FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Computational Methods enabling Interactivity in Analysis and Exploration of Volumetric Images

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

eingereicht von

Florian Schulze
Matrikelnummer 1129831

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao. Prof. Eduard Gröller

Diese Dissertation haben begutachtet:

(Ao. Prof. Eduard Gröller)	(Ass. Prof. Univ. Doz. Dr. med. univ. Stefan Wolfsberger)

Wien, 20.12.2012

(Florian Schulze)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Computational Methods enabling Interactivity in Analysis and Exploration of Volumetric Images

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Florian Schulze
Registration Number 1129831

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao. Prof. Eduard Gröller

The dissertation has been reviewed by:

(Ao. Prof. Eduard Gröller)

(Ass. Prof. Univ. Doz. Dr. med. univ. Stefan Wolfsberger)

Wien, 20.12.2012

(Florian Schulze)

Erklärung zur Verfassung der Arbeit

Florian Schulze
Dittesgasse 2, 1180 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Acknowledgements

I would like to thank Katja Bühler for giving me the opportunity to work on various very interesting projects which are the basis of this work and Prof. “Meister” Eduard Gröller for being my supervisor for this dissertation.

I would like to thank all the people I cooperated with:
Stefan Wolfsberger, the work on the neurosurgery project and especially the time in the operating room is an experience I will never forget. Special thanks also for being my 2nd supervisor.
Leslie Holton and Ross Smetzer, seeing how much effort is necessary to transform basic research to real medical products was an important lesson in my career.
Barry Dickson, allowing me to get insight into such a “mysterious” field as circuit neuroscience is a mind-broadening experience.

I would like to thank my parents Hanne und Rainer Schulze for their support. It’s been more than twelve years since I started my education in computer science and it wouldn’t been possible without you.

The presented work was funded by the Competence Centers for Excellent Technologies (COMET, VRVis-Center 824190), the Vienna Spot of Excellence (VSOE, VCV-Visual Computing 339889) and the Kplus-Kompetenzzentren program (Kplus 811485).

Abstract

Volumetric imaging is widely used in medicine and life sciences allowing to gain insight into otherwise opaque objects. Volumetric images do not unveil their content in a direct way due to their spatial structure. Therefore a variety of computational methods are used for visualization and processing which allow to explore and analyze the data.

Analysis and exploration of the data is usually performed in an interactive way either manually or with support of semi-automatic algorithms. It is crucial for an efficient completion of the task that the system performs interactively and responsively. Thus, software supporting the user in an effective way relies on three basic requirements. First, the system must deliver feedback in a short period of time. Second, results of any computation must be presented or visualized in a way that the user can efficiently recognize the important information. Third, the user must be able to efficiently control, initialize or adjust the algorithm through a suitable user interface. In this thesis four approaches are presented which aim to solve different aspects of the problem of enabling interactivity in analysis and exploration of volumetric image data.

The first presented project studies the design of an application which has strict limitations concerning the user interface due to the application environment which requires almost a hands free interaction. The problem is approached by the development of efficient and robust visualization which makes adjustments needless, and by the development of sophisticated interaction patterns which reduce the needed interface to the minimum.

The second project focuses on methods which optimize a computationally intensive feature detection task that can be used in an interactive scenario which requires the algorithm to produce results in just a few seconds. To achieve this goal the probabilistic boosting tree classification algorithm is extended and optimized for runtime and memory efficiency.

The third and the fourth project focus on the interactive exploration of large image and object collections. Two approaches are presented for this problem area. For the retrieval of neuronal objects by similarity, measures for different neuronal substructures have been developed which are able to perform efficiently on large amounts of data. For retrieval of images and objects by local means such as neighborhood, overlap, local image expression and local image similarity a sophisticated updatable high performance index structure has been developed. The index allows to store local properties of volumetric data in a space efficient way and to retrieve this data with low latency.

The presented projects demonstrate that the challenge of achieving interactivity often is the development of methods which allow to balance processing speed with result quality. Furthermore it is shown that time performance is not the only property which needs to be respected, result presentation as well as interaction patterns deserve similar attention and contribute greatly to an interactive user experience.

Kurzfassung

Volumetrische Bildgebung ist ein Verfahren, das ermöglicht, Einblick in eigentlich undurchsichtige Körper zu erhalten. Sie werden daher häufig im medizinischen Bereich und in den Biowissenschaften angewendet. Volumetrische Bilddaten geben ihren Inhalt allerdings nicht ohne Weiteres preis. Deshalb wird eine große Anzahl verschiedener Visualisierungs- und Verarbeitungsmethoden verwendet, die bei der Analyse und Explorierung der Daten helfen können.

