for both is the question of how to handle multiple volumes captured from the same subject.

2.1.1 Visualizing multimodal and multi-variate data

In multimodal and multi-variate visualization we deal with the task of creating a single image based on several different data sources, in the medical domain this usually means volumes captured from different types of scanning technologies. Each technology has it's own characteristics; a set of strengths and weaknesses. For example, a PET scan can provide metabolic information, but will only provide fairly limited (in terms of physiology) anatomical information. PET scans are therefore often accompanied either by a CT or MRI scan – a combination that allow seeing both what is going on and where this is happening.

Before any combined visualization can be done, the different data sources need to be in proper alignment or matched. Having multiple data sources containing complementary information does not help if corresponding voxels contains data from different physical locations – the volumetric data needs to be transformed into a common coordinate system. This can be accomplished using *Image Registration*, an iterative process of mapping voxels from one volume to an other, applying a correspondence metric, and adjusting the voxel mapping until converging on a maximum correspondence measurement [11]. Measuring the correspondence of multimodal data is often achieved using a concept from information theory called *Mutual Information* (MI) – a measure of mutual dependence of two variables. MI has successfully been used on a wide range of different multimodal data [11]. This can be a slow process and is usually done as a preprocessing step.

We can make use of multi-modal and multi-variate data at different stages in the rendering pipeline. Fuchs and Hauser [12] have presented the following six possible pipelines for rendering multi-variate scientific data: *standard*, *feature based*, *interactive visual analysis*, *data intermixing*, *layering* and *"goal"* (goal here meaning the ultimate goal of integrated visualizations). The different groups of techniques use, abstract and combine the original data at different levels of the visualization pipeline (see figure 2.1).

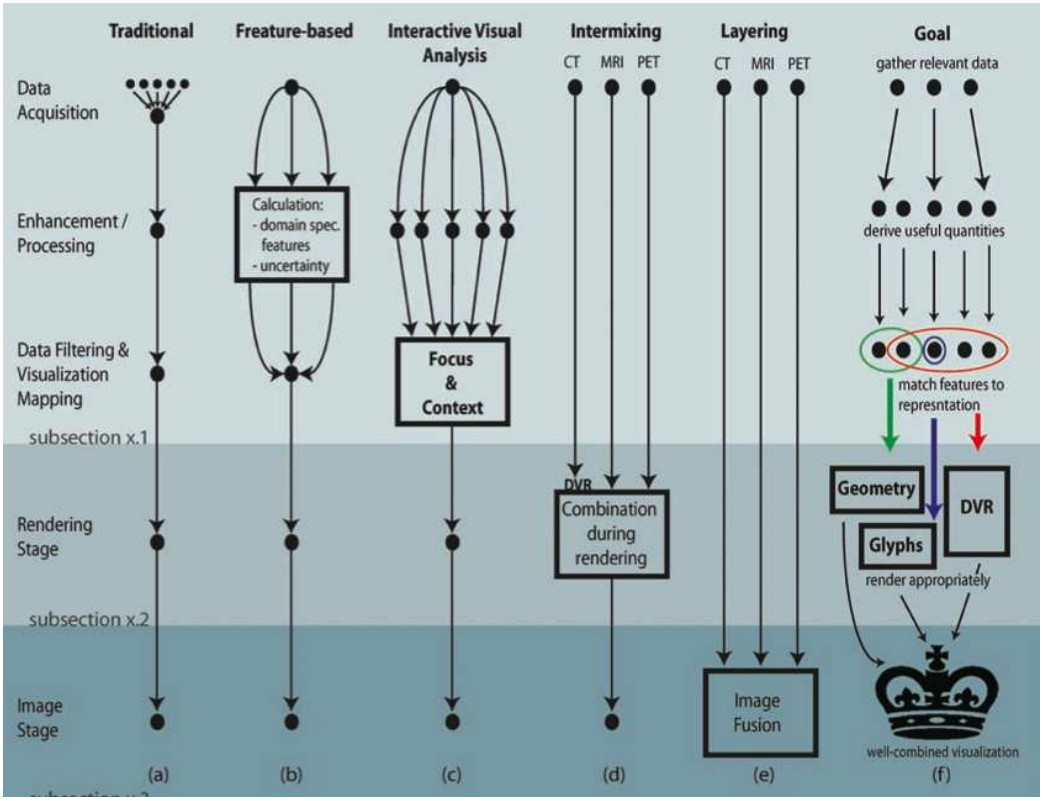


Figure 2.1: The different visualization pipelines for multi-variate scientific data visualization as described by Fuchs and Hauser [12].

The standard pipeline (figure 2.1a): Here the data sources are combined in a preprocessing or filtering step directly after data acquisition. A set of filters are defined to select what data source to use at each spatial position, thus generating a single volume containing contributions from multiple sources. This volume can then be rendered like any other normal volume, requiring no changes to existing single-modality volume renderers. The approach can for example be used to combine co-registered CT and MRI scans; we know that voxels of high intensity values in medical CT scan usually correspond to bone structures, so by creating a filter that selects high-intensity CT voxels and uses MRI for all others it is possible to create a hybrid CT/MRI volume, showing anatomical structures of both bone and tissue. This method is however inherently inflexible, as it needs to go all the way back to the top of the pipeline and create a new volume every time a change needs to be made.

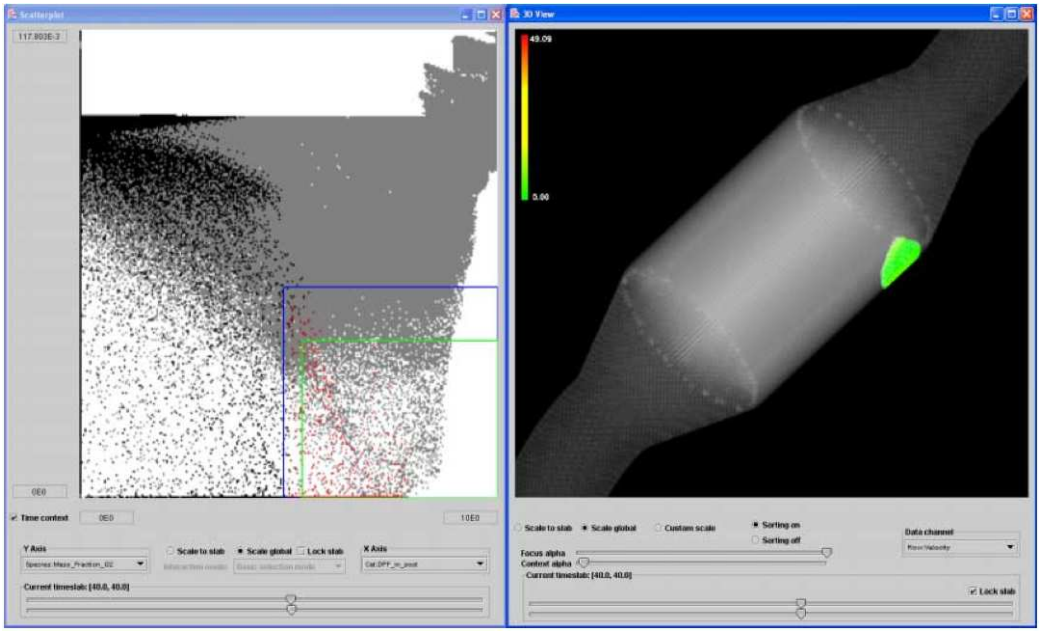


Figure 2.2: Doleisch et al. track down incomplete soot oxidation in a Diesel particulate filter using brushing and two linked views in SimVis [13].

The feature-based pipeline (figure 2.1b): By taking a step up the abstraction ladder, instead of working on simple intensity values one can concentrate on features and segments. By using feature extracting techniques and segmentation tools it is possible to label voxels in the data. For example in a medical volume, a subset of the voxels may be labeled to represent the liver, and another subset labeled as belonging to the heart. Having this higher level understanding about the data can help select the most appropriate modality for a given structure or feature in the data, and it can also help with tasks such as transfer function selection. Techniques from this group are often similar to techniques from the standard pipeline seen above, the difference being that knowledge about segments and features is available to the rendering algorithm.

Interactive visual analysis (figure 2.1c): Interactive visual analysis tools allow for powerful interaction and exploration of scientific data through a mixture of techniques from both the field of *Scientific Visualization* and *Information Visualization*. By using linking, brushing [14] and selective rendering in multiple views the information visualization concept of focus+context visualization is brought together with scientific

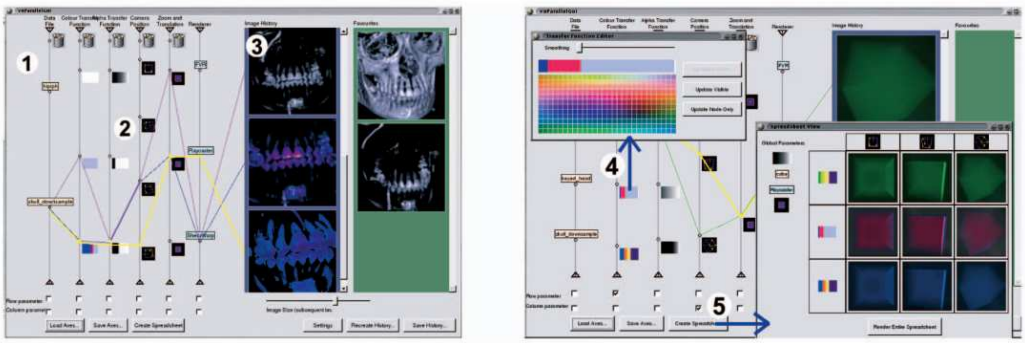
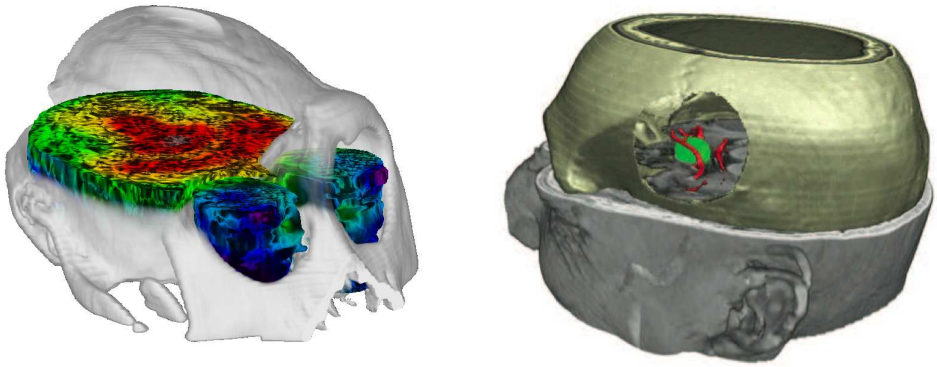


Figure 2.3: Tory et al. [18] using a parallel coordinates system as a means of exploring a volumetric dataset instead of traditional transfer function graph widgets.

volume rendering, allowing the exploration of volumetric data [15] [16]. Tools like these are often used to explore large simulated dataset, often containing multiple attributes per spatial position, for example the results from a computational fluid dynamics simulation. The concept of interactively brushing values directly in selected views (scatterplots, parallel coordinates, volume renderings, etc.) to make selections of interesting data and then quickly seeing the distribution of the selected data (in the context of the rest of the data) in other linked views allows for very effective exploration of multi-variate volumetric data. An example of such a tool is the SimVis system developed by Doleisch et al. which has been successfully used to explore and visualize multi-variate volumetric data ranging from Diesel exhaust systems [13] to hurricane Isabel [17]. In figure 2.2 SimVis is show visualizing incomplete soot oxidation due to the lack of oxygen in a Diesel particulate filter. The authors have accomplished this by making an interactive selection of values indicating an incomplete oxidation in the right view, due to the linked views the location of this phenomenon is also shown in the volume rendering in the left view.

One interesting concept from information visualization, that also often is utilized in scientific data visualization, is the parallel coordinates system[19]. It is a simple setup where multiple parallel axes (representing variables or dimensions) are used to indicate the data value of a series of items plotted as polylines – the position a line intersect the axis visualizes the value. It's an old concept that has proven to be very



(a) Rendering of a monkey's head from CT, MRI, and PET data. The skull is rendered from CT data, the brain is colored based on PET activity and shaded based on MRI data to reveal structures in the tissue. The visualization was made using the Volumeshop framework and a custom renderer.

(b) Multi-volume rendering of segmented data for preoperative planning (green: tumor – MRI, red: vessels – MRA, brown: skull – CT) by Beyer et al. [22] [23]

Figure 2.4: Two examples of data intermixing multimodal volume renderings.

useful for plotting and exploring high- and hyper-dimensional data [20], and has even been used for purposes such as providing a interface for exploratory volume visualization in the works of Tory et al. [18]. Tory et al. maps each parameter of the visualization is to axes in a parallel coordinates system, shown in figure 2.3. Parallel coordinates has also successfully been used for visualizing multi-dimensional geometry in a 4D (space+time) visualization of airplane flight paths [21], clearly showing intersections that could be hard to spot in normal 3D plots.

Data intermixing (figure 2.1d): Combining multiple data sources in the rendering stage is called data intermixing or fusion, it's a technique often used in combination with either segmentation data, advanced transfer functions or brushing to select which data source to emphasize in a given region of the data. For example, by utilizing segmentation data it is possible to select a modality or combination of modalities that makes most sense for any given structure, such as using CT for bones and MRI for tissue. In figure 2.4 two examples of techniques that use segmentation data is shown, figure 2.4(a) is a rendering of a monkey head using the Volumeshop [24] framework and a custom renderer that selects it's data source based on a segmentation mask. In this example, the segmentation mask marks an area of the volume

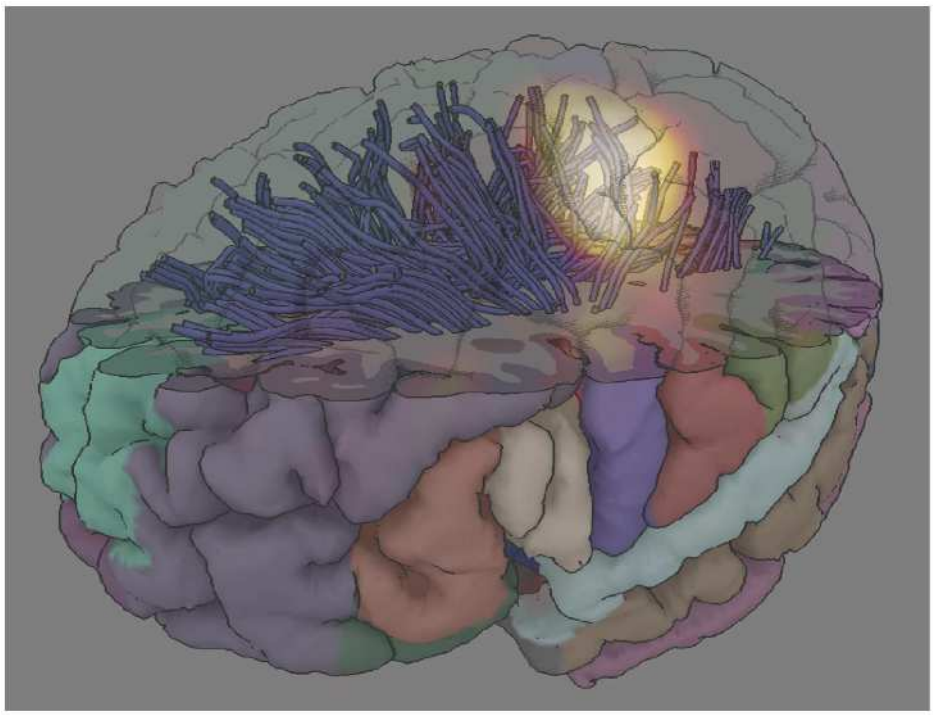


Figure 2.5: Born et al. visualize anatomical, functional and reconstructed fiber tracts data using illustrative styles [25].

which represents the brain – the area outside of the mask can be rendered using a different style than to the area covered by the mask. The brain area is colored based on PET activation data and shaded based on MRI data to show the tissue structures, the area outside (bones + skin) is rendered from CT data colored by a normal transfer function. A similar example by Beyer et al. can be seen in figure 2.4(b) where MRI, MRA and CT data have been combined [22].

Layering (figure 2.1e): Layering or pixel intermixing are image space techniques that render modalities separately before combing the results pixel by pixel based on color and opacity values. The clear separation of the rendering pipelines for each modality often makes these techniques simple to implement. However, there is a major drawback when combining multiple rendering results in image space, i.e., the lack of depth cueing [26]. It can be very hard to understand from the combined images which structure from one modality is in front of another.

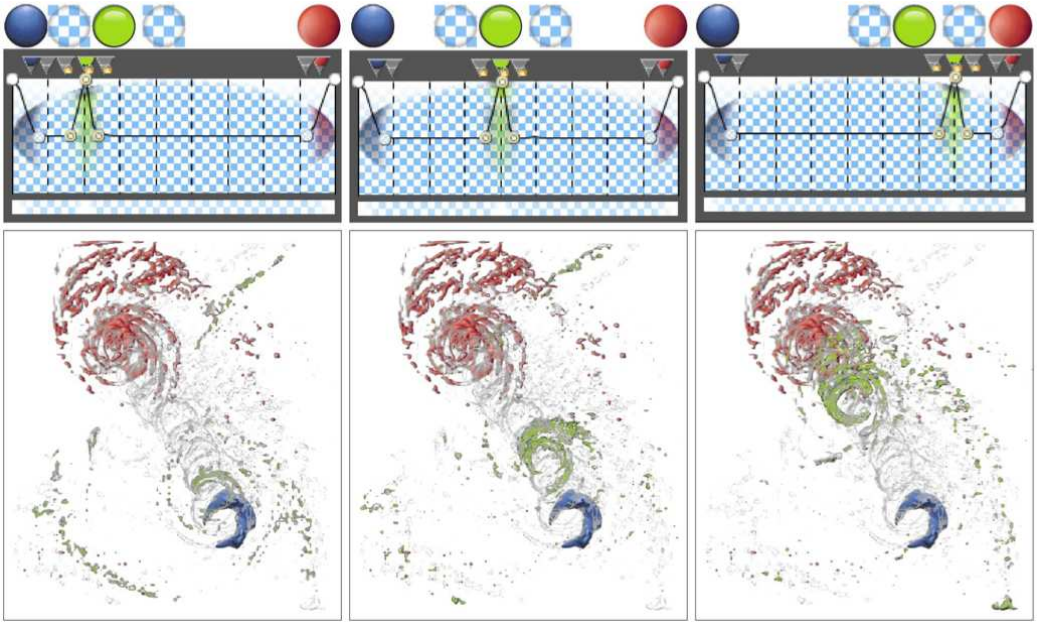


Figure 2.6: Temporal transfer functions by Balabanian et al. allow the rendering multiple timesteps of hurricane Isabel in a single image [27].