Analyse und Explorierung sind interaktive Vorgänge, die entweder manuell oder mit Hilfe von automatisierten Verfahren durchgeführt werden. Für eine effiziente Durchführung dieser Aufgabe ist es wichtig, dass sich das verwendete System interaktiv und flüssig verhält. Daher müssen Systeme, die den Anwender bei diesen Aufgaben unterstützen, drei Bedingungen erfüllen. Erstens: Das System muss dem Benutzer in jedem Fall innerhalb kurzer Zeit Rückmeldung geben können. Zweitens: Berechnete Ergebnisse müssen in einer Form dargestellt werden, die es dem Benutzer ermöglicht, die wichtigen Informationen gut zu erkennen. Drittens: Das System muss eine Kontrollmöglichkeit bieten, die es erlaubt, die Software effizient zu steuern.

In der vorliegenden Dissertation wird anhand von vier Projekte gezeigt, wie verschiedene Aspekte des Interaktivitätsproblems gelöst werden können.

In dem ersten Projekt wird der Entwurf einer Applikation vorgestellt, die, bedingt durch ihren Einsatzort, stark in ihrer Interaktionsmöglichkeit beschränkt ist. Dies wird gelöst, indem zum einen robuste Visualisierungsmethoden verwendet werden, die Interaktion möglichst unnötig machen, und zum anderen ein spezielles Bedienungskonzept erarbeitet wird, das Interaktion auf das Minimum beschränkt.

Das zweite Projekt behandelt Methoden zur Optimierung eines rechenaufwändigen Verfahrens zur Landmarkenlokalisierung, sodass diese in einem interaktiven Anwendungsfall eingesetzt werden können. Die Ergebnisse müssen in wenigen Sekunden vorliegen. Dies wird erreicht, indem das “Probabilistic Boosting Tree”-Klassifikationsverfahren durch verschiedene Methoden erweitert und optimiert wird und so den gestellten Bedingungen bezüglich Laufzeit und Speicherverbrauch genügt.

Das dritte und das vierte Projekt setzen den Fokus auf den Aufgabenbereich der interaktiven Exploration von großen Objekt- und Bildsammlungen. Zwei verschiedene Herangehensweisen werden präsentiert:
Erstens: Zum schnellen Auffinden von ähnlichen Objekten in einer Datenbank für neuronale Substrukturen werden spezialisierte Ähnlichkeitsmaße sowie Verfahren zur schnellen Auswertung dieser vorgestellt.
Zweitens: Zum schnellen Auffinden von Objekten und volumetrischen Bildern, basierend auf lokalen Eigenschafften, wie z.B. Nachbarschaft, Überlappung, lokaler Bildinhalt und lokale Bildähnlichkeit, wird eine optimierte Indexstruktur entwickelt. Diese Datenstruktur erlaubt das indi-

zieren lokaler Eigenschaften dieser Objekt- und Bilddaten auf eine Speicherplatzeffiziente Art und Weise, die auf den räumlich kohärenten Zugriff optimiert ist.

Die präsentierten Projekte zeigen, dass die Herausforderung für das Erreichen von Interaktivität oft beim Entwickeln von Methoden liegt, die die Balance zwischen Berechnungsgeschwindigkeit und Qualität der Ergebnisse auf eine für den Benutzer optimale Weise herstellt. Ausserdem wird gezeigt, dass die Geschwindigkeit der Algorithmen nicht die einzige Eigenschaft ist, die für die Interaktivität verantwortlich ist. Die Präsentation der Ergebnisse, sowie effiziente Kontrollschnittstellen, sind als ähnlich wichtig zu erachten und tragen zur Qualität der interaktiven Benutzererfahrung bei.

Contents

CHAPTER 1

Introduction

Figure 1.1: "*The Anatomy Lesson of Dr Nicolaes Tulp*", Rembrandt, 1632

1.1 A Brief History on Volumetric Imaging and Tomography

The ability to look inside dense and opaque objects must have ever been promising and most desirable. Especially the knowledge about the inner structure and function of the human body remained a mystery for a long time. The dissection of human cadavers was until the end of the 19th century the only possibility to gain insight, but was also off-limits and often prohibited. This gap in knowledge could be systematically closed, at least since the age of enlightenment, when prohibition was eventually lifted. Figure 1.1 shows Dr Nicolaes Tulp at one of his famous anatomy lessons in Amsterdam surrounded by prominent spectators. The increasing knowledge of the human anatomy was accompanied by the progress of medical knowledge. Together the function of body and the effect of diseases could be explored and resolved.

The discovery of X-radiation, a kind of electromagnetic radiation, finally paved the way for modern medical imaging. X-rays with a wave length from 0.01 to 10 nanometers are able to penetrate through dense material while leaving an impression on photographic plates. Wilhelm Röntgen did not discover X-radiation but was the first who studied them systematically. In 1895 he published the first medical X-ray image "Hand with Rings" (Figure 1.2). The utility of the new technique was so obvious that the adoption into medical use took an extremely short time. For example the first medical X-ray image in America was published only three months after Röntgen's publication [125].