This challenge has been partially solved by storing Z-buffer depth values per pixel, making it possible to combine the rendered images in the right order, but this is still not a good solution and these techniques are therefor not that much used in general, at least compared to data intermixing techniques.

Integrated (figure 2.1f): Integrated are higher level visualization techniques that utilizes derived data and detection of features of interest together with appropriate rendering algorithms [12]. An example can be seen in figure 2.5 where Born et al. has developed a technique for visualization of anatomical, functional and reconstructed fiber tracts based on DTI information.

2.1.2 Visualizing time-varying volumetric data

When visualizing time-varying volumetric data, we are actually working with 4D data – volumes built up of 3 spatial dimension with the addition of time. These datasets can consist of data from only a few timesteps – a patient get-

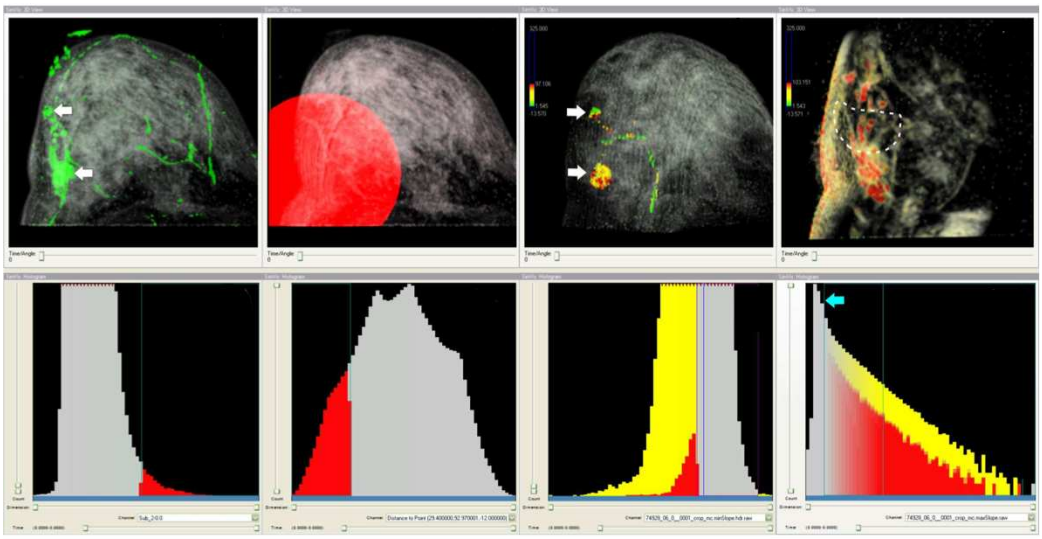


Figure 2.7: Visualization of breast tumor diagnosis with perfusion data using the SimVis system by Oeltze et al. [28] The user can brush interesting tissue blood flow features using linked multiple views and then track these features over time by scrolling over the timeaxis.

ting follow-up scans every few months to track the progress of a disease, to many hundreds of timesteps – captured brain activity from a fMRI session. Time-varying datasets are common in many different parts of the scientific visualization community, ranging from medical scans to engine simulations or even the visualization of large weather systems. Time-varying volumetric data (as with other higher-dimensional data) have a tendency to be quite large – added dimensionality quickly increase the size of the data. When vizualizing time-varying data, the goal of the visualizing is often to find and show interesting changes that occur in the data over time.

Most approaches for rendering time-varying data use either static or dynamic time dependency [29], they try to display changes in the data in a still image or in the form of a animation. Static time dependency visualizations can be really powerful – they are able to concentrate time-varying data into a single representation that can be observed all at once. The viewer does not have to relate to earlier animation steps that passed already seconds, or maybe even minutes ago. One such technique was presented in the work of Balabanian et al. [27], and it allow the interactive rendering of multiple timesteps of volumetric data into a single image using temporal styles and

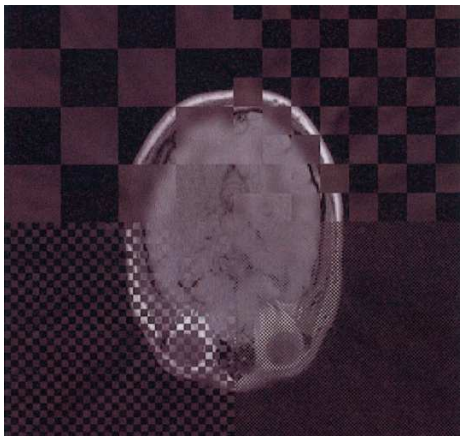
transfer functions, as shown in figure 2.6. Multiple timesteps from a simulation of hurricane Isabel is rendered in different styles on the same canvas, giving a good concentrated overview of the hurricane's progress over time. One of the main challenge with static methods is occlusion, fitting a lot of data into a single image can easily result in visual clutter. Because of this, the information that is displayed should be minimized but still contain the essence of the data. This can be accomplished through interactive selections by the user, or even by automatic solutions, such as feature and event trackers – algorithms that (semi-)automatically locate and track important changes and features in time-varying data [30] [31].

Dynamic representations can be used when the time-varying data is of a such a type that it is hard or impossible to condense the data enough to fit into a static representation. Many solutions for interactive visual analysis support exploration of time-varying data using dynamic time dependency, one example is the SimVis system discussed earlier in section 2.1.1. In the works of Oeltze et al. SimVis is used to explore time oriented blood perfusion data [28], as shown in figure 2.7. The system allows the user to scroll along the timeaxis, displaying the time-dependent data as an animation. Interesting features can be selected and tracked over time through brushing in multiple focus+context enabled views. A very helpful option for visualizing changes over time in such as system is functionality for deriving data. Data derivation can be applied to discover small and subtle changes of the data, by visualizing the derivative (data value change) between timesteps (instead of the original data values) hard to spot differences can be discovered.

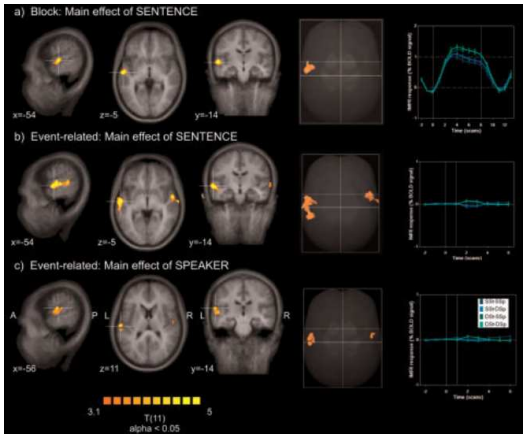
2.2 Inter-subject multi-volume visualization

Inter-subject volumes are volumes of the same modality captured from multiple objects or subjects, for example the medical scans from the ADNI- and OASIS-project discussed in chapter 1. In the most simple cases, only a few volumes need to be visualized, typically for comparison purposes. The most obvious methods for exploring and comparing only a few volumetric datasets is using multiple linked views or interlaced renderings. One example is the use of a checkerboard-patterned rendering scheme as seen in the works of Stokking et al. [32] in figure 2.8(a).

Multiple volumes of this type can be represented as 4D volumes (three spatial and one subject dimension), similar to time-varying volumetric data.



(a) Two overlaid datasets rendered for comparison in a checkerboard pattern by Stokking et. al. [32].



(b) Group-level random effects analysis on fMRI data from 12 subjects using BrainVoyager by Goebel et. al. [33].

Rendering the entire dataset becomes more difficult as the size of the subject dimension increase, and trivial techniques like side-by-side or checkerboard becomes more or less useless. Researchers working on large image studies therefore often move away from the raw data, and rely on derived values and statistical software. One example is a recent imaging study by Walhovd et al. [34] where attributes and measurements of structures in the data (like cortical thickness) are extracted using automated tools. Statistical software can then be used for hypothesis testing on the derived data.

BrainVoyager QX is a system for visualization and statistical analysis of structural and functional MRI-data often used in group studies of volume. Using BrainVoyager it is possible to apply statistical models such as a general linear model [36] (GLM) to volumetric activation data of multiple cortically aligned volumes, shown in figure 2.8(b) from a study by Goebel et. al. [33].

When visualizing large numbers of volumes there is not always a need for studying every single volume alone. Volumes can be grouped based on some criteria, and then reduced through averaging. Inter-subject averaging [35] [37] are techniques that take multiple volumetric datasets as input and provide a single volume as output – a volume representative for the whole group. These techniques are sometimes used for task and group comparison studies of functional magnetic resonance imaging (fMRI), shown in figure 2.8.

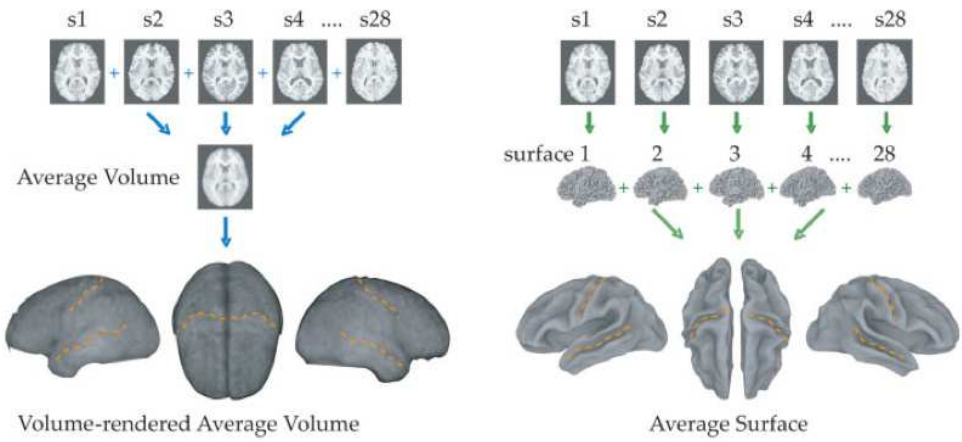


Figure 2.8: Techniques for creating average volumes and surfaces by Argall et al. [35]. One volume and cortical surface is created to represent the average of 28 different volumes, for further use in group comparisons in functional magnetic resonance imaging.

2.3 General techniques for high-dimensional volume visualization

Time-varying and multi-volume datasets can be generalized as four dimension scalar fields by not consider space and time/individual as separate entities. General techniques for visualizing higher-dimensional spatial data can then be used to explore and render the data as if they were 4D volumes (hypervolumes).

By ray-casting through hyper-slices and 3D projections of hypervolumes it is possible to discover interesting space-time relationships in some datasets, as demonstrated by Woodring et al. [38]. In figure 2.9 two example visualizations of 4D volumes (3D space + time) made by Woodring et. al.'s technique is shown, integrating over hyper-slices and hyper-projections using a normal ray-caster. While this technique does indeed provide interesting looking results, interpretation of the results can be a major challenge because of our limited ability to imagine and understand high-dimensional space. Understanding where a hyperplane is placed and what it means for the visualization is very hard for us, there are only a few naturally emerging hyperplanes that actually make sense.

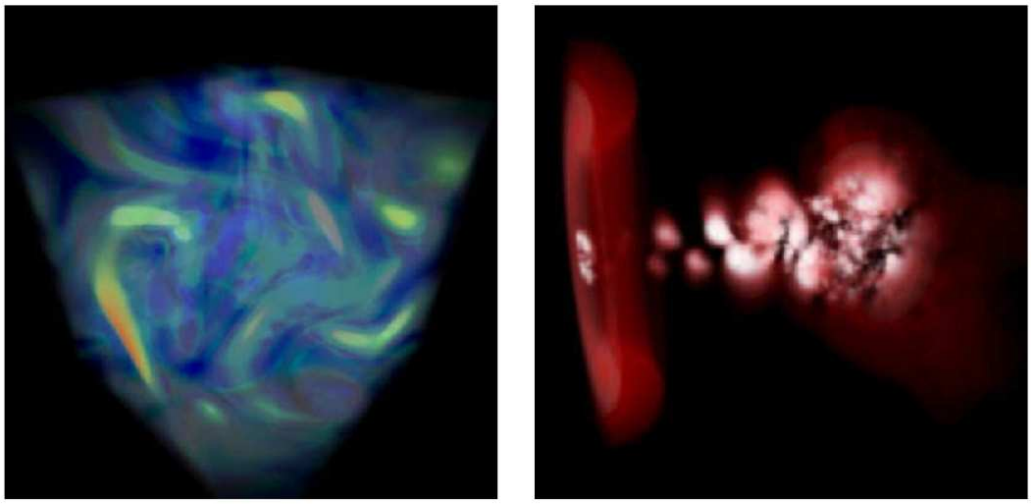


Figure 2.9: Visualizing time-varying volumes as high-dimensional data using hyper slicing and ray casting by Woodring et al. [38]. Here showing higher-dimensional data (3D+time) rendering of a vortex (left) and a jet (right).

3

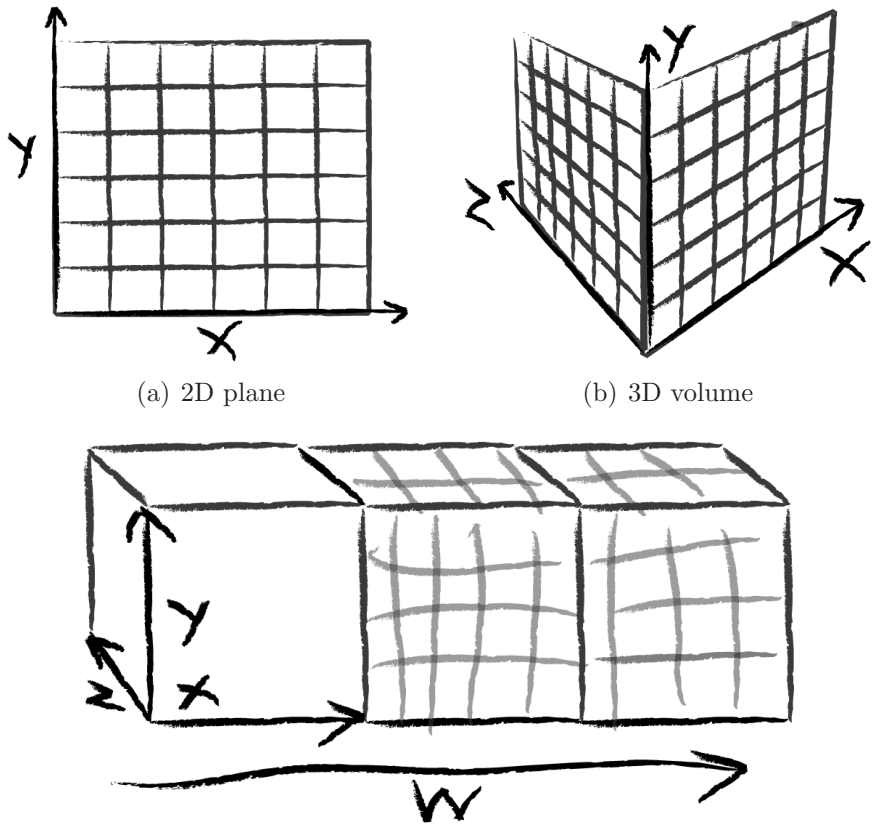
Multi-volume visualization and exploration

Suppose a person of the Fourth Dimension, condescending to visit you, were to say, 'Whenever you open your eyes, you see a Plane (which is of Two Dimensions) and you infer a Solid (which is of Three); but in reality you also see (though you do not recognize) a Fourth Dimension, which is not colour nor brightness nor anything of the kind, but a true Dimension, although I cannot point out to you its direction, nor can you possibly measure it.

- Flatland (1884), by Edwin A. Abbott

When faced with the task of visualizing and exploring multiple volumes – may they be separated by time, individuality, space, or modality – it is in fact a specific case of dealing with a four dimensional problem. Thinking about four (or even more) dimensions in general can be very challenging, it is somehow comparable to being an inhabitant of Abbott's Flatland and trying to understand our (3D) world. If all you have sensed your entire life existed in two dimensional space and all you have ever seen is lines and points, it would be very hard for you to understand the structures and shapes existing in our world. Try imagining living inside the surface of the paper of this thesis, what would it look like if a sphere temporarily pass through your two dimensions? You would experience it as a line, a line that grows and then shrinks before vanishing without a trace. It's the same for us when trying to relate to high-dimensionality and can make dealing with high-dimensional data really hard. There are however some special cases of high-dimensional data that we can relate to – if the extra dimensions are not spatial, but rather concepts we understand, for example time, or different items or individuals – it becomes much easier for us to interpret the data.

So what can we do to visualize and interactively explore four-dimensional datasets on a computer monitor? The computer monitor is only able to



(a) 2D plane

(b) 3D volume

(c) Simplification: Multiple 3d volumes

Figure 3.1: A conceptual model of how to think about 2D and 3D scalar fields, and collections of multiple 3D volumes.

display two dimensions, meaning that the four-dimensional data needs to be reduced by two whole dimensions before it can be displayed through a series of data and dimensionality reductions.

3.1 Multi-volume data

The volumetric data we want to visualize is typically in the form discretely sampled scalar values, captured at locations that are distributed evenly in space – a scalar field in Euclidian space. Each data point's location is inherently defined by, and can be accessed based on it's coordinates. For example, every data point in a 2D scalar field of this type can be accessed using it's X and Y coordinates, see the conceptual model in figure 3.1(a). The same holds for spatial 3D volumes, i.e. every data point in the volume can be accessed by

a vector describing it's $[X, Y, Z]$ location as shown in the conceptual model of figure 3.1(b). The distance between sampling points in real space can vary depending on the axis, for example in medical imaging a volume is usually built up from multiple 2D images captured at given distances, so points along the X- and Y-axis (an image slice) are therefore often sampled at a higher frequency than along the Z-axis (the distance between image slices). The volumetric data can have different resolutions depending on the axis.

As discussed earlier, four-dimensional volumetric data is hard to conceptualize, but often the 4th dimension encountered in scientific visualization is not spatial, but rather representing time or item / individual. In these situations it's easier to think about the data as illustrated in figure 3.1(c), as a stack of volumes. This data is accessed through an $[X, Y, Z, W]$-vector, where W either represents a spatial dimension in the case of a real spatial hyper-volume, or a given time-step / individual / modality, as in other cases. Some of the tools and techniques presented in the next sections will only make sense for use on real hyper-volumes, while others work best on stacks of multiple 3D volumes.

An added challenge of high-dimensional volumetric data is having to deal with the possibly large memory requirement needed to represent the data data. Every time an extra dimension is added, the amount of data has a tendency to increase exponentially. Handling a 500x500x500 voxel 3D volume is no problem for computers today, but if visualization calls for 500 time-steps then the large size of the data can turn out to be problem. So, not only can it be conceptually hard to understand the data, but can also be hard to handle for a computer, especially if everything needs to be represented in RAM for a visualization technique to work.

3.2 The goal

The goal for this thesis is to create a system of techniques capable of visually and interactively exploring higher-dimensional volumetric data – most typically in the form of multiple 3D volumes (time-steps, individuals, or modalities), but some of the techniques presented can be used for general hyper-volumes as well.

Exploring is a quest for uncover new knowledge about the (potentially unknown) data. Knowing exactly what to look for would make this task much easier, but that's the thing about exploring, you don't necessarily know

what you are looking for. To quote Matthew Ward and his visual information seeking mantra: *I'll Know it When I See it* [39]. Exploring is all about discovering new interesting features in the data, many of these discoveries can of course be anticipated, but the most interesting discoveries can often be ones that noone predicted were there in the first place.

Ben Shneiderman's visual information-seeking mantra [40] is a very good starting point for visualization and exploration in general: *Overview first, zoom and filter, and finally, details on demand.* First give the user a broad view of the data (an overview), allow him or her to drill down onto possible interesting parts of the data, and then eventually provide details as needed. This means that a system for exploration based on Shneiderman's mantra has three importation requirement:

- The ability to provide a good higher-level overview of the data.
- The ability to drill down, reduce dimensionality and visualizing interesting subsets of the data.
- The ability to visualize details in the data as it is needed.

A natural and popular basis for such as system is the use of multiple linked views and focus+context visualization, as discussed earlier in chapter 2. In this thesis we try to identify and assemble a series of techniques that together can enable the abilities listed above in a multiple linked views system. In addition, we want to keep it simple by using only a few powerful views, and also retain the connection to the original data. As discussed earlier, researchers often abstract away from the original multi-volume data, analyzing derived values using statistical software instead. Working only on derived data can be very good for hypothesis testing, but not necessarily good for exploration as it limits the ability to discover unexpected features in the data.

3.3 Two-level multi-volume visualization

Given the large amount of data contained in the multi-volume datasets we want to visualize, there is no question that exploration by only working directly on the volumetric data will be highly inefficient. Making a good overview visualization based on only high-dimensional raw volumetric data is hard, and we want to avoid this if possible.

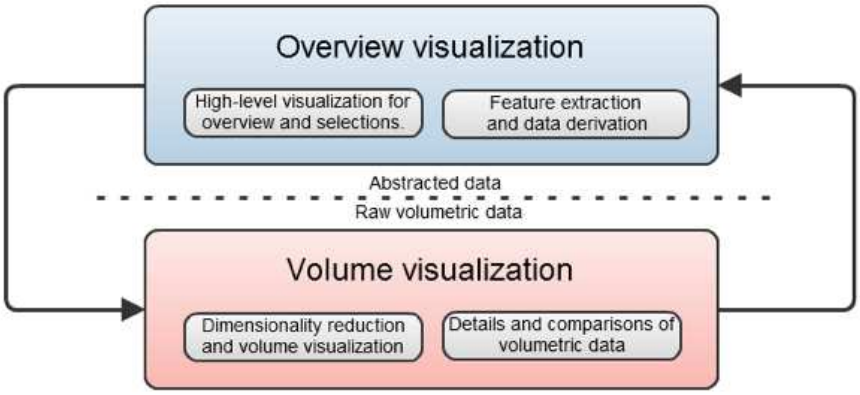


Figure 3.2: A two level approuch for visualizing and exploring multi-volume datasets, using derived values for an efficient overview visualization and visualizing raw volumetric data for drill-downs and details.

Through the need of providing a good overview and also having the ability to navigate and explore raw volumetric data the choice of a two level approach is natural. The concept is shown in figure 3.2. The overview visualizes extracted features and derived data (or metadata) instead of working directly on raw volumetric data. It is responsible for providing a good high-level overview of the multi-volume data. It needs to have the ability to make selections of parts of the data for further visualization, and also the ability to extract and derive new metadata as needed. The volume visualization on the other hand is responsible for visualizing raw volumetric data – the part of the data selected using the overview. It needs to have the ability to break down dimensionality (for drilling down into the data for exploration) and also be capable of making comparisons and visualizing details as needed.

3.4 The overview – metadata visualization

The main goal of this first component is to provide an overview of the data. Given the large amounts of raw volumetric data this is quite a challenge, both in terms of visualization and interpretation. So instead of relying directly on volumetric data this component will instead visualize derived data. While the required abstraction in general is domain-specific, there are many good sources for such derived data – both external and derived directly from the volumetric data.

3.4.1 Collecting metadata

To get a useful overview, we are interested in data that describe features or parts of our volumes. This could mean information about each time step when dealing with multiple volumes separated by time, information about each subject when dealing with multiple volumes from medical scans, or information about extracted features from 3D "slices" in the case of general 4D volumes.

Metadata can come from external and internal sources, both from before (from a preprocessing step) and during visualization. External datasources can for example be information describing a patient (in the case of volumetric data from the medical domain), volumetric data from medical scans are usually accompanied by large amounts of information about the patient as found in patient journals or general information like sex and age.

Extracting metadata from volumetric data can often be a very computationally heavy and time consuming operation and therefore needs to be done as preprocessing step if possible. Domain specific tools can be applied to (semi-)automatically extract features and segmentation information from scientific volumetric data. There exist algorithms that can be applied to detect and track critical points in time and space for features such as vortexes and shock waves in flow data [30]. For the medical domain several different automatic segmentation tools are available, such as Freesurfer, a tool capable of automatically segmenting volumes containing MRI scans – providing large amounts of information about each segment such as size, thickness and curvature. Other methods can discover horizons and faults in volumetric seismic data [41]. Such tools exist for many purposes of scientific volume visualization and by taking advantage of them it's often possible to extract large amounts of information about features, segments, or individual volumes.

It is also possible to derive metadata interactively during visualization, this can be done either using simple operations on the volumetric data or by manipulating existing metadata using mathematical functions.

Interactively deriving new metadata

It is possible to derive new metadata based on existing metadata by applying mathematical expressions to one or more metadata variable. For example, a user might want to know which subject's brain contains the largest left

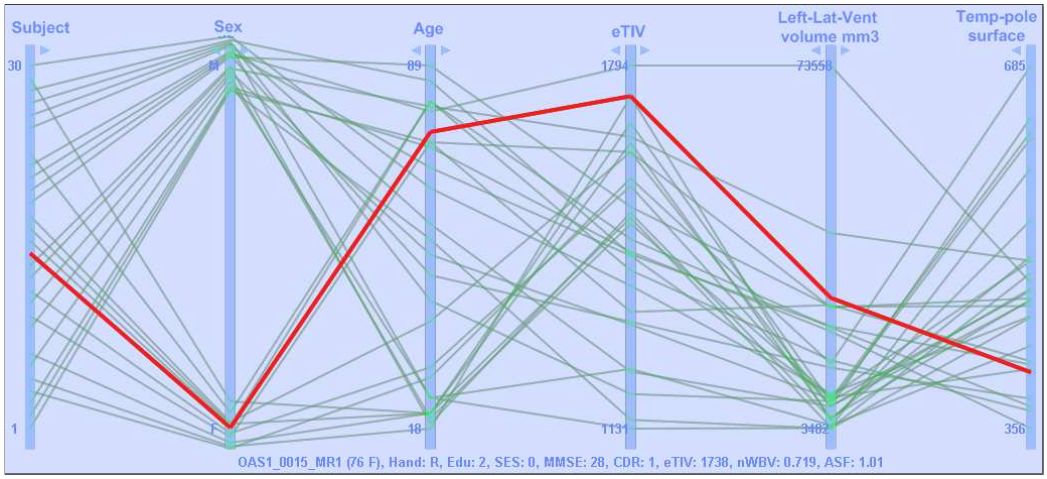


Figure 3.3: Metadata from 30 subjects being visualized in a parallel coordinates system, a mixture of both external metadata (sex, age) and derived (estimated total intercranial volume, volume of the left lateral ventricle, surface area of the temporal pole) is shown.

brain hemisphere, but only have the size of the segments building up the hemisphere available. It would then be easy to produce new metadata with the total volume of the hemispheres by summing up all of it's parts (given no overlapping segments).

By including a mathematical parser in the implementation, it's easy for the user to create new metadata. Examples of such data derivation can be seen in appendix B.

Intensity value counting

Another useful non domain specific metadata extraction method that work directly on raw data is *intensity value range counting*. The user can supply a value range he is interested in, the system can then count the number of voxels that fall within the defined range, in either slices or whole volumes, to produce new metadata. This can for example be used to find slices or volumes containing large bone structures in CT as bone show up as some of the highest intensity values.

3.4.2 Visualizing metadata

Using knowledge about features and attributes in the volumetric data we have a better basis for creating a good overview visualization. Some of the data derivation tools mentioned, such as Freesurfer for segmentation of MRI data, can produce hundreds of metadata variables from a volume. By turning to techniques from information visualization and interactive visual analysis the metadata can be visualized and navigated. A very effective visualization used when dealing with high-dimensional data is the parallel coordinates system, we have seen it used for volume exploration of a single dataset earlier [18] in chapter 2. We extend this idea and use it to provide an overview for exploring multiple (or higher-dimensional) volumes through metadata visualization.

A parallel coordinates system visualizes multi-variate data by plotting polylines (representing the data) across multiple axis representing different variables. In figure 3.3 we see a parallel coordinates system used to visualize metadata from multiple subjects, data about each subject (each coordinate along the volume w-axis) is represented as one of the green/yellow/red lines, by tracing the lines as they intersect with the axis we can read out the metadata values and see how this particular subject compares to the others. The parallel coordinates systems needs to be able to change the order of the axis, which can be crucial for discovering correlations. Other important features is to be able to scale axis to zoom in on important parts. Rotating or reversing an axis can be achieved through simple metadata manipulations. Parallel coordinates have a tendency to become quite cluttered when visualizing categorical data, a problem that can be avoided by jittering the position of an intersection on categorical axes by a random value, as demonstrated by the sex axis in figure 3.3

3.4.3 Making selections based on metadata

Another important feature allows to make selections in the parallel coordinate system. One often used mechanism for selections is brushing [14]. Brushing is a process where the user marks a value range to be selected by marking data in an interactive view. In the parallel coordinates system in figure 3.3 we see metadata from multiple subjects - including both external metadata (sex and age) and volume-based derived metadata (estimated total intercranial volume, volume of left lateral ventricle and surface area of temporal pole). A brush can be applied by marking an area or range us-

ing the mouse, lines passing through this area are considered selected and often rendered using a different style to differentiate the selected data from contextual information. Brushes have the ability to be moved, and it is also possible to combine multiple brushes through AND-combinations.

Using brushing, the user is able to select a range of interesting volumes / timesteps / features, or similar, which can be used as a basis for further visualization by the *Multi-volume visualization* component.

3.5 Multi-volume visualization

This component is responsible for the next part of the visual analysis process, to act as an interactive window into the raw volumetric data. While the parallel coordinates setup provide a good overview based on derived data, and allow selection of interesting volumes in a multi-volume dataset, there is also the need for visualizing and navigating raw volumetric data. There are several possibilities for breaking down multi-volume data for visualization in multiple views, to help chose the most useful one we have a set of requirements we need to fulfill:

- The ability to focus on a single volume and render it in detail.
- The ability to render and compare data from multiple selected volumes.
- Minimize the amount of views, if possible only use two or three for volume visualization.
- It would be preferable if the result could be rendered using a standard volume renderer.

The multi-volume 4D data is accessible through a vector $[X, Y, Z, W]$, where the W-axis typically indexes volumes. X, Y and Z is used to address coordinates in the corresponding 3D volumes. One of the requirements is to be able to render a single volume, meaning that one view should be used for rendering volumes it is possible to access by addressing the W-axis – this will be the *focus volume visualization*.

There is also a need for making direct comparisons between volumes in the multi-volume data, while this choice of view can be fairly domain specific, the system will mostly be used for visualizing medical data where it's natural to visualize and compare slices from the X/Y-plane. The second volumetric view is therefor based on the data addressable by the Z-axis, meaning X/Y-slices from all W volumes.

3.5.1 The focus volume

The focus volume visualization visualizes a single selected volume from the multi-volume 4D data. These are volumes that are accessible along W-axis of the $[X, Y, Z, W]$ 4D multi-volume, typically meaning a volume, time-step or feature. It consists of a standard volume renderer that by applyig direct volume rendering (integration) can output the volume to the screen as on the lower right in figure 3.5.

By using a cutting plane or slice selector as shown in figure 3.5 the focus volume visualization can be used for selecting a position along the Z-axis, which can be used as a basis for building the slice stack.

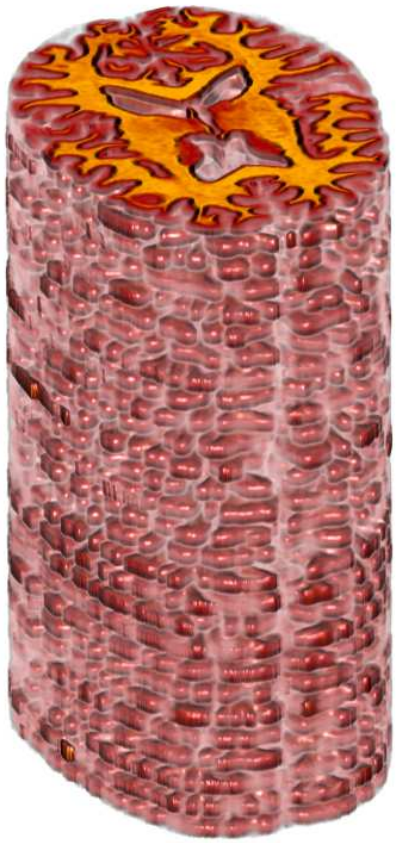


Figure 3.4: Slice stack – same slice from 30 subjects.

3.5.2 The slice stack

The slice stack is a simple but very useful visualization, it is built up from the same corresponding slice from multiple volumes. It's an representation of the XYW-volume for any selected slice Z as selected using the focus volume. Instead of including all volumes when creating the stack, a sorted subset of volumes is used, namely the selection from the metadata overview. In figure 3.4 an example stack built up from 30 different subjects from the OASIS-database is shown.

By linking the focus visualization with a cutting plane or slice selector for the stack, it is possible to select the volume displayed by the focus visualization by moving the selector up and down the stack. The duo of the slice stack and the focus volume are two very simple, yet powerful views, especially when linked. Interaction with a slice selector on the focus volume

allow moving over the entire 4D multi-volume (or selected subset) as the slice visualized is changed. And accordingly, interaction on the slice stack allow visualizing the entire 4D multi-volume (or selected subset) through time as the focus volume is changed.

The slice stack can then be rendered using direct volume rendering (integration). It is possible to use the stack both as a comparative tool and as a selection tool. Comparing the slices in the stack can be done by interactively cutting into the stack from different angles.

3.6 Bringing it all together – an example

By combining the three visualization components discussed so far, it is possible to create the setup illustrated in figure 3.5. On the top we see the metadata overview, on the lower left a multi-volume renderer in form of a slice stack and on the lower right the focus volume visualization.

In this example 30 volumes from the OASIS project have been loaded and a few of the many hundred metadata variables has been rendered in the overview: sex, age, estimated intercranial volume, left lateral ventricle volume and the size of the temporal pole surface area. A brush has been applied to select the volumes with the largest estimated intercranial volumes, as data is brushed the rendering style changed to differentiate it from the rest of the data. The brushed selection is used as a basis for building the stack visualization seen on the lower left. As the brush is moved to select new data, the linked slice stack is automatically updated. By moving the focus volumes' slice selector up and down, different selected slices is be used to build the slice stack.