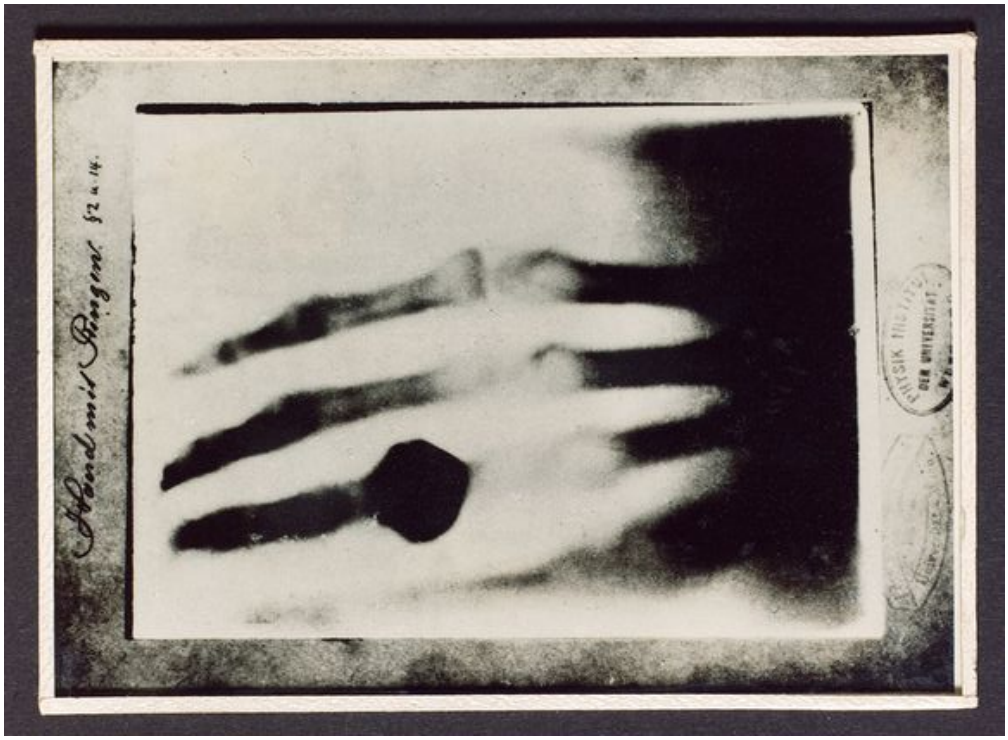


Figure 1.2: "*Hand mit Ringen*" (Hand with rings), Wilhelm Röntgen, 1895. First published X-ray image

.

Down to the present day X-ray imaging plays a major role in medical diagnosis. However due to its projective nature a X-ray image does not reveal the real three dimensional structure of the object. Important structures can be obstructed by dense objects such as bone. Furthermore depth and order of visible objects can not be evaluated without prior knowledge.

A method to reveal the real spatial structure of tissue and cell samples is histology. A tissue sample is fixed and sliced into thin layers. A slice can then be examined using a conventional light microscope. This method enabled for the first time a view inside a cell and lead to numerous discoveries in the 19th century. The restriction of producing only two dimensional cross sectional images can be overcome by acquiring sequences of slices through the sample (e.g. the "Mouse Connectome" project [28] uses this technique to produce a three dimensional brain atlas of the mouse).

In 1930 the Italian radiologist Alessandro Vallebona found a way to produce two dimensional cross section images from dense bodies without the need to physically harm them. The method utilizes X-ray imaging and requires that X-ray source and photographic plate is moved exactly contrary during exposure. In this way only one focal plane remains sharp while everything before and behind is blurred. The method was called X-ray Tomography. The word Tomography is a coinage assembled from the Greek words *tome* (cut) and *graphein* (to write).

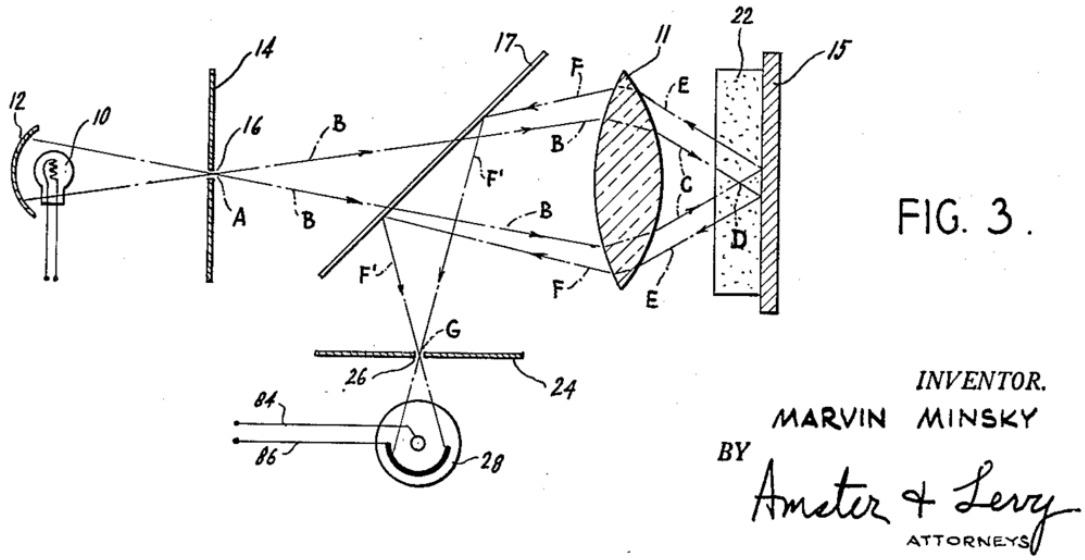


FIG. 3.