The same tools that was used to build the stack is available for further cutting and slicing into the stack, here they have been applied to cut out part of the stack, this particular cut makes the location of the ventricles from each volume quite apparent. By scrolling through the stack along the z-axis (by interactively cutting) it's possible to get a good look at each slice in the stack, this is also used as an interaction tool to select the focus volume. By either cutting into the stack or using a linked slice viewer (as the black plane we see on the focus volume) a single volume is selected and displayed in the focus view, by moving through the stack the current focus element is updated. The metadata overview is also updated to show the current focus element, in figure 3.5 we see the contextual volumes as green lines, the brushed volumes

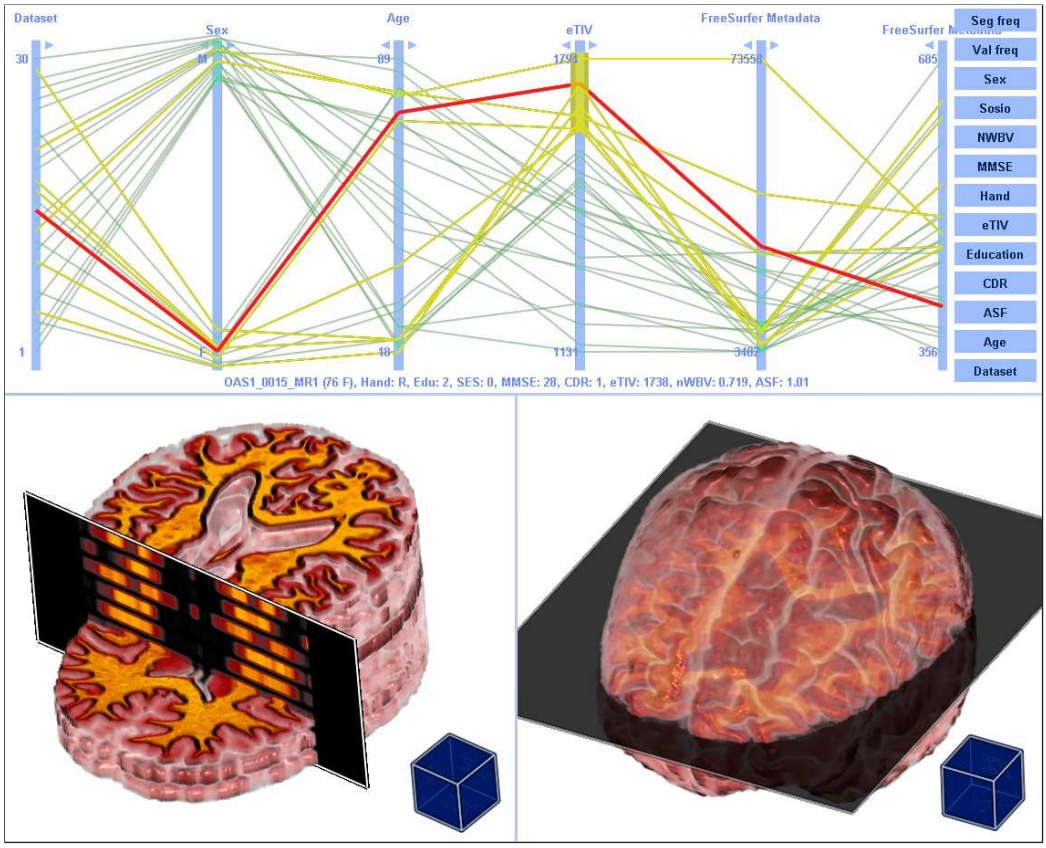


Figure 3.5: Selecting the same slice from 8 volumes with the largest intercranial volume using brushing in the top view. A direct volume rendering of a slice stack can be seen on the lower left. A direct volume rendered focus (detail-on-demand) of selected volume is shown on the lower right.

as yellow lines and the selected focus element as a red line.

By connecting the three components presented we achieve an efficient three-way interaction – the metadata overview selects volumes for the stack, the stack selects a focus volume to be displayed, the focus volume selects a slice for the stack. This interaction pattern allow for fast exploration of large collections of multiple volumes, as demonstrated in the next chapter.

4
Results

An attempt at visualizing the Fourth Dimension: Take a point, stretch it into a line, curl it into a circle, twist it into a sphere, and punch through the sphere.

- Albert Einstein

An excellent dataset to demonstrate the presented techniques and their implementation for multi-volume visualization is the OASIS database. It is one of the largest collections of freely available high quality MRI-scans, introduced earlier in section 1.1.2. To recap, the OASIS project is actually two different imaging studies – both focusing on dementia and Alzheimer's disease. One of them, the *"Cross-sectional MRI Data in Young, Middle Aged, Non-demented, and Demented Older Adults"* [7] study has captured imaging data from over 400 subject, and made the data available in a wide range of formats, along with a respective set of metadata describing each subject in the study.

The slice stack implementation relies on having corresponding slices in the different volumes, it therefore makes most sense to use the atlas-registered versions of the OASIS data. In the atlas-registered version, the brain of each subject has been extracted, scaled, and moved to fit inside an anatomical model (the classical Talairach atlas, also known as T88 [42]), and then transformed until reaching a correspondence between the atlas and the original scan. The result is that when viewing the same slice from multiple subjects we can expect to see the same structures – very useful for visualizing large amounts of data from multiple subjects. The atlas-registered datasets are available in the popular *Analyze 7.5* format. Also available is a simple grey/white-matter segmentation mask, classifying the tissue types of every voxel. The skull and other structures (other than the brain) hav been masked out in this particular version of the data, but unfiltered versions also exist. Each volume is 176x208x176 voxels large (every voxel represented by a 16

bit big endian scalar value) or about 12 MB in size.

All visualizations in this chapter is made from the following atlas-registered volumetric data and segmentation mask from the OASIS-project, each subject is identified by the OAS1_XXXX tag:

```
OAS1_0001_MR1_mpr_n4_anon_111_t88_masked_gfc.img
OAS1_0002_MR1_mpr_n4_anon_111_t88_masked_gfc.img
OAS1_0003_MR1_mpr_n4_anon_111_t88_masked_gfc.img
...
OAS1_0001_MR1_mpr_n4_anon_111_t88_masked_gfc_fseg.img
OAS1_0002_MR1_mpr_n4_anon_111_t88_masked_gfc_fseg.img
OAS1_0003_MR1_mpr_n4_anon_111_t88_masked_gfc_fseg.img
...
```

Also needed is the corresponding metadata for overview visualization, data that is available both as comma separated files and the popular XML-format. The OASIS project has collected the following information for every subject in their database:

Subject:
An identifier for each subject, ex: OAS1_0001. Used for connecting metadata information to a volume.

Sex:
Sex of the subject, 256 females and 160 males.

Handedness:
Handedness of the subject (all subjects in the study are right handed).

Age:
Subject's age, ranging from 18 to 96. Most of the subjects are either in their 20s or 70s and 80s.

Education level:
Education level corresponding to: 0 - not defined, 1 - less than high school, 2 - high school grad, 3 - some college, 4 - college grad, 5 - beyond college.

Socioeconomic Status:
SES - a standard reference for measuring a person's work experience, economic and social position. Ranging from 0 to 5 – 0 meaning low income and social position, 5 meaning high income and social position.

Clinical Dementia Rating:
CDR - Clinical assessment, a standard numeric scale to qualify the severity and stage of dementia: 0 - non-demented (316 subjects), 1 - very mild dementia (70 subjects), 2 - mild dementia (28 subjects), 4 - moderate dementia (2 subjects).

Mini-Mental State Examination:
MMSE - a quick standard 30 question test used for screening cognitive impairment. Maximum score: 30, anything above 25 is considered normal.

Estimated Total Intracranial Volume:
eTIV - size of intercranial volume in cm^3. Ranging from 1123 cm^3 to 1992 cm^3 in this study.

Normalized Whole Brain Volume:
nWBV - normalized estimate of white/gray-matter voxels in relation to the size of the brain. Ranges from 0.644 to 0.893.

Atlas Scaling Factor:
ASF - the volume scaling factor as applied by the automated atlas transformation. Ranges from 0.881 to 1.563.

4.1 Grazing the surface – metadata

After loading the large database of volumes we are presented with a multiple views setup including a metadata overview similar to the one seen in figure 4.1. This is a parallel coordinates system visualizing metadata of each subject. Each line represent a subject from the OASIS-database and each axis represent a metadata variable, axes can be added, or removed, and moved as needed. A polyline going from *Age* = 96 to *Dementia Rating* = 2 show that this particular subject was 96 years old and suffered from mild dementia. A red line show us the currently highlighted subject (metadata information about this subject is printed on the bottom of the view), yellow lines are the currently *brushed* subjects and the green lines represent the rest of the subjects (used for contextual information).

This metadata overview makes it possible to interactively select subjects using one or more brush strokes, brushed subjects are made selected and colored accordingly. It is even possible to move or swipe an existing brush quickly over the data, the metadata visualization is updated real time making

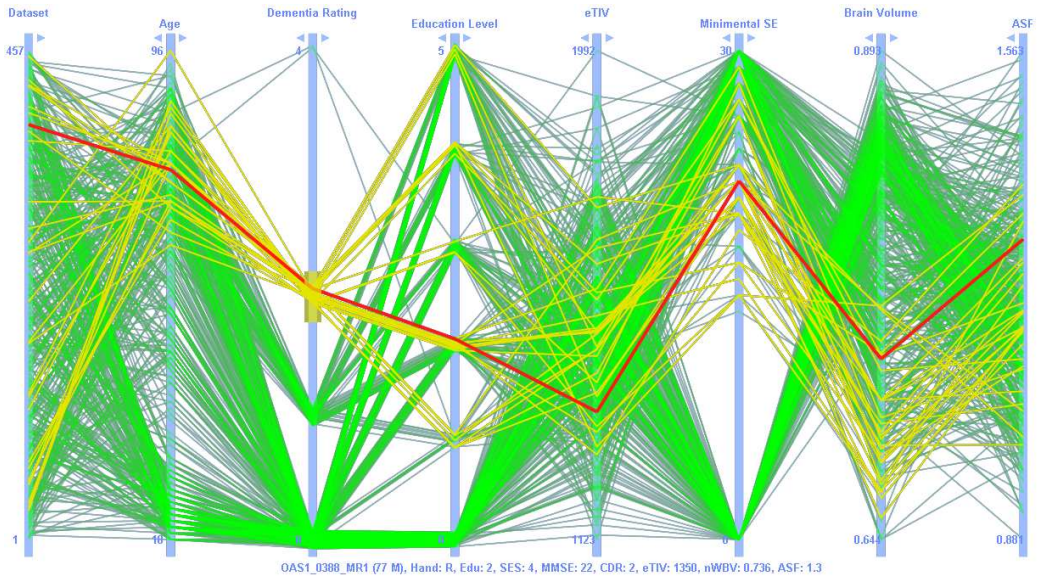
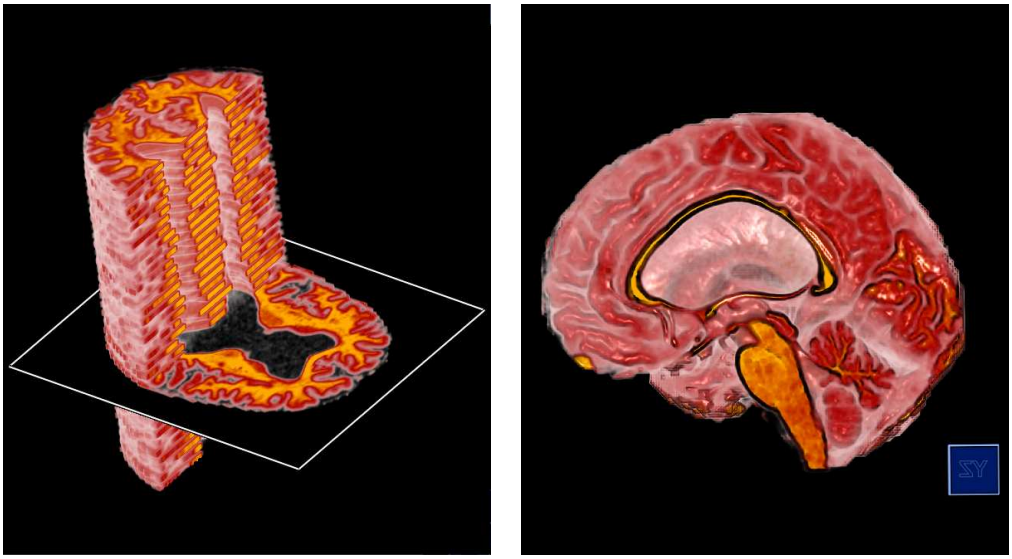


Figure 4.1: Metadata visualization giving an overview of the OASIS database using a parallel coordinates system. Each line represents a subject in the database, while the axes represent the different metadata variables.

it possible to quickly reveal correlations or interesting outliers in the data. In figure 4.1 a brush has been applied to the dementia rating axis, selecting all mildly dement subjects – visualized as yellow lines. It is immediately clear that all of these subjects are elderly people, the youngest person in the selection is a 67 years old man and the oldest a 97 years old man. It is not surprising of course, Alzheimer's disease and dementia in general are degenerative diseases that affect the brains of older people.

Also noticable in the overview with the selected mildly demented subjects, is the fact that not one of them managed a full score on the MMSE-test, and only a few of them managed a score of above what is concidered normal – 25 correct answers. There is also a clear correlation between normalized whole brain volume and dementatia rating to be seen in the overview – subjects where a lower percentage of the brain contains white and gray-matter seems more likely to be demented. This is actually one of the tell tale signs of Alzheimer's disease, of of the effects is overall shrinkage of brain tissue. Scientists have shown that healthy subjects have on average 10 percent larger brains than people suffering from Alzheimer's disease, and it even seems that having a big brain in the first place can protect you from the effects of Alzheimer's [43].



(a) Slice stack of the brushed subjects (shortened for illustrative purposes).

(b) Volume rendering of the selected subject's right brain half.

Figure 4.2: A slice stack build up of the corresponding slice from multiple brains, one of the slices has been selected from the stack on the left, and is rendered as a focus volume in the view on the right.

By using the parallel coordinates metadata visualization it is possible to visually find correlations and interesting features within minutes of loading the data, the real strength of this multiple views setup is however the tight link to the volumetric data. The next step is therefore to explore volumetric data using the parallel coordinates as a navigation interface.

4.2 Grazing the surface – volumes

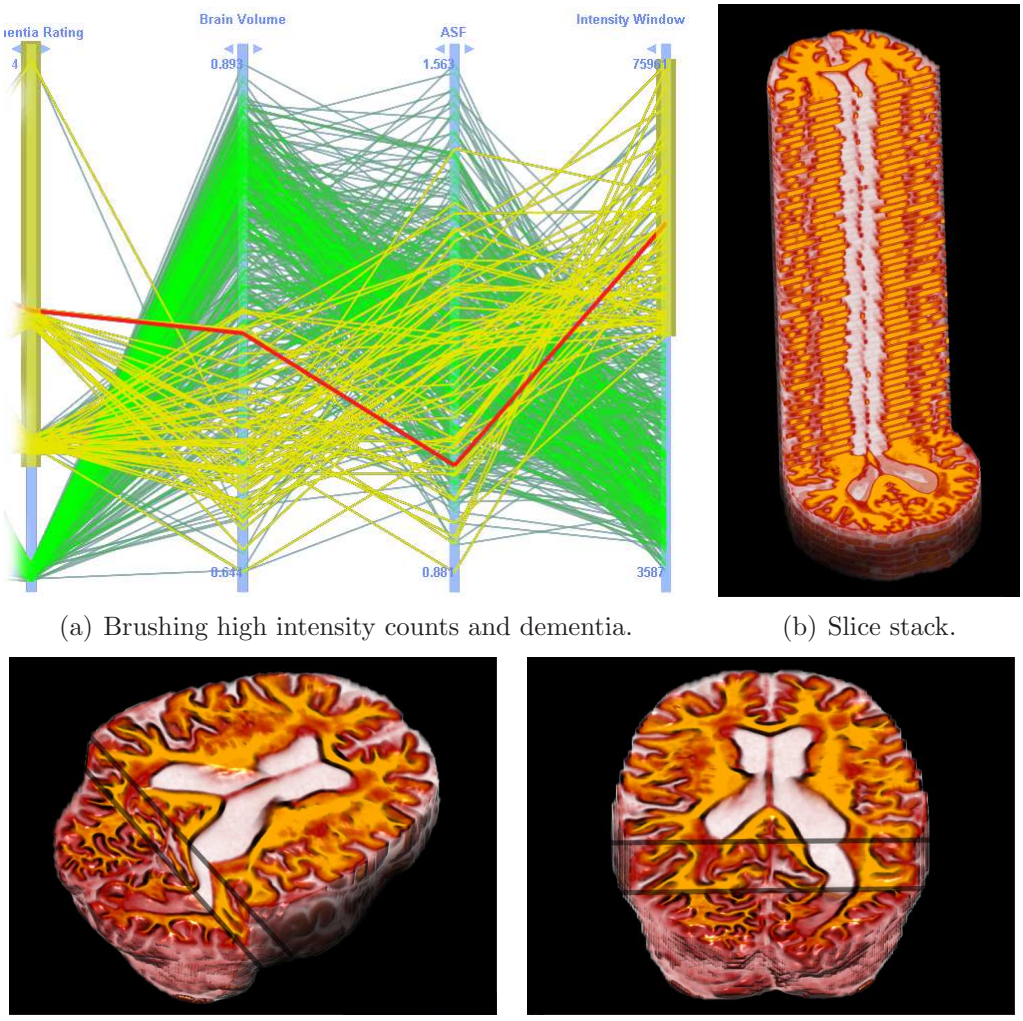
As a next step, the brush is extended to include all subjects classified with a dementia rating above 0 (very mild, mild and moderate), selecting about one hundred subjects. A slice more or less in the center of the brain (a slice cutting the lateral ventricles) is selected using a slice selector in the focus volume visualization. Based on these two selections a stack of slices is created, a stack built up from the corresponding slices from hundred selected subjects. By rendering the stack in a new view it is possible to interactively cut, explore, and make comparisons of slices. The selected subjects and the selected slice can of course be interactively changed as needed. By using

the mouse to move the selected slice up or down in the focus volume, it is possible to break down one of the dimensions of the dataset using interaction and time, thereby making the stack visualization both a useful interaction tool and window into the 4D OASIS dataset.