Figure 1.3: Patent drawing of the confocal microscope [83]. The light originating from the light source (10) is is guide through a semitransparent mirror (17) and the lens to the specimen (22). The pinhole (16) and the focus of the lens makes sure that only a very small region in the specimen is lit. The lit region therefore emits light because of fluorescence. This light travels back through the lens and is reflected by the mirror onto the photo detector (84). A second pinhole (26) removes scattered light which does not originate from the region of interest.

Small structures are often translucent enough so that visible light can be used to expose inner structures. However the projective nature of microscopic imaging results in the same limitations as X-ray imaging. A *confocal microscope* however is able to measure the emitted light from a single point in the specimen. This is done focusing the light source on exactly the point of interest. The specimen therefore emits light from this region which is directed to a photo detector. Scattered light is suppressed by a pinhole, at the focal point of the optical path, in front of the detector (Figure 1.3). Since only one point in space is illuminated at a time, the object has to be sampled on each discrete position sequentially [100]. *Confocal microscopy* was developed by Marvin Minsky and patented in 1957 [83]. Later the spatial resolution was improved by the use of a laser as light source.

Computed X-ray tomography (CT) finally solved the problem for larger and opaque objects. Godfrey Houndsfield created in 1971 the first praticable CT scanner. The scanner consists of a X-ray source, which is rotated around the sample and a photo detector, which is translated linearly on each source position to measure the source on different angles. Later the single detector was replaced by a linear array of detectors which improved scanning time dramatically (Figure 1.4).

Shortly afterwards a second modality was added to the collection of volumetric imaging techniques. In 1973 Paul C. Lauterbur developed together with Sir Peter Mansfield the first magnetic resonance imaging (MRI) device. MRI uses a strong static magnetic field to align the

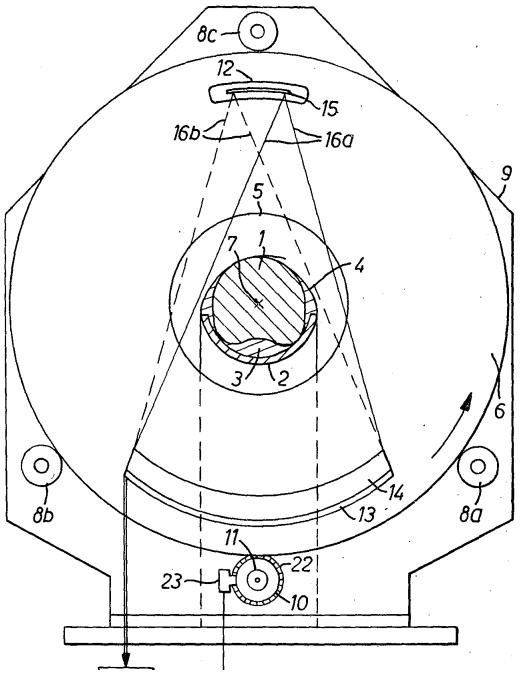


Figure 1.4: Drawing from the "Radiography" patent from G.N. Hounsfiled [47]. A radiation source (15) emits an X-ray beam which penetrates the patient (4). The detector (13) measures the attenuation of radiation. Source and detector are installed on a rotatable frame 6 which enables the exposure at different angles.

spin of the water molecules and a second varying magnetic field which is able to flip the spin of these molecules. The following re-alignment of the spin to the static field creates electromagnetic radiation which can be measured. Due to technological difficulties it took until 1985 for the first MRI scanners to be usable for medical diagnosis.

CT and MR imaging are by far the most important volumetric imaging techniques, especially in the medical field. However there is a large number of other volumetric imaging methods using different physical phenomena. For example sound waves above 20kHz (ultrasound) are used in medical imaging. Sound waves with longer wavelength can be used for seismic imaging. Radioactive decay is used in emission tomography (SPECT, PET). This requires that gamma rays emitting radionuclei is delivered into the patient which concentrates in structures of interest.