Some of the structures encountered by selecting a slice in the middle of the brain belong to the ventricle system. The ventricles are spaces filled with cerebrospinal fluid located near the center of the brain. By interactively cutting into the slice stack it's easy to compare the size of these structures across all the selected subjects. Immediatly a few of the selected subject stand out because of their large ventricles. Enlarged ventricles are one of the usual signs of Alzheimer's disease, some of the brain's volume has actually been replaced by fluid, meaning that these can be interesting subjects to inspect further. In figure 4.2(a) a small version of the slice stack is shown, a slice selector has been moved through the stack to reveal and select subject #21. The interaction of cutting into and moving up and down the stack can be very useful for visually finding outliers and anomalities, in this case we were able to very quickly locate this particular subject because of his very large ventricles. When selecting subject #21 a volume rendering of his full brain volume is shown in a neighboring linked view, the volume has been cut it in half to give an impression of the ventricles in figure 4.2(b). The selected subject is also highlighted in the parallel coordinates system, and we can see that this is an eighty years old male with very mild dementia. Another observation is that this subject's brain is actually quite large, with a total intercranial volume of around $1800cm^3$ and a scaling factor of 0.978 it's one of the largest brains in the database. So even though the ventricles are quite large his actual brain size is more or less normal. This is one of the strenghts of this approach, the ability to seamless relate to both volumetric data and metadata. By using the slice stack it is possible to see how structures in the currently selected (focus) volume compare to structures in other similar neighboring volumes, while at the same time being able to relate the selected (focus) subject to other subjects in the metadata overview.

4.3 Diving deeper – volumetric data derivation

Enlarged ventricles are very normal results of Alzheimer's disease, and it can therefore be interesting to try to find more brains with enlarged ventricles. Because ventricles are spaces filled with fluid, MRI register them as voxels with very low and uniform intensity values. Accordingly, it is possible to



(a) Brushing high intensity counts and dementia.

(b) Slice stack.

(c) Volume with asymmetric ventricles.

(d) Volume with asymmetric ventricles.

Figure 4.3: Looking through the stack of brushed large ventricles to discover a dataset with very asymmetric ventricles.

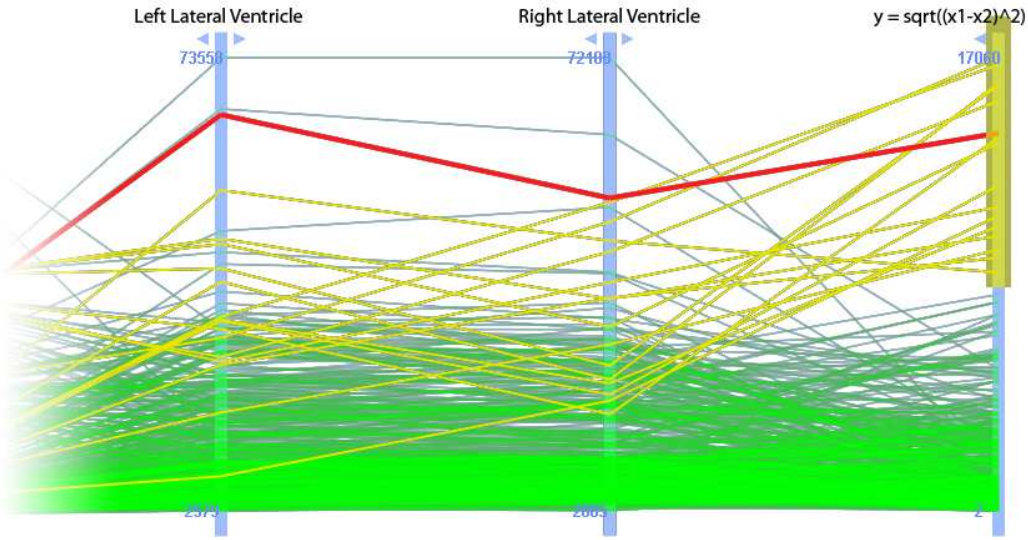
estimate the size of ventricles by defining a range of intensity values and do intensity value counting of voxels, the range can be defined interactively by selection in a single subject's brain. New metadata for the overview visualization is then generated by counting the number of voxels that fall within the defined range in either single slices or whole volumes. In this example counting intensity values between 3000 and 3600 in whole volumes provided the wanted result, shown as a new axis to the left in figure 4.3(a). By moving a brush over the new axis we notice that the hundred lowest values (smallest ventricles) all belong to healthy brains.

As a next step we are interested in a few of the subjects with large ventricles, subjects that also have been classified as having dementia. By using AND-combined brushes on the top half of the new intensity axis and the dementia axis, shown in figure 4.3(a), these subjects are selected. By scrolling through the slice stack as before, a few interesting subjects can be discovered, the volumes containing the largest ventricles is shown as expected, but a few of them stand out from the rest – because of asymmetry. Some of the slices, such as the one seen selected from the stack in figure 4.3(b) have really noticeable asymmetric ventricles, the right one is much larger than the left. By selecting it, the whole volume is rendered in a neighboring view as a focus volume. Cutting into the volume makes it possible to see the lateral ventricles, as in figure 4.3(c) and figure 4.3(d). By manually going through slices of the entire database it is possible to discover other similar subject, but maybe it is possible to automatically find them through further data derivation?

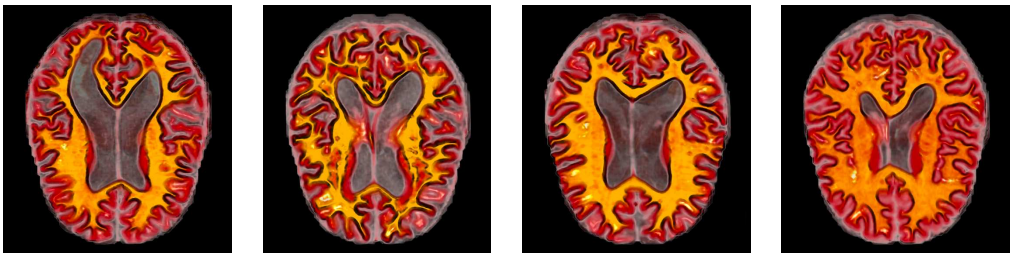
4.4 Diving deeper – metadata derivation

To find volumes containing asymmetric structures the program need a bit more information than what can be derived using intensity value counting. It would of course be possible to only count values from one half of the volume, but a better and more useful approach is to exploit knowledge from segmentation data – data automatically derived by existing tools such as Freesurfer [10]. Freesurfer can automatically label brain segments and provide statistical values through both volume and surface based analysis. However, segmentation is a slow process requiring hours per volume, making it something that needs to be done in preprocessing. Luckily, freesurfer data is already available on the OASIS FTP site for the OASIS datasets. By loading the freesurfer data into the program it is possible to access all kinds of measurements such as thickness, volume, surface size, curvature and similar for all freesurfer segmented structures in the brain.

With this new knowledge about the data, it is now possible to plot the volume of the *Left Lateral Ventricle* and the *Right Lateral Ventricle* into the parallel coordinates overview. It is easy to see the sizes of the left and right ventricles for the subjects, but this is not really the wanted information – we want to find subjects like #405 (figure 4.3), subjects with asymmetric ventricles. By using the system's data derivation feature it is possible to



(a) Brushing the top 15 largest ventricle differences.



(b) #205 (75F) (c) #73 (69M) (d) #124 (73F) (e) #376 (31F)

Figure 4.4: Deriving new data and selecting volumes based on a right and left lateral ventricle volume difference calculation.

create new metadata based on existing data. A new axis is created to visualize either the ratio between the left and right lateral ventricle, or the actual difference in volume:

$$y_0 = |axis_1 - axis_2| \tag{4.1}$$

$$y_1 = \frac{axis_1}{axis_2} \tag{4.2}$$

Using expression 4.1 it is possible to calculate values for absolute volumetric difference between the lateral ventricles for the new axis. It is also possible to find the relative difference by using expression 4.2, or even to scale the value by the intercranial volume or similar. The ability to do interactive metadata derivation while visualizing is a simple, yet very powerful. By applying expression 4.1 and brushing the fifteen largest values of the new

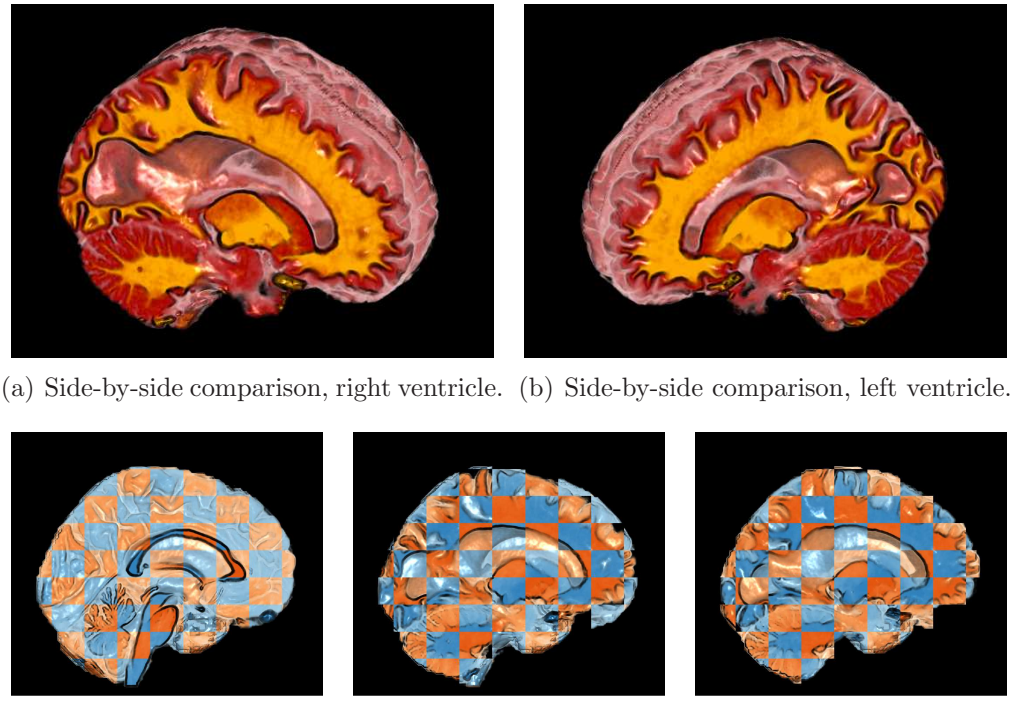
axis makes the parallel coordinates overview look like figure 4.4(a).

In figure 4.4 a subset of four of the fifteen brushed volumes with asymmetric lateral ventricles is show. We especially notice subject #205 (a 75 year old very mildly demented female) in figure 4.4(b) whose left ventricle is much larger and longer than the right, this is also clearly represented in the parallel coordinates system in figure 4.4(a) as the red line. Again, the system's linked views of volume and metadata visualization act together to provide a good view of this feature. Studies have shown that lateral ventricular asymmetry is present in about 5% of the population [44], making it a relatively normal occurrence and explaining why it's found in several of the OASIS volumes. There does not seem to be any clear links between ventricular asymmetry and dementia, but it has in some cases been linked to psychological disorders such as schizophrenia [44].

4.5 Focusing – details and comparisons

Instead of just comparing derived values in the metadata overview, it can in some cases also be interesting and useful to directly compare volumetric data – either as slices or rendered volumes. In the previous section a subject with highly asymmetric lateral ventricles was discovered, namely subject #205 (figure 4.4(b)). What steps can be taken to compare this subject's volume to others, or even compare the left and right ventricle directly? There are a few possibilities, one is to set up a couple of linked views as in figure 4.5(a) and figure 4.5(b). Side by side linked views allow direct comparison of the left an right lateral ventricle in this example, done by rendering the volume from two different viewpoints. Any interaction in one view is replicated in the other (mirrored) view, for example rotation and movement of applied cutting planes. A cutting plane has been applied to reveal the lateral ventricle in the same location on opposite sides of the brain.

Another method is to use selectively overlaid volumes rendered in a checkerboard pattern [32]. An implementation of this approach is demonstrated in figure 4.5(c). Subject #205's brain volume and a mirrored (X-axis) version is shown with a cutting plane applied to the center of the volume. By using different rendering styles or color schemes this can be a powerful method to compare volumes, it does however however require interactivity and the ability to move either the volume or the checkerboard pattern – static checkerboard comparisons can be hard to read, as shown in figure 4.5(d) and figure 4.5(e) where the left/right and right/left lateral ventricles are rendered.



(a) Side-by-side comparison, right ventricle. (b) Side-by-side comparison, left ventricle.

(c) Checkerbord comparison with mirrored brain, middle. (d) Left and right ventricle. (e) Right and left ventricle.

Figure 4.5: Different options for comparing volumes in use, we see a side-by-side linked view and a checkerboard version.

4.6 Bringing it all together

Finding possible correlations between variables visually can be done quickly and easy using the parallel coordinates system as shown earlier. With access to a wide array of metadata describing brain structures contained in the volumetric data of the OASIS databases, it should be possible to locate structures and properties that show the strongest correlation to high dementia rating values. By plotting the different metadata variables a few structures of interesting, appearing to be most strongly affected by Alzheimer's disease, is quickly identified. In figure 4.6 the resulting plot of the left cerebral cortex' volume and mean intensity, the left hippocampus' volume and mean intensity, and the inferior parietal lobule's average thickness is shown. Values for the corresponding structures on the right side of the brain where more or less the same. Also notice that the mean intensity values are negative because of axis reversal for increased readability by a simple $y = -x$ formula.

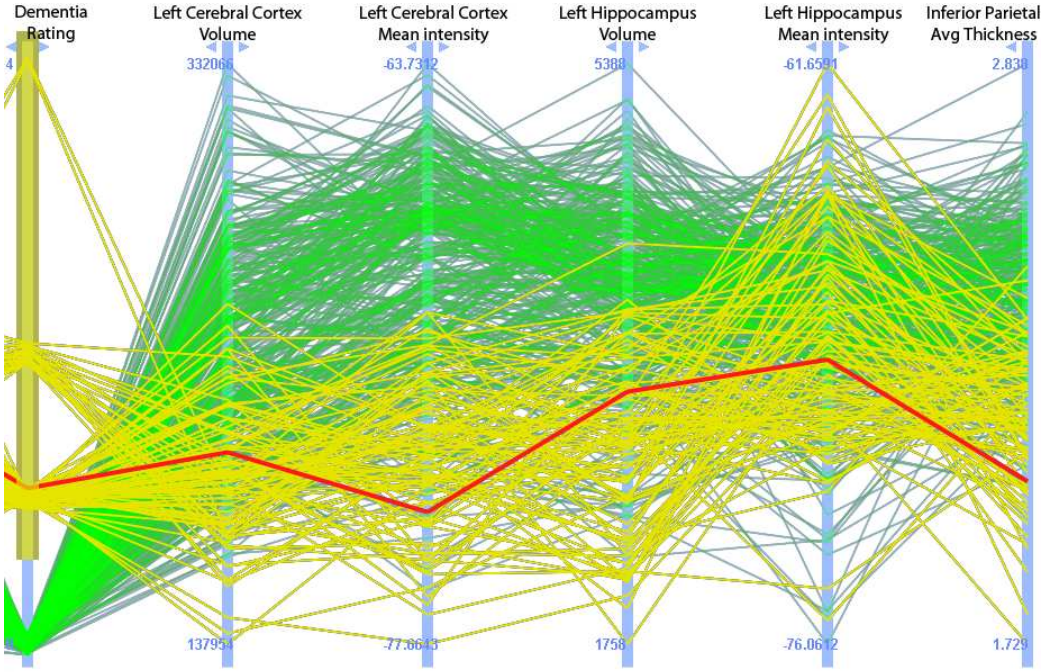
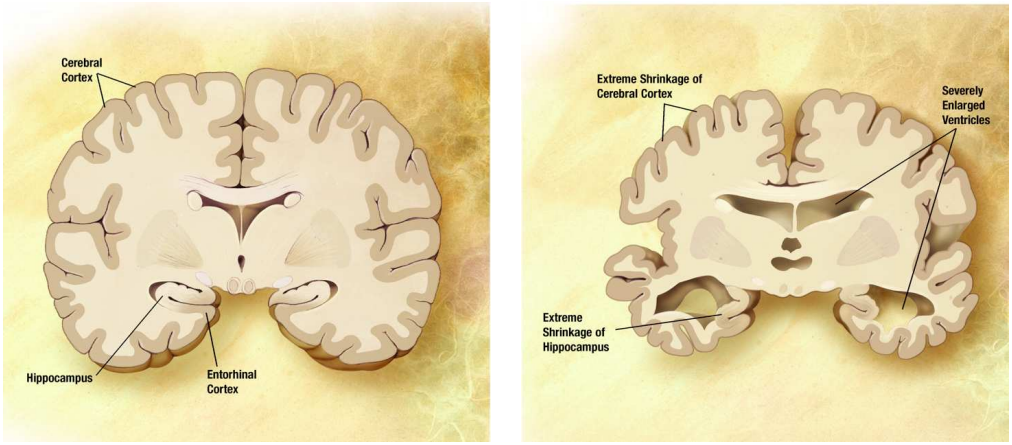


Figure 4.6: Trying to identify brain structures and cortical areas with a high correlation to dementia rating. Dementia ratings larger or equal to 1 has been brushed.