The relevance of volumetric imaging techniques is evident by the number of Nobel prizes awarded to the developers of these imaging techniques. The first Nobel prize in physics was assigned to Röntgen in 1901 [110]. In 1906 followed Camilo Golgi [35] and Santiago Ramon Cajal [17] with the Nobel prizes in physiology and medicine for their work "on the structure of the nervous system". Both scientists where histologists and received the award for the development of histology and staining methods which made the discovery possible. The inventors of the two most influential 3D imaging techniques received nobel prizes in 1979 "for the development of computer assisted tomography" [22, 48] and 2003 "for their discoveries concerning magnetic

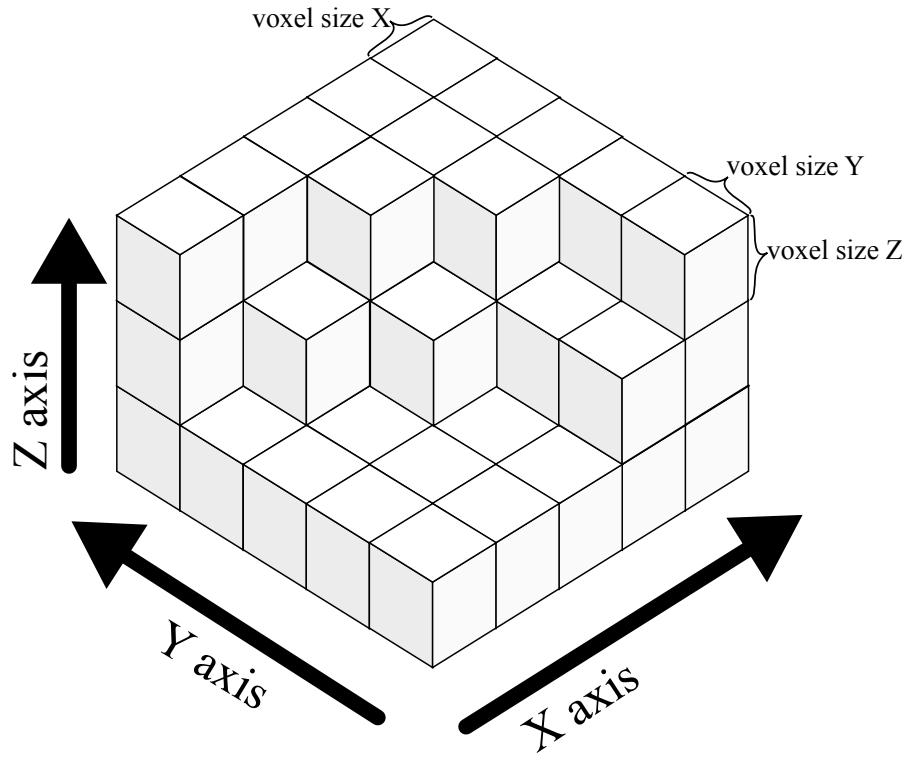


Figure 1.5: Volumetric image data:

resonance imaging" [68, 78].

Medical and biological interest was the main driver for the development of volumetric imaging methods. However, also other fields use this kind of data to a growing extent. Examples are seismic imaging and the use of industrial CT devices for quality control and non destructive testing.

At the current day we are confronted with a growing number of volumetric imaging modalities. At the same time, taking advantage of volumetric imaging gets more and more affordable. Therefore there is a growing need for methods which help to reveal the important information hidden in the volumetric image data.

1.2 Volumetric Images

Volumetric imaging techniques exploit physical phenomenons to measure a three dimensional field of a specific properties of the scanned object. This field is approximated by discrete sampling the volume and quantization of each value. For representation of volume data a variety of topologies are possible. Speray and Kennon [124] propose a taxonomy for this topologies ranging from *cartesian* and *regular* grids to *structured* and *unstructured* patterns. Irregular topologies are are often used to represent volumetric data created by simulation (e.g., *finite element method* (FEM)), or by non linear transformation of regular grids which for example originates

from volume image registration. However, usually volume data originated from imaging is represented by a regular topology consisting of evenly sized cuboids (Figure 1.5). Therefore the work presented in this thesis is restricted to regular volume representations.

The main axes of the volume are denoted with $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$. The discrete cells of the volume image are called *voxels* (volume picture elements). The resolution of a volumetric image is defined as the number of voxels along each axis. The volume is called *isotropic* if the distance between sample positions is equal for all dimensions, respectively if length, depth and height (size in $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ direction) of the voxels are equal. Otherwise it is called *anisotropic*.

In medical applications the three axes-parallel planes are named *traverse* (or *axial*), *sagittal*, and *coronal*. The traverse or axial plane divides the (human) body in the upper (*superior*) und lower (*inferior*) part. The sagittal plane divides the body in the left and right part. The coronal plane divides the body in the frontal (*ventral*) and rear (*dorsal*) part.

In the simplest case each voxel contains one scalar value. This is common for CT data where the value relates to density or MR data where the value relates to a specific spin relaxation property. However volume data is not necessarily limited to one scalar value per voxel:

- **Multi channel data**: Certain imaging methods acquire instantly multiple channels. For example confocal microscopy can capture multiple wave lengths (colors).

- **Tensor fields**: Diffusion MRI measures for each voxel the diffusion process of water as a 3×3 tensor.

- **Multi modal data**: An object can be scanned using different imaging modalities. Subsequently this images are registered to each other. As a result each voxel provides information created from multiple modalities. The quality of this information depends on the accuracy of the registration method.