The cerebral cortex is the outermost tissue of the brain and is linked to functions such as memory and language. In the overview it is possible to clearly see that subjects diagnosed with Alzheimer's disease have a below average sized cerebral cortex in general. Maybe just as interesting is the above average mean intensity value for this structure, meaning a different white/grey-matter distribution for these subjects. White matter has a higher intensity value than grey matter and is therefore skewing the average towards a higher value. The currently selected subjects generally have a smaller cerebral cortex with less grey matter than healthy subjects, this is another of the main effects Alzheimer's disease has on the brain – atrophy of the cerebral cortex and grey-matter loss [45].

One other structure that seemed much smaller in OASIS subjects diagnosed with Alzheimer's disease is the hippocampus, generally they have a much lower hippocampus volume. In contrast to the cerebral cortex, there is however not much difference in the white-matter/grey-matter distribution. The hippocampus is a structure in the brain partly responsible for both long-



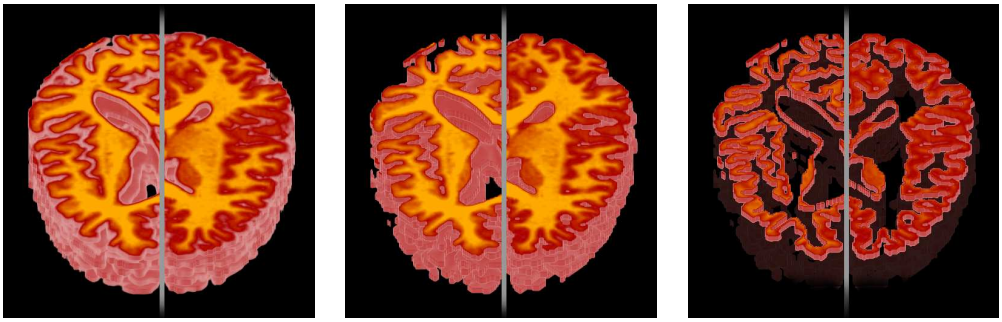
(a) Healthy aged brain. (b) Demented aged brain.

Figure 4.7: Public domain images from ADEAR (Alzheimer's Disease Education and Referral Center) showing the physical effects of Alzheimer's disease.

term memory and spatial navigation, it is also one of the first places that Alzheimer's disease strikes – causing memory loss and disorientation. In figure 4.7 an illustration of the effects of Alzheimer's on the hippocampus and the cerebral cortex is shown.

In an attempt to visualize grey-matter loss of the cerebral cortex from the volumetric data, two stacks are created, one based on 5 randomly selected healthy subjects and another based on 5 randomly selected demented subjects. By rendering the stacks next to each other as shown in figure 4.8 it's possible to see some of the effects of the grey-matter loss in the cerebral cortex. Even without the help of the white-matter/grey-matter segmentation mask the difference in color distribution is noticeable between the healthy and demented slices. The other slices further down the stack show the same results. The difference between the healthy and demented subjects become even clearer by applying a segmentation mask, especially when only rendering grey-matter as shown in figure 4.8(c) – the line representing the cerebral cortex' grey-matter seems much thinner in the case of the demented brains.

Hopefully this chapter has demonstrated some of the abilities of the proposed system for multi-volume dataset visualization. By using a linked multiple views setup consisting of a metadata overview and two volumetric renderers it is possible to navigate and explore large volumetric databases. We've seen that the parallel coordinates metadata overview can be an excellent tool for discovering correlation and interesting subsets of data for further exploration, and the importance of having the ability to interactively derive new



(a) No segmentation mask. (b) White and grey matter. (c) Only grey matter.

Figure 4.8: Comparing of a stack of 5 healthy brains (right) and a stack of 5 demented brains (left).

metadata. By keeping the link to the original volumetric data it is possible to drill down into the volumetric data through the use of a slice stack and a focus volume view. These two views together can efficiently be used as navigations tools and as windows for peering into the multi-volume (4D) data from the OASIS-project.

5 Implementation

In C++ it's harder to shoot yourself in the foot, but when you do, you blow off your whole leg.

- Bjarne Stroustrup

The project was implemented using the C++ programming language supported by QT, OpenGL, and some GLSL shader code. It was integrated into an existing framework for volume visualization called Volumeshop [24] as a series of plugins. Volumeshop is a highly adaptive and flexible framework originally made by Stefan Bruckner, containing many readily made components that can be combined as needed – making prototyping and implementing proof-of-concepts easier by not having to build everything from scratch. Given our requirement of loading the entire OASIS-database it's also a very good thing that a 64 bit version of Volumeshop is available, without it we would only be able to address 2 GB of data, while having the entire database in memory does requires somewhere around 6 GB.

Because of these features, Volumeshop is a great choice for implementing quick prototype projects for volume visualization. In the following sections we will see what components were made and how it all fits together with the existing framework.

5.1 Application overview

The project is built up by four main components, each responsible for their own tasks and operations. The components are linked together – interacting with each other. Each of these main components are realized by several subcomponents (or Volumeshop plugins). The main components building up our solution is the *Volumeshop Core*, the *Oasis Interactor*, the *Stack Visualization* and the *Focus Volume Visualization*. These are implementations of the three components described in chapter 3.2 with the addition of *Volumeshop*

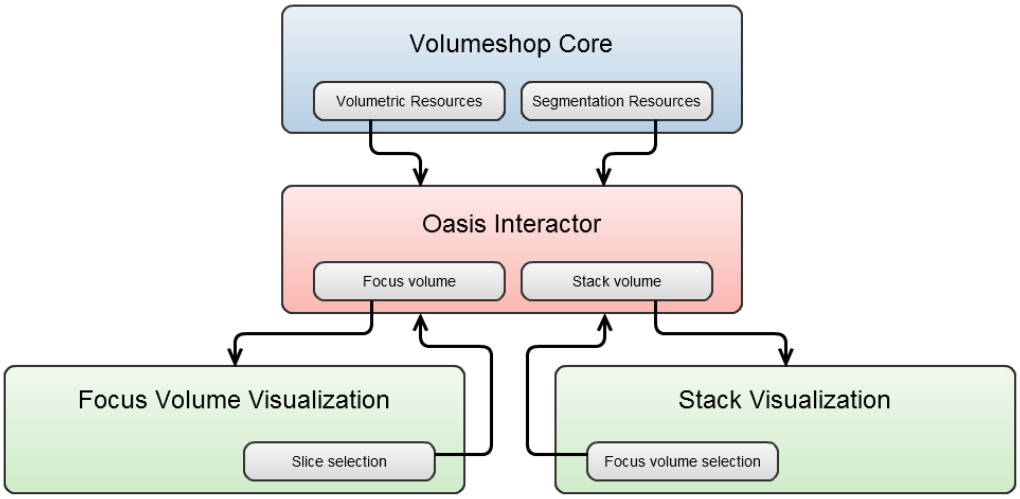


Figure 5.1: The main components and interaction between them.

Core which deals mainly with loading and bookkeeping of volumetric data. See figure 5.1 for a diagram of the components and how they interact.

Volumeshop Core is the foundation that everything else runs on top off and is responsible for, among others, loading and bookkeeping of volumetric and segmentation data. Some core changes in Volumeshop were necessary to enable loading of large lists of volumes and segmentation information, and also to keep track of the different types of loaded volumetric data for later use. This component is also responsible for providing the other components with resource lists as needed.

Oasis Interactor is the most important component in this setup and is an implementation of the *Metadata Visualization and Interaction* component from section 3.4. It's main task is to provide the user with an overview of all loaded volumetric datasets through metadata visualization. It is responsible for both loading and interactively deriving new metadata, and to allow the user to make selections of interesting volumes – a task accomplished through brushing of metadata in a parallel coordinates system. It is able to construct new volumetric data in the form of *slice stacks* based on the volumetric resources provided to it by the *Volumeshop Core*, the users brushing of volumes, and the selected slice from the *Focus Volume Visualization*. The produced *slice stacks* can be sorted as needed, for example by the size of a structure, and can also be further reduced in size by only including selected segments. It

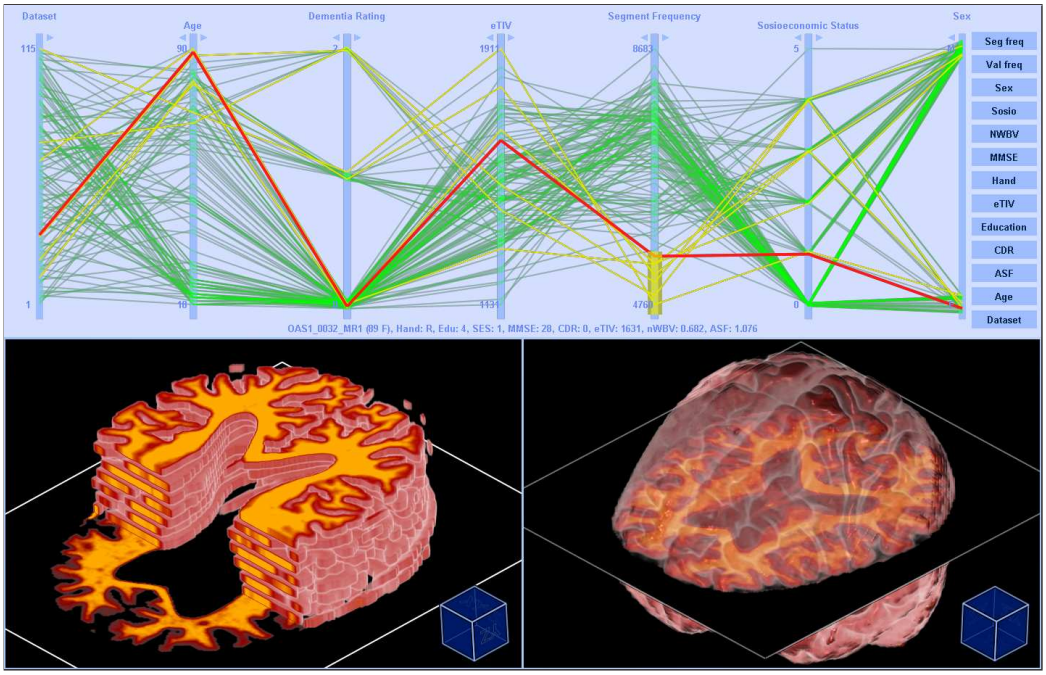


Figure 5.2: The main visualization components all linked together, the *Oasis Interactor* with metadata visualization and brushing on the top, *Stack Visualization* on the lower left and *Focus Volume Visualization* on the lower right.

exposes the created slice stacks to the *Stack Visualization* component and the currently selected volume to the *Focus Volume Visualization* component for rendering and further interaction.

Stack Visualization is an implementation of the *Multi-Volume Visualization* from section 3.5 and renders the stack volume provided to it by the *Oasis Interactor*. It allows the user to drill down into interesting parts of the volume collection for further exploration. Refined selections can be made by cutting into the stack with a slice selector. By moving the cut and selector up and down the stack, the currently selected subject's volume is reported to the *Oasis Interactor*, highlighting it in the metadata overview and rendering it in the *Focus Volume Visualization* component for detailed visualization of a single selected volume, as seen in figure 5.2.

Focus Volume Visualization is responsible for visualizing a single volume (or a comparison of volumes) and is an implementation of the *Focus*

Volume Visualization from section 3.5.1. It renders the selected volume provided to it by the *Oasis Interactor*, and allows cutting and slicing in the volume as needed. By moving the slice selector up and down in the volume the *Oasis Interactor* will update the stack so that it contains slices corresponding to the selected slice. A screenshot of the implementation of this component can be seen on the lower right of figure 5.2.

5.2 Volumeshop Core

Volumeshop Core is the existing main program which handles everything from setting up the graphical interface to loading volumetric data and other resources. Because Volumeshop is normally used for visualizing only a few volumes at the time there really isn't functionality for loading multiple volumes – every time a volume is loaded some interaction from the user is required. The task of loading over four-hundred volumes and four-hundred segmentation masks is too cumbersome when having to manually load one at a time. This meant that some core and GUI changes had to be implemented to allow batch loading of multiple volumes. Keeping track of the different types of already loaded resources was a bit challenging in Volumeshop, especially when you need to access all volumes or a subset of segments quickly. Accordingly, a basic resource book-keeping functionality had to be added.

5.2.1 Loading and tracking multiple volumes

The existing Resource Editor was upgraded to allow the loading of multiple volumes without user interaction (apart from selecting a list of files). This was done by iterating over the list of selected files, identifying the correct data loader plugin for the current file's file extension, create a new resource, and import the file using the most likely data loader plugin to this resource.

To keep track of the different resources a standard C++ Map was created for each type, mapping the OASIS identifier (ex: OAS1_0001) to the corresponding volumetric resource. The type of every imported resource is identified either as volume resource or segmentation resource and added to the appropriate map. These maps can then later be accessed by the other components either as a map (for fast lookup) or as a list (for iterating over all or subsets of volumes).

5.2.2 Data loader - ANALYZE™7.5

The OASIS-project data is available in the ANALYZE™7.5 format. While Volumeshop already contains loaders for many common data-formats, this one was not available. An ANALYZE™7.5 formatted dataset consists of two files, a header file describing a second raw datafile. The format is flexible enough to describe size, orientation, data type, compression, voxel dimensions, and so on. Because of time constraints, and because the OASIS-data is very consistent, it was possible to create a simplified reader, featuring just the basic functionality needed for reading volumetric OASIS-data (the main volumetric t88 registered data as well as the accompanied segmentation masks). For more information about the ANALYZE™7.5 format, see Robb and Hanson's article from 1991 [46].

5.3 Oasis Interactor

The task of this component is to give an overview and act as an interaction tool of the data that is visualized. In this section we look at the different sub-components that enable us to import, derive, and visualize metadata.

5.3.1 OASIS metadata reader

One of the available metadata sources in OASIS consists of general information about each subject such as age, sex, and so on. This information is available both in XML and CSV (Comma Separated File) format. A simple CSV parser was implemented that adds the information about each subject to a map, keyed by the OASIS identifier from the sample CSV file below. This allows for fast lookup of information during visualization.

```
ID,M/F,Hand,Age,Educ,SES,MMSE,CDR,eTIV,nWBV,ASF,Delay
OAS1_0001_MR1,F,R,74,2,3,29,0,1344,0.743,1.306,N/A
OAS1_0002_MR1,F,R,55,4,1,29,0,1147,0.81,1.531,N/A
OAS1_0003_MR1,F,R,73,4,3,27,0.5,1454,0.708,1.207,N/A
....
```

5.3.2 Freesurfer segmentation stats reader

After loading the general OASIS metadata some information about each dataset is now available, it is possible to supplement this with detailed information about the structures contained in each volume. The Freesurfer segmentations process generates three stats files for every subject. The first

one, aseg.stats contains information about the size and intensity values of 48 brain structures:

```
# ColHeaders  Index SegId NVoxels Volume_mm3 StructName
#  normMean normStdDev normMin normMax normRange
  1   2    156115   156115.0  Left-Cerebral-White-Matter
   108.2804    10.2808    27.0000   144.0000   117.0000
  2   3    188486   188486.0  Left-Cerebral-Cortex
    71.4874    10.8038     0.0000   135.0000   135.0000
  3   4     27330    27330.0  Left-Lateral-Ventricle
    21.9581    11.3552     7.0000    93.0000    86.0000
```

The two next files, lh.aparc.stats and rh.aparc.stats, contain information based on the cortical parcellation – one for the left and one for the right half of the brain.

```
# ColHeaders StructName NumVert SurfArea GrayVol ThickAvg
#  ThickStd MeanCurv GausCurv FoldInd CurvInd
  unknown                                    12183  7671
   10803  1.140 1.461     0.092     0.036  294.483  20.825
  bankssts                                     990   700
   1549  2.111 0.535     0.090     0.016    4.050   0.740
  caudalanteriorcingulate                      548   394
   1482  3.095 0.759     0.156     0.037   10.351   0.927
  caudalmiddlefrontal                         2939  2057
   5678  2.447 0.615     0.137     0.032   34.445   4.241
```

Parsing this isn't very different from CSV, a parser was implemented that reads these files and adds the information to a map like the previous importer.

5.3.3 Deriving new metadata

In addition to the already very large amount of metadata available (several hundred variables for every subject), it can be very useful to extract new specific metadata of features we're interesting in. We do this based on manipulating existing metadata mathematically or by extracting it from calculations directly on the volumetric data.

From existing metadata

By letting the user type in mathematically expressions and pass it through a mathematical parser and solver it's possible to easily create new metadata based on existing metadata. An existing parser was used, muParser, i.e., a fast open source math parser library by Ingo Berg[1]. The library was simply added to Volumeshop and set up to allow the user to write expressions,

[1] http://muparser.sourceforge.net/

```
for volumes in volumelist:
    count = 0

    if countWholeVolume:
        for slices in volume:
            for pixels in slice:
                if pixel intensity within range
                    count++
    else:
        for pixels in selected slice:
            if pixel intensity within range
                count++

    add count to metadata map, keyed by volume id.
```

Listing 5.1: Steps doing frequency counting in slices and volumes.

referencing existing (selectable) metadata variables as a and b. For example $y = a * b$ would give the new data the value of metadata variable a multiplied by metadata variable b. A large list of functions are available in this library, see Appendix B for more information.

From volumetric data

It is possible to extract new metadata directly from the volumetric data, one way to achieve this is by doing intensity counting. By letting the user interactively define a range of interesting values, the system can count the number of voxels with intensity values of interest in either slices or volumes. When counting values only from a particular slice, the slice selected in the *Focus Volume Visualization* is used. Listing 5.1 summarizes how intensity counting can be done.

From segmentation data

Another similar method to extract new metadata from volumetric data is to use the segmentation mask. The user selects a segment defined in the segmentation masks, the system can then count how many voxels in the selected slice or in the entire volume belong to the selected segment. The only differences compared to intensity counting is that counting is done based on segments in the segmentation volumes, not on intensity ranges in raw volumetric data.