- **Time series**: Some imaging techniques allow to capture image series in short intervals. This data can be used to analyze the motion of the heart. Time series with longer intervals are used to monitor changes in tumor growth. Therefore this modality is extensively used for intra-subject studies.

- **Data collections**: For intra- and more often inter-subject studies large image collections are created with the property that all images contain the same structure. If additionally all datasets are registered to the same frame of reference, the collection can be approached as a multi channel volume where each channel denotes the expression of a different dataset. This data representation is extensively used for statistical analysis and the creation of inter-subject atlases.

Derived Data

Volumetric image data is often the base for different derived datasets. These datasets can either be image data again, or some other but also spatial representation. These derived data have in common that they depict a special property of the original image data and live in the same space.

- **Derived 3D image data**: The original dataset can be transformed by image processing into various kinds of secondary image data. Image processing methods are used to enhance the original image data by the reduction of noise or removal of other artifacts originating from the imaging processing. Features of interest for example edges or vessel like structures can be detected using other image processing methods. The resulting image data contains for each voxel a score or probability for the existence of the specific feature at the position. First and second order image derivatives are computed through convolution with appropriate filter kernels. This results in higher order image data containing gradient vectors (first derivative) or hessian matrices (second derivative) .

- **3D structural data**: Based on the original image data, structures of interest can be extracted. Segmentation algorithms are used to define regions which belong to a specific area. This areas are described either by binary masks (binary image data with the same resolution as the input image) or by its boundary defined as triangulated mesh data. Vessel tracking algorithms extract location and shape of blood vessels or similar shaped structures. The information can subsequently be represent as skeleton graphs with radius information. Fiber tracking algorithms are used to estimate the process of axons in tensor field MR image data. This information is represented as bundles of three dimensional trace lines.

- **Semantic information**: Image data can be enriched with semantic information, either globally by describing the content of the image or locally in combination with extracted 3D structural information by describing what the structure shows.

- **Quantitative information**: Properties of the image can be quantified by measuring number, sizes, diameters or volume of specific structures.

- **High level information**: Based on overall analysis of derived information a high level statement concerning the image content can be made. In the medical field this can be the diagnosis of a disease which is visible on the image data.

Storage Formats

In the medical field image data is stored in the DICOM format [90]. DICOM defines the storage format as well as a network communication protocol. Furthermore meta information such as patient data, imaging properties and other additional data is handled through the same protocol. This provides compatibility between different storage systems and diagnosis workstations. Volume data is stored as a series of independent files. Each file contains one slice of the data. Meta information describe how the slices are spatially arranged.

In other areas no such standardization has been established. In general two different approaches are used. Either all slices of the volume are stored in separate files. This has the advantage that conventional 2D image formats can be used. However the drawback is that additional information such as sizes and slice order must be provided separately. Or the volume is stored as one piece in one file. This is for example possible with the *TIFF* file format which allows to store sets of images within one file. An extension to this is the *LSM* format, used by

confocal microscopes produced by *Carl Zeiss AG*, which stores additional data such as the size of the volume.

Another common example is the AmiraMesh format introduced by the Amira® [126] 3D visualization and and analysis software. This format is most flexible regarding data types and compression methods and stores the volume data as one piece.

1.3 Analysis and Exploration

Volumetric imaging is always used to depict some property of interest inside the sample. Therefore each volumetric image contains at least one important information which to extract was the purpose of producing this image in the first place.

Blank and Kalender [9] describe the process of "gaining insights" into (medical) volume data as a process consisting of two steps, analysis and exploration:

- **Exploration** of volume data is the task which lets the user get an overview over the data and locate region of interest.
- **Analysis** of volume data is the process which transforms image information into quantifiable results.

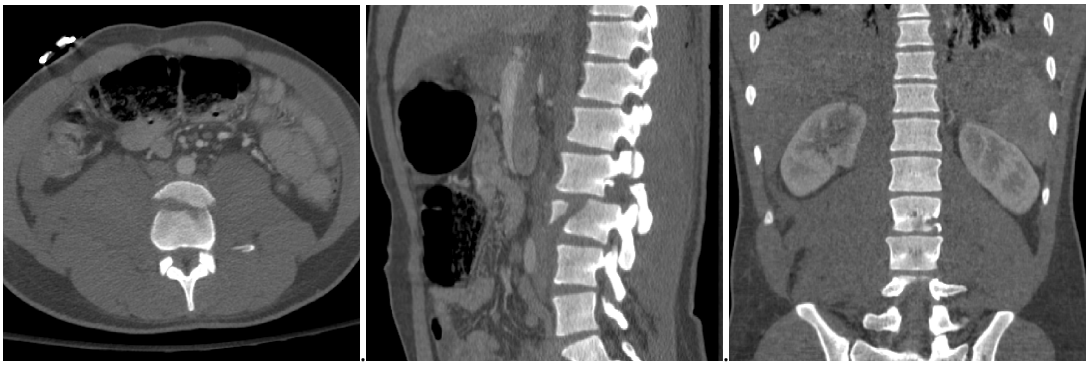
The purpose of any software system in this context is to improve and/or simplify the analysis and exploration task. Since tomography data is essentially image data defined in a three dimensional domain, analysis can be regarded as an image perception problem. Image perception can be performed by the user, automatically by the computer, or by a combination of these. A software system designed to support the analysis task can either try to present the data in a form which makes the task considerable easier for the human operator or aims to extract the important information directly. In reality often a mix of both methods is used.