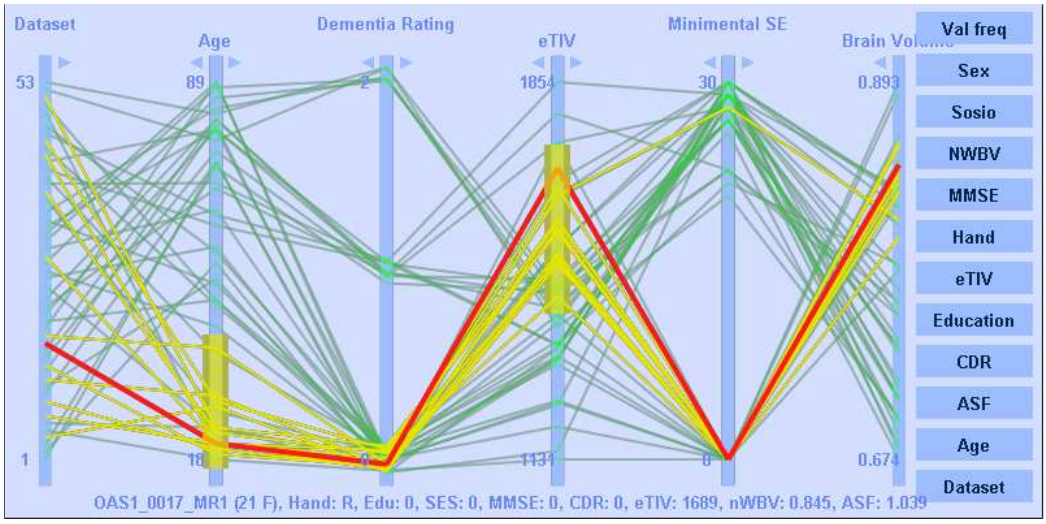


Figure 5.3: The parallel coordinates system used for visualizing metadata to give us an overview of the database.

5.3.4 Parallel coordinates – metadata visualization

The purpose of the parallel coordinates system (shown in figure 5.3) is to give an overview of the data by visualizing both existing and derived metadata, it is an implementation of the *Metadata Visualization and Interaction* component described in section 3.4. Every axis represents a metadata variable and every line represents a volume/subject's values for the given metadata variables. It's implemented using OpenGL by drawing a series of quads and lines on a canvas in Volumeshop.

Internally it contains a list of the currently added axis (each axis represent a metadata variable, such as age and dementia rating as in figure 5.3), every time a new axis is added by pressing one of the buttons on the right side of the widget (or deleted by pressing the middle mouse button on top of an axis) this list is updated. The list contains values such as the current axis scaling, which metadata type it displays, and how the data should be transformed and if jittering is enabled. An axis runs from 0.0 to 1.0, it's possible to change this scaling to get a better look at the data by zooming in on the interesting value scope. It's also possible to add a mathematical function to each axis to transform the data, as seen in section 5.3.3.

It contains a list of brushes with information about which axis a brush

```
clear the canvas
create empty list: renderlast

draw buttons

for all axes in axislist:
    draw quad representing the axis

for all volumes in volumelist:
    if volume is selected:
        add volume to renderlast list
    else:
        set render style to context-style
        for all axes in axislist:
            lookup value of current volume on axis
            scale value to unit interval
            apply jittering of value if enabled
            draw line from previous axis to current.

for all volumes in renderlast:
    set render style to brushed-style
    lookup value, scale, jitter, draw line

set render style to selected-style
lookup value, scale, jitter, draw line of selected volume
```

Listing 5.2: Steps for rendering the parallel coordinates system.

is applied to and the start and end values of the brush. In addition, it also uses a standard map, mapping volumes to a boolean value describing if a volume is currently selected. This enables fast lookup, needed for rendering the metadata overview.

Visualizing metadata

Every time data is updated, a brush or an axis is added, moved, or removed a new updated parallel coordinates system is rendered on the OpenGL canvas. First it renders the axes and buttons, then it renders subjects that are not brushed in a style suitable for context information (semitransparent green) before rendering the brushed subjects using a different style (solid yellow) and lastly the selected subject (solid thick red). To avoid cluttering of many lines when rendering categorical data, the y coordinate of a line can be jittered by a random value if needed. See listing 5.2 for a more detailed outline of the rendering process.

```
initialize map brushed<volume, boolean>
set all values in brushed map to true

if brushlist is empty:
    set all values in brushed map to false

for all brushes in brushlist:
    for all volumes in volumelist:
        lookup scaled data value of volume on brushaxis
        if data value is outside brush range:
            set brushed[volume] to false
```

Listing 5.3: Steps for brushing data.

Brushing

The parallel coordinates widget implements simple AND-brushing. The user can apply a new brush by pressing and releasing the right mouse button. The system records the y coordinate location of the press and release, and transforms it into a unit interval coordinate system as used by the axis. The brush is applied to the closest axis with respect to the release x coordinate. The entire brushing process is interactive, the rendering function described in section 5.3.4 runs every time the mouse is moved while pressing a button, enabling the user to see which data would be brushed if he or she were to release the button at any given position. When a new brush is created, a new object is added to the brush list, describing the brush axis, start and end point. Brushes can be moved by clicking and holding the left button, which works similar to the way new brushes are created. Brushes can be removed from the list by pressing the middle mouse button.

When a change to the brushlist has occurred a function to identify all brushed date is executed. Since we work using AND-brushes a volume has to pass through all the brushes to be considered selected. The function that takes care of calculating the brushed items can be seen in listing 5.3. It works by iterating over all brushes and all volumes, if a volume data value for a particular axis is outside the range of a brush it is removed from the brushed items list.

5.3.5 Building the stack volume

Every time the brushed volumes list has been updated (and no brush is currently being made or moved) a new slice stack is generated. The slice stack is a simply a volume built up from corresponding slices from all brushed

```
create brushedlist from brushed map values
sort brushedlist according to sorting criteria
create stack volume with size z = 3*(brushedlist size) - 1

for all volumes in brushedlist:
    load selected slice from volume

    if segmentation options selected:
        load segmentation volume
        apply segmentation mask to slice

    copy slice to slice stack
    copy slice to slice stack
    if current volume is not the last:
        add empty slice to slice stack

send slice stack volume to the GPU for rendering
```

Listing 5.4: Steps for creating a slice stack.

volumes, made in such a way that it easily can be rendered by the default Volumeshop volume renderer. The process of making the stack volume starts by the system creating a new volume, the volume size for the stack is the same in the x and y directions as any normal loaded volume. In the z direction it is three times the size of the number of brushed volumes minus one.

This new stack volume is then filled with corresponding slices from every selected volume, one slice from each is represented twice in a row in the slice stack to help avoid interpolation issues when rendering. Three copies of the same slice in a row has also been used, this allows for the use of a default volumeshop renderer without running into interpolation issues, but at the cost of higher memory usage for the slice stack. An empty slice is inserted between slices from different volumes to create spacing, making the slice stack build up be like the following example:

$$\{v_1, v_1, \emptyset, v_2, v_2, \emptyset, ..., \emptyset, v_n, v_n\}$$

See listing 5.4 for an overview of how the stack is built.

5.4 Stack Visualization

The stack visualization and interaction is achieved through the already existing volume and slice renderers in Volumeshop, *RendererVolumeSimple* and

RendererVolumeSlice. Through the use of a cropping tool, the *InteractorBoundingBox*, the user can interact with, crop, and cut into the stack of slices to inspect and get a closer look of the current selection. By linking either the slice renderer's or the cropping tool's position to the parallel coordinates system the currently selected slice is reported back and can be visualized in the metadata overview using a different rendering style. This provides the user with metadata information about this specific subject and because it is also linked with the *Focus Volume Visualization* we instantly get a view of the entire volume of the now selected slice belongs to. This makes the stack visualization functions both as a selection tool for the focus visualization as well as an overview and comparison tool of the currently selected volumes. See the lower left window of figure 5.2 for a view of a stack rendering.

5.5 Focus Volume Visualization

The task of the *Focus Volume Visualization* components is to provide us with a detailed volume rendering of the currently selected volume. As with the *Stack Visualization* components this is mostly built up from already existing renderers in Volumeshop such as *RendererVolumeSimple* and *RendererVolumeSlice*. In addition to this, a couple of simple shaders where implemented for comparison rendering (see 5.5.1) and contour rendering (see 5.5.2). See the lower right window of figure 5.2 for a focus volume visualization.

5.5.1 Comparison rendering

The comparison renderer loads two volumes onto the GPU and selectively renders the volumes in a variable sized checkerboard view using different transfer functions as styles. Each pixel is rendered based on a volume and style corresponding to which checkerboard block it belongs to, the fragment shader is able to determine this based on the code in listing 5.5. To really see the difference between two volumes it's a good idea to move either the checkerboard pattern or the volumes around a bit, static views like the ones in figure 4.5 are hard to read when using checkerboard rendering.

5.5.2 Contour renderer

The contour renderer is an implementation of a contour technique originally developed by Bruckner and Gröller [47]. The approach uses a curvature approximation along the viewing direction to control the thickness of contours

```
int xpos = int(gl_TexCoord[0].x * blocksize)
int ypos = int(gl_TexCoord[0].y * blocksize)

if (mod(float(xpos), 2.0) == 1) {
    if (mod(float(ypos), 2.0) == 1)
        drawVolumeA();
    else
        drawVolumeB();
else
    if (mod(float(ypos), 2.0) != 1)
        drawVolumeB();
    else
        drawVolumeA();
```

Listing 5.5: Comparison renderer fragment shader.

– making nice and uniform contours without too much computational overhead. To find out if a given sample is a contour we use Kindlemann et al.'s method [48]:

$$|n \cdot v| <= \sqrt{T * K_v(2 - T * K_v)}$$

where T is the contour thickness and K_v is the curvature approximation by Bruckner and Gröller [47]:

$$K_v = \arccos(currentNormal \cdot prevNormal)/stepsize$$

which is evaluated for every step taken along the view direction while integrating over the volume, *currentNormal* is the normal calculated at the current sampling position, *prevNormal* is the normal at the previous sampling position, and *stepsize* refers to the distance between sampling positions.

5.5.3 Slice selector

This component is also responsible for selecting the slice which is used for building up the slice stack, a task achieved by linking the current slice viewers position to the *Oasis Interactor*. This means that we have enabled three-way interaction between the main components; the *Oasis Interactor* visualizes metadata of the whole dataset and allows selections of subsets of slices, which in turn are visualized by the *Stack Visualization* component. The *Stack Visualization* component helps with further refining the selection to a single volume visualized by the *Focus Volume Visualization* component. This last component in turn lets us select the slice the *Oasis Interactor* bases it's stack on. In total there are three linked views, as shown in figure 5.2 – each one

Performance measures		
Operation	Time needed	Bottleneck
Loading 425 volumes (6 GB of data)	7 min 13 sec	Disk IO
Loading 425 volumes (6 GB of data) from SSD	4 min 21 sec	CPU
Slice frequency counting 425 volumes	¡ 1 sec	CPU/RAM
Volume frequency counting 425 volumes	1 min 3 sec	CPU/RAM
Generating stack – 10 slices	¡ 1 sec	CPU/RAM
Generating stack – 50 slices	1 sec	CPU/RAM
Generating stack – 100 slices	2 sec	CPU/RAM
Generating stack – 400 slices	5.9 sec	CPU/RAM
Data derivation – multiply two axis	1.2 sec	CPU

Table 5.1: Benchmark of the most computational intensive tasks done by the program.

responsible for it's own part of the visualization, but all acting together to provide both overview and detailed information about the data.

5.6 Performance

The project has been developed and used on a dual core desktop computer (Intel Core 2 Duo E6400) with 8 GB RAM and a 512 MB Nvidia Geforce 9600 GT graphics card running a 64 bit version of Windows Vista; a setup that is not exactly considered top of the line by today's standards. But, even on this computer the project implementation is highly interactive, with most operations experienced as being real time or close to real time. There are however some operations that take longer than others, mainly because of the massive amounts of data being handled. The loss of real time interaction at times is however not that much of a problem, many of these operations are done once only, like the initial loading of volumetric data or while deriving new metadata.

In table 5.6 a benchmark of the most computational intensive tasks can be seen, tasks that interrupt interaction. The most time consuming task is clearly the initial loading of volumes, taking over 7 minutes from a normal hard drive. It is hard to avoid this, using proxy volumes and only load data when it is needed will slow down performance later, so it is preferred to have as much data in memory as possible. Reading from a faster hard drive (such as a solid state drive) helps quite a bit – making the CPU become the new bottleneck. By threading and utilizing multiple cores it would b This is

however a task that only is done when first starting up, making it less of a problem.

It might be possible to make the frequency counting task faster by approximation, working on a lower resolution volume (such as skipping every other voxel) could make the counting twice as fast. Accuracy test would have to be done, but it is reasonable to believe that counting every other voxel would give acceptable results.

Generating large stacks is done quite fast, but even a 5 second delay is long when doing interactive exploration – a bit problematic because it is something that often is done. A solution to this performance issue could be to not replicate the same slice multiple times when generating stacks, and instead rely on a more advanced renderer, a renderer programmed to build a similar looking visualization based on a volume (3D texture) with single slices and interpolation disabled.

When doing data derivation from existing data one is actually doing a series of simple mathematical operations on multiple items. The operation is the same, executed on data from every volume – an operation that should be faster than the benchmarks, especially when working on only about 400 data items. Some performance could be most likely be gained by changing the implementation. This is also a case where a single instruction is executed on multiple data (SIMD), meaning that if set up in the right way it can be calculated really fast by using SIMD features such as SSE-instructions on a CPU. There also exist a SSE-enabled version of the muParser[2] library which could be useful for data derivation on really large numbers of subjects.

[2]http://beltoforion.de/muparsersse/math_expression_compiler_en.html

6
Summary and Conclusion

The brain – Use it or lose it!

- Unknown

6.1 Summary

Following the introduction of the CT-scanner almost 40 years ago we have witnessed an explosion in the use of volumetric data visualization. Volumetric data was first and foremost used within the medical realm, but has in the later years expanded to many different disciplines of science. How to render and explore single volume datasets interactively is by many considered a solved problem, and we are now starting to see a need for methods to visualize and explore sets of multiple volumes.

Large databases for image studies have been created to help solve the mysteries of diseases such as Alzheimer's disease. These databases can contain volumetric medical scans of several hundred different patients, one example is the *Open Access Series of Imaging Studies (OASIS) [7]* where over four hundred MRI datasets have been made available to the general scientific community. Following the availability of such databases is the need for tools that can be used to visualize and explore the data. When researchers work with large databases of volumetric data like the OASIS-project today, they often start by formulating a hypothesis and then derive the needed data (from the volumetric images) to test their hypothesis using statistical software. This efficiently decouples the analysis from the original data, something that can be a bad thing for data exploration where we do not always know what we are looking for, to quote Matthew Ward's visual information seeking mantra: *I'll Know it When I See it* [39].

Many different tools exist to extract information about features in volu-

metric data, one example is Freesurfer, a program able to segment and extract statistical information from volumetric MRI data. This thesis presents a two-level technique for visualizing multi-volume datasets that try to take advantage of this available metadata (information describing the raw volumetric data). The method in this thesis consists of a system for metadata visualization (a parallel coordinates system) in combination with two volumetric views – a *focus volume visualization* and a *slice stack*. The components are carefully linked, allowing efficient navigation of large multi-volume datasets.

Metadata overview

The metadata overview visualizes derived data using a standard parallel coordinates system, acting both as an overview and as a navigation interface. Parallel coordinates systems are very efficient for displaying high-dimensional data such as the large variety of metadata often available when visualizing multiple volumes. An example of a parallel coordinates systems visualizing metadata from the OASIS-project is show in figure 4.1 in chapter 4. In this parallel coordinates system, each volume is represented by a polyline intersecting several parallel axes (an axis represents a variable). The value of a variable is determined by which position the polyline intersect the variables axis, this makes it possible to easily compare and relate different datasets and variables to each other. By also including the concept of brushing it's possible to make selections of interesting subsets of volumes for further visualization of the raw volumetric data.

Volume view

This is the component responsible for visualizing and interacting with the raw volumetric data. It consists of two paired views, working together to providing different but synergetic windows into the volumetric data. The first view is the *slice stack*, a visualization of volumes selected from the metadata overview. The visualization is in the form of a slice stack, displaying one corresponding slice from every selected volume. It's possible to navigate up and down the stack to select a *focus volume*. The focus volume is a standard volume rendering representing the currently selected volume, while also acting as a selection interface for which slices to display in the slice stack. This duo of views along with the metadata overview allow for efficient navigation, exploration and comparisons of multi-volume datasets.

6.2 Future work

The techniques and implementation presented in this thesis provides a good basis for visualization and exploration of multi-volume datasets, such as data from the OASIS-project. An important aspect of the method is derivation, manipulation and visualization of metadata to provide a good overview of the volumetric data. Currently, only a simple parallel coordinates system is used for metadata visualization, and I believe that implementing other types of linked views for metadata visualization could prove useful, for example 2D and 3D scatterplots.

Researchers often turn to derived data and statistical software for hypothesis testing when working on large multi-volume datasets, it would be very interesting to combine an existing program for statistical analysis with the implementation made for this thesis. R[1] is an open source suite for statistical computing and graphics. I think that a fusion between the metadata and volume visualization capabilities presented in this thesis and the power of statistical analysis from a program like R could be a very interesting next step.

6.3 Conclusion

If one lesson can be learned from this whole endeavor, it is that even with the hippocampus and it's abilities for spatial orientation intact, relating to and navigating high-dimensional datasets can be very challenging. There is an old saying about the brain – "Use it or lose it!"– regular mental gymnastics have been proven to potentially delay the effects of a aging brain and Alzheimer's disease [49]. So to conclude this thesis, it is the hope of the author that the work presented here one day can help prevent diseases like Alzheimer's, either through the scientific contribution of multi-volume exploration or more likely, through the mental gymnastics it hopefully has provided both the author and it's readers.