Exploration through Visualization

Approaches which support the human operator by visually exploring the volume data can be subsumed as visualization systems. The purpose of such systems is to display the volumetric image data in a way that the perception of the important structures is as simple as possible.

The first and most natural form is to display the axial slices (Figure 1.6(a)) of the image stack. In the medical field tomography data often get printed out on transparent foil which can be hanged on a light box. In cases where the distance between slices is high axial reconstruction might be the only viable solution since cross sections, which intersect different slices, would contain visible artifacts. However the problem with this two dimensional representation of a three dimensional dataset is that locality in z-direction is broken, which leads to a poor representation of spatial context. Therefore the user has to be well trained to compensate with spatial imaginative power.

This limitation can partly be overcome by presenting additional sections with different orientations. These so-called "multiplanar reformations" (MPR) [49] (Figure 1.6(a-c)) add sagittal and coronal sections to the visualization. Furthermore state of the art visualization systems



(a) . (b) . (c)

Figure 1.6: MPR visualization of a CT dataset. (a) traverse or axial section, (b) sagital section, (c) coronal section.

do not limit the reformatted sections to be axis aligned which allows to adjust the view to the specific structure.

One drawback of this method is that not all structures can be adequately visualized by a planar section. Especially in the medical field curved structures, such as blood vessels, are of great interest. For this cases a "curved planar reformation" (CPR) [53] might be suitable which reconstructs a planar image along a curved path.

Visualization of volume data by rendering of cross sections has the disadvantage that it is difficult to get an overview over the whole dataset. Data which is in front or behind the section is completely omitted. Even the reconstruction of multiple orthogonal sectional views can not reveal the whole dataset at a glance. *Maximum intensity projection* (MIP) is a visualization method which aims to display the whole dataset at once. For this the three dimensional volume is projected on a two dimensional image plane. From all values which are projected to the same pixel on the image plane, only the highest is retained. This results in a two dimensional representation of the dataset which exposes especially high intensity features. The drawback in this approach is that structures recorded with lower intensities might be shadowed by higher intensity structures. The application of this method is therefore favourable in situations where the structure of interest is represented by high values (e.g., MR or CT angiography [30]). Furthermore the method does not retain depth information.

Shaded surface rendering is a basic 3D rendering technique which can also be used for rendering of volumetric images. In this case the image data is represented by its isosurface for a specific isovalue. In order to render the isosurface using common rasterization methods the surface must be extracted and transformed into a triangulated representation. A fundamental method for this is the *marching cubes* algorithm presented by Lorensen and Cline in 1987 [74] (Figure 1.7).

In opposite to this indirect approach, *direct volume rendering* (DVR) aims to compute the rendering directly from the volume data. Generally the goal of direct approaches is to evaluate each light ray which determines the color of a pixel on the screen [80]. This can be done in object-order (*splatting* [138], *shear warp* [64], *texture mapping* [45]), or in image-order (*ray*

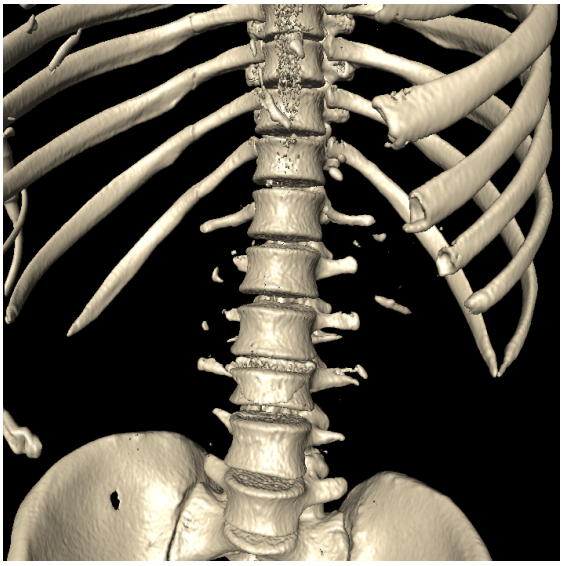


Figure 1.7: Isosurface rendering of a CT dataset.

casting [104]).

State-of-the-art volume rendering frameworks use image-order *ray casting* and utilize graphics accelerator hardware to achieve real-time performance. The methods operate by sampling each viewing ray, beginning from the virtual camera through the volume. While visiting the voxels of the volume, the final color of the corresponding pixel is computed. With this approach many different rendering methods can be implemented. For example, isosurfaces are rendered by *first-hit ray casting* [5, 42, 70].