[1] R – http://www.r-project.org/

A

Getting everything up and running

Because of the shear number of components that needs to be added and linked it can be a bit challenging to set up a new volumeshop project properly. It is recommended to start from an existing project, but if needed the following appendix provides give a few useful pointers that can be of help when setting everything up from scratch. The order of some of the steps can be important to avoid crashes or unexpected behavior, the program is a proof of concept implementation and not production level software.

A.1 General window setup

1. Start up volumeshop, maximize the window. Right click in the main volumeshop window or the black top bar and add the following components: *Plugins*, *Resources* and two new *Viewports*. It makes sense to move the resources and plugins panes so that they occupy only a small part to the left of the screen. The two viewports should be arranged on top of each other, one occupying the top half and the other one occupying the bottom half of the screen.

2. The top viewport is normally used for metadata interaction, and the bottom one for renderers. Add a new *Viewer* to the top one by right clicking in the viewport. Add two (or more if needed) *Viewers* to the bottom viewport for renderers. See figure A.1 for a possible window setup.

A.2 Loading the data

1. Load the main volumetric data by clicking on *Import Multiple Resources* in the *Resources*-tab. Locate the OASIS volume files (OAS1-

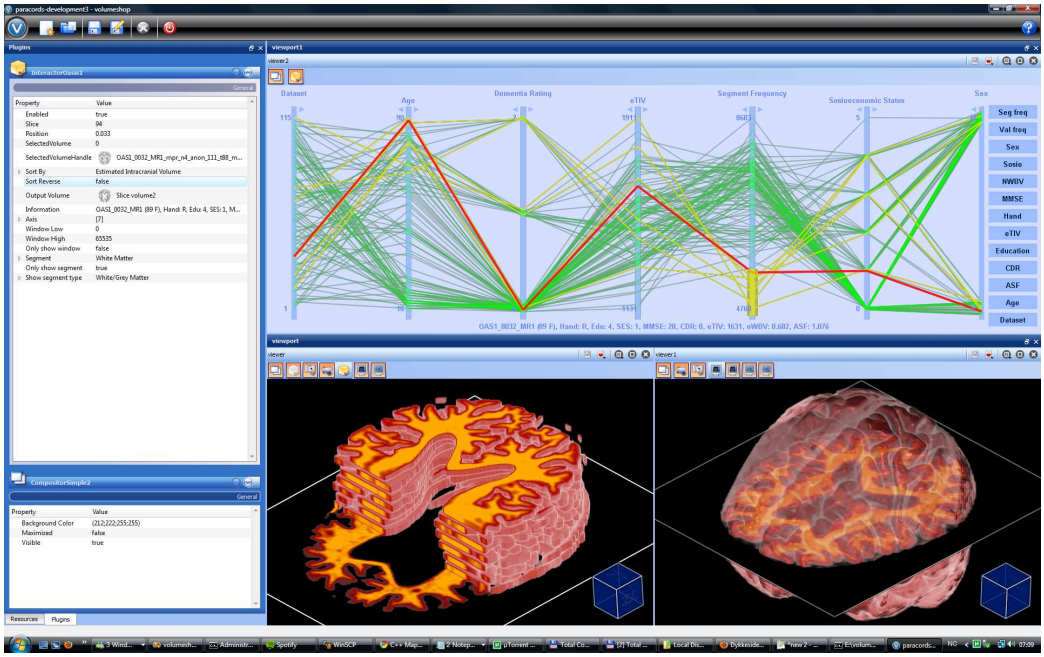


Figure A.1: Showing the program running with a functional linked window setup.

XXXX-MRX-mpr-nX-anon-111-t88-masked-gfc.img), select the needed files (multiple files can be loaded in one operation) and click load.

2. Segmentation masks are loaded the same way, *Import Multiple Resources*, locate the corresponding OAS1-XXXX-MRX-mpr-nX-anon-111-t88-masked-gfc-fseg.img files and click load. If you want to use any segmentation features in the program it is important to have the same volumes and segmentation masks loaded.

A.3 Metadata visualization - Top Viewport

1. In the top viewer, add the following plugins: *InteractorOasis*, *CompositorSimple*.

2. Activate both plugins. Add a new column (variable) to the parallell axis coordinates system by pressing one of the buttons on the right, this initializes the visualization. Also select a slice using the *Position* slider in the *InteractorOasis* interface, this initializes the stack output volume.

A.4 Stack visualization - Bottom left viewport

1. Add the following plugins to the bottom left viewport: *CompositorSimple*, *InteractorOrthographicCamera*, *InteractorViewingTrackball*, *RendererVolumeSimple*, *RendererVolumeSlice*, *InteractorBoundingBox* and *EditorColorTransferFunction*.

2. Activate all plugins.

3. Create the following links: Link the *Projection Transformation* from *InteractorOrthographicCamera* to the current viewer (right click on property and add link). Link *Viewing Transformation* from *InteractorViewingTrackball* to the current viewer.

4. In *RendererVolumeSimple* link the two transformations to the viewer. Select the generated *Slice volume* as volume. Select the transfer function that was just added. Link the *Cropping Box* to the *InteractorBoundingBox's Output Bounding Box*. Disable *Shading* and activate *Use Cropping Box*.

5. Setup the *Transfer Function*.

6. In *RendererVolumeSlice* link the two transformations to the viewer. Select the generated *Slice volume* as volume. Select the transfer function that was just added. Link *Position* to the *InteractorOasis' SelectedVolume* if needed, this allows the selection of volumes (red) in the parallell coordinates by moving the slice up and down along the stack. It can be smart to arrange the slice renderer in a position higher than the volume renderer in the plugin list to remove rendering artifacts.

7. Set up *InteractorBoundingBox* by linking *Input Bounding Box* to the stack volume *Slice volume's Bounding Box*.

A.5 Volume visualization - Bottom right viewport

1. Add the following plugins to the bottom right viewport: *CompositorSimple*, *InteractorOrthographicCamera*, *InteractorViewingTrackball*, *RendererVolumeSimple*, *RendererVolumeSlice* and *InteractorBoundingBox*.

2. Activate all plugins.

3. Link *Projection Transformation* and *Viewing Transformation* to the **other** viewer in *InteractorOrthographicCamera*, *InteractorViewingTrackball*, *RendererVolumeSimple* and *RendererVolumeSlice*. This will allow rotating both renderers at the same time.

4. In *RendererVolumeSimple* link *Volume* to *InteractorOasis' SelectedVolumeHandle*. The selected volume (red) will now show up in this view. Do the same for *RendererVolumeSlice*. Select transfer function for both *RendererVolumeSimple* and *RendererVolumeSlice*.

5. Link the *RendererVolumeSlice's Position* to *InteractorOasis' Position*, this is used for selecting the slice to build the stack from.

6. Set up the *InteractorBoundingBox* plugin to a random volume if needed (they all have the same dimensions).

You should now have a setup looking similar to figure A.1.

B

Data derivation functions

New data can be derived using operations from the built in math library. It's possible to define functions on the axis of the parallel coordinates system. By referring to the variable a one can access the variable of the current item and by using variable b one can access data from the current item from another selected axis. By defining a function such as a/b (real value of this axis divided by the value of the other selected axis) new data can be derived.

For example, to relate two axes to each other in terms of proportions:

$$f(a) = a/b$$

To normalize an axis' range to $[0, 1]$ (smallest and larges value can be read from the axis, but needs to be entered manually):

$$f(a) = (a - smallest)/(largest - smallest)$$

To scale an axis' range to $[-1, +1]$:

$$f(a) = ((a - smallest)/(largest - smallest)) * 2 - 1$$

To select the largest value of two axes:

$$f(a) = max(a, b)$$

To find the absolute difference between two axes:

$$f(a) = \sqrt{(a - b)^2)}$$

Mathematical operations and functions		
Op	Description	Arguments
+	addition	-
-	subtraction	-
*	multiplication	-
/	division	-
^	to the power of...	
sin	sine function	1
cos	cosine function	1
tan	tangens function	1
asin	arcus sine function	1
acos	arcus cosine function	1
atan	arcus tangens function	1
sinh	hyperbolic sine function	1
cosh	hyperbolic cosine function	1
tanh	hyperbolic tangens function	1
asinh	hyperbolic arcus sine function	1
acosh	hyperbolic arcus cosine function	1
atanh	hyperbolic arcus tangens function	1
log2	logarithm to the base 2	1
log10	logarithm to the base 10	1
log	logarithm to the base 10	1
ln	natural logarithm, base e	1
exp	e to the power of...	1
sqrt	square root of value	1
sign	-1 if less than 0, 1 if larger than 0	1
rint	round to nearest integer	1
abs	absolute value	1
if	if ... then ... else ...	3
min	min of all arguments	list
max	max of all arguments	list
sum	sum of all arguments	list
avg	average of all arguments	list

Bibliography

[1] V. Petrik, V. Apok, J. A. Britton, B. A. Bell, and M. C. Papadopoulos, "Godfrey Hounsfield and the dawn of computed tomography," *Neurosurgery*, vol. 58, no. 4, pp. 780–787, 2006.

[2] A. Berrington de Gonzalez, M. Mahesh, K.-P. Kim, M. Bhargavan, R. Lewis, F. Mettler, and C. Land, "Projected cancer risks from computed tomographic scans performed in the United States in 2007," *Archives of internal medicine*, vol. 169, no. 22, pp. 2071–2077, 2009.

[3] G. Hounsfield, "Nobel lecture," tech. rep., The Medical Systems Department of Central Research Laboratories EMI, 1979.

[4] W. E. Lorensen and H. E. Cline, "Marching Cubes: A high resolution 3D surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

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

[6] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans, "BrainWeb: Online interface to a 3D MRI simulated brain database simulated brain database," *NeuroImage*, vol. 5, no. 4, pp. 425–429, 1997.

[7] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *Journal of Cognitive Neuroscience*, vol. 19, no. 9, pp. 1498–1507, 2007.

[8] S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett, "The Alzheimer's disease neuroimaging initiative," *Advances in Behavioral Biology*, vol. 57, no. 1, pp. 183–189, 2008.

[9] D. S. Marcus, A. F. Fotenos, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): Longitudinal MRI data in nondemented and demented older adults.," *Journal of Cognitive Neuroscience*, vol. 22, no. 12, pp. 2677–2684, 2009.

[10] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo,

N. Makris, B. Rosen, and A. M. Dale, "Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain," *Neuron*, vol. 33, no. 3, pp. 341–355, 2002.

[11] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[12] R. Fuchs and H. Hauser, "Visualization of multi-variate scientific data," *Computer Graphics Forum*, vol. 28, no. 6, pp. 1670–1690, 2009.

[13] H. Doleisch, M. Mayer, M. Gasser, and H. Hauser, "Case study: Visual analysis of complex, time-dependent simulation results of a Diesel exhaust system," in *Proceedings of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization*, pp. 91–96, 2004.

[14] R. A. Becker and W. S. Cleveland, "Brushing scatterplots," *Technometrics*, vol. 29, no. 2, pp. 127–142, 1987.

[15] M. Cohen and K. Brodlie, "Focus and context for volume visualization," in *proceedings of Theory and Practice of Computer Graphics*, pp. 32–39, IEEE Computer Society, 2004.

[16] H. Hauser, "Generalizing focus+context visualization," in *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pp. 305–327, Springer Berlin Heidelberg, 2006.

[17] H. Doleisch, P. Muigg, and H. Hauser, "Interactive visual analysis of hurricane Isabel with SimVis," 58, VRVis Research Center, 2004.

[18] T. Möller, "A parallel coordinates style interface for exploratory volume visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1, pp. 71–80, 2005.

[19] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.

[20] E. J. Wegman, "Hyperdimensional data analysis using parallel coordinates," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664–675, 1990.

[21] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *VIS '90: Proceedings of the 1st conference on Visualization '90*, pp. 361–378, IEEE Computer Society Press, 1990.

[22] J. Beyer, M. Hadwiger, S. Wolfsberger, and K. Bühler, "High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1696–1703, 2007.

[23] J. Beyer, *GPU-based Multi-Volume Rendering of Complex Data in Neuroscience and Neurosurgery*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2009.

[24] S. Bruckner, I. Viola, and M. E. Gröller, "Volumeshop: interactive direct volume illustration," in *ACM Siggraph 2005 DVD Proceedings (Technical Sketch)*, pp. 60–61, ACM, August 2005.

[25] S. Born, W. Jainek, M. Hlawitschka, G. Scheuermann, C. Trantakis, J. Meixensberger, and D. Bartz, "Multimodal visualization of DTI and fMRI data using illustrative methods," in *Bildverarbeitung für die Medizin 2009*, Informatik aktuell, pp. 6–10, Springer Berlin Heidelberg, 2009.

[26] W. Cai and G. Sakas, "Data intermixing and multi-volume rendering," *Computer Graphics Forum*, vol. 18, no. 3, pp. 359–368, 1999.

[27] J.-P. Balabanian, I. Viola, T. Möller, and E. Gröller, "Temporal styles for time-varying volume data," in *Proceedings of 3DPVT'08 - the Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, pp. 81–89, 2008.

[28] S. Oeltze, H. Doleisch, H. Hauser, P. Muigg, and B. Preim, "Interactive visual analysis of perfusion data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1392–1399, 2007.

[29] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, "Visualizing time-oriented data – a systematic view," *Computers and Graphics*, vol. 31, no. 3, pp. 401–409, 2007.

[30] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," *Computer Graphics Forum*, vol. 22, no. 4, pp. 775–792, 2003.

[31] F. Reinders, F. H. Post, and H. J. W. Spoelder, "Visualization of time-dependent data using feature tracking and event detection," *The Visual Computer*, vol. 17, no. 1, pp. 55–71, 2001.

[32] R. Stokking, I. Zubal, and M. A. Viergever, "Display of fused images: Methods, interpretation, and diagnostic improvements," *Seminars in Nuclear Medicine*, vol. 33, no. 3, pp. 219–227, 2003.

[33] R. Goebel, F. Esposito, and E. Formisano, "Analysis of functional image analysis contest (FIAC) data with BrainVoyager QX: From single-subject to cortically aligned group general linear model analysis and self-organizing group independent component analysis," *Human Brain Mapping*, vol. 27, no. 5, pp. 392–401, 2006.

[34] K. B. Walhovd, A. M. Fjell, J. Brewer, L. K. McEvoy, C. Fennema-Notestine, D. J. Hagler, R. G. Jennings, D. Karow, and A. M. Dale, "Combining MR imaging, positron-emission tomography, and CSF biomarkers in the diagnosis and prognosis of Alzheimer disease," *American Journal of Neuroradiology*, vol. 31, no. 2, pp. 347–354, 2010.

[35] B. D. Argall, Z. S. Saad, and M. S. Beauchamp, "Simplified intersubject averaging on the cortical surface using SUMA," *Human Brain Mapping*, vol. 27, no. 1, pp. 14–27, 2006.

[36] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society*, vol. 135, no. 3, pp. 370–384, 1972.

[37] B. Fischl, M. I. Sereno, R. B. Tootell, and A. M. Dale, "High-resolution intersubject averaging and a coordinate system for the cortical surface," *Human Brain Mapping*, vol. 8, no. 4, pp. 272–284, 1999.

[38] J. Woodring, C. Wang, and H.-W. Shen, "High dimensional direct rendering of time-varying volumetric data," in *Proceedings of the 14th IEEE Visualization 2003*, pp. 417–424, IEEE Computer Society, 2003.

[39] M. O. Ward, "A taxonomy of glyph placement strategies for multidimensional data visualization," *Information Visualization*, vol. 1, no. 3/4, pp. 194–210, 2002.

[40] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *proceedings of the 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[41] D. Gibson, M. Spann, and J. Turner, "Automatic fault detection for 3D seismic data," in *Proceedings of Digital Image Computing*, pp. 821–830, 2009.

[42] J. Talairach and P. Tournoux, *Co-Planar Stereotaxic Atlas of the Human Brain: 3-Dimensional Proportional System : An Approach to Cerebral Imaging.* Thieme Medical Publishers, January 1988.

[43] D. Erten-Lyons, R. L. Woltjer, H. Dodge, R. Nixon, R. Vorobik, J. F. Calvert, M. Leahy, T. Montine, and J. Kaye, "Factors associated with resistance to dementia despite high Alzheimer disease pathology," *Neurology*, vol. 72, no. 4, pp. 354–360, 2009.

[44] M. S. Buchsbaum, S. Yang, E. Hazlett, B. V. Siegel, Jr, M. Germans, M. Haznedar, S. O'Flaithbheartaigh, T. Wei, J. Silverman, and L. J. Siever, "Ventricular volume and asymmetry in schizotypal personality disorder and schizophrenia assessed with magnetic resonance imaging," *Schizophrenia Research*, vol. 27, no. 1, pp. 45–53, 1997.

[45] G. B. Frisoni, M. Pievani, C. Testa, F. Sabattoli, L. Bresciani, M. Bonetti, A. Beltramello, K. M. Hayashi, A. W. Toga, and P. M. Thompson, "The topography of grey matter involvement in early and late onset Alzheimer's disease," *Brain*, vol. 130, no. 3, pp. 720–730, 2007.

[46] R. A. Robb and D. P. Hanson, "A software system for interactive and quantitative visualization of multidimensional biomedical images.," *Australasian Physical & Engineering Sciences in Medicine*, vol. 14, no. 1, pp. 9–30, 1991.

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

[48] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, "Curvature-based transfer functions for direct volume rendering: Methods and applications," in *Proceedings of the 14th IEEE Visualization 2003*, pp. 513–520, IEEE Computer Society, 2003.

[49] D. A. Snowdon, L. H. Greiner, J. A. Mortimer, K. P. Riley, P. A. Greiner, and W. R. Markesbery, "Brain Infarction and the Clinical Expression of Alzheimer Disease: The Nun Study," *JAMA*, vol. 277, no. 10, pp. 813–817, 1997.