Volumetric datasets often contain more than one material which should be displayed at the same time while providing enough contrast to distinguish between the materials. The definition of materials and colors is done using a *transfer function* which maps each scalar value to a specific colour and opacity. The final appearance of a pixel is computed by accumulating color and opacity along the ray until the result is opaque or the ray exits the volume. Additionally each sampled color can be modified by a shading function to simulate a lightning effect.

The quality and expressiveness of the rendering depends completely on the transfer function. Therefore a lot of research was done to improve this part, either by supporting or automating the design of transfer functions [32, 34, 59], by using different properties of the scalar field for the mapping (e.g. curvature [46]), or by providing multi dimensional transfer functions [60].

An important extension is the so-called *multi volume rendering* [8, 37, 88] which enables to render several volumetric datasets at the same time. Assuming that the datasets are co-registered, the overlay of visual informations originated from different modalities can effectively increase expressiveness of the produced visualization.

Which visualization technique performs best depends on the analysis task as well as on the kind of volume data. Often it is advantageous to combine different techniques to benefit from the different visualization properties. Sets of visualizations whose navigation is coupled together

are called *multiple coordinated views* [106, 136] A related approach is the *focus+context* [95] method which tries to combine a detailed visualization of the structure of interest with a detail less visualization for the surrounding context.

Volume Analysis

Volume analysis describes the process where specific features and properties of the data are revealed and quantified. Obvious analysis tasks, which have to be performed on volumetric data, are localization, segmentation, and measurement. The software can support these tasks by providing specialized tools. Furthermore it is often requested to automate this task to reduce the workload of human resources. Besides this, there are tasks where computed analysis is simpler and more successful or even the only possible option.

Localization of objects, structures or landmarks is a basic building block of image analysis algorithms. Wang et al. [135] localize, intentionally on the skin of the patient placed, markers from images of different modalities. The location information is subsequently used to initialize an image registration process to fuse these images.

Methods for recognition of arbitrary feature points are the widely used SIFT algorithm [75, 76] and its extension PCA-SIFT [56]. These methods are designed for 2D image processing but can be extended for 3D volumetric images [3, 120]. Often it is required to locate larger and more complex structures. This can be addressed by describing objects as a combination of multiple, spatially related feature points. Therefore Donner et al. [29] propose a method for object recognition based on a larger number of feature points combined by a *Markov Random Field* (MRF).

Other methods approach the localization problem applying knowledge about the expected shape. A good example for this is the the *Hough transform* for circles [23] which can be used to locate the left ventricle in cardiac MRI scans [103, 143].

Zheng et al. [144] present a method which uses a machine learning based approach (*marginal space learning*) and *steerable features* to locate the heart and initialize a following segmentation step.

Segmentation describes the task where the shape of a specific structure is extracted from the image data. Often it is difficult to recognize the shape of a structure if it is embedded in the volumetric image data. Sectional visualization methods are naturally limited in terms of three dimensional representation of the structure, but also three dimensional rendering methods have difficulties to visualize exactly the structure of interest. Separating the region of the object from the background enables a separate analysis of the structure of interest. Therefore the knowledge of the exact shape makes analysis of this property possible and makes it usable for further applications such as measurement.

Segmentation is often done manually based on cross-sectional images of the dataset. Software systems provide usually a range of tools to allow the definition of a specific area such as brushes or polygonal selection tools.

However manual segmentation is a time consuming task, which causes the wish for automatic or at least semi-automatic approaches. Segmentation algorithms basically produce a separation of the image in background and foreground regions based on image features and can

be controlled by algorithm parameters. One of the most basic segmentation algorithms is called *thresholding*. It separates the image based on the voxel intensity value. Only the threshold parameter is adjustable.

Since the segmentation problem is manifold, a great number of segmentation algorithms have been developed. These algorithms can be separated into two classes, those who use prior knowledge about the expected shape and those who do not. Prior knowledge agnostic methods are for example *watershed* [133], *region-growing* [1], *active contours* [54] and *level sets* [97,98]. These algorithms compute a segmentation only on a set of algorithm parameters.

In contrast so-called *model-based* algorithms have prior knowledge about the shape of the targeted structure. The shape as well as the possible variation in shape is learned from a set of manually labeled training data. Examples for this are *active shape models* (ASM) [21] and *active appearance models* (AAM) [20] which also take the variation of the underlying image data into account.

Measurement describes the process which extracts reliable analysis relevant numbers from the volumetric dataset. In medical applications the measurement of geometrical sizes such as diameter, thickness, volume, distance and angles are of great interest. Based on these numbers pathologies can be systematically classified. Examples for this are classification of scoliosis by measuring vertebral rotations [123] or the volume quantification of multiple sclerosis lesions [84].

Another important tool is the measurement of tissue properties. The appearance of tissue can give information of pathological states. For example the quality of tubercular bone micro architecture can be quantized by texture analysis methods [137] the results relate to the diagnosis of osteoporosis.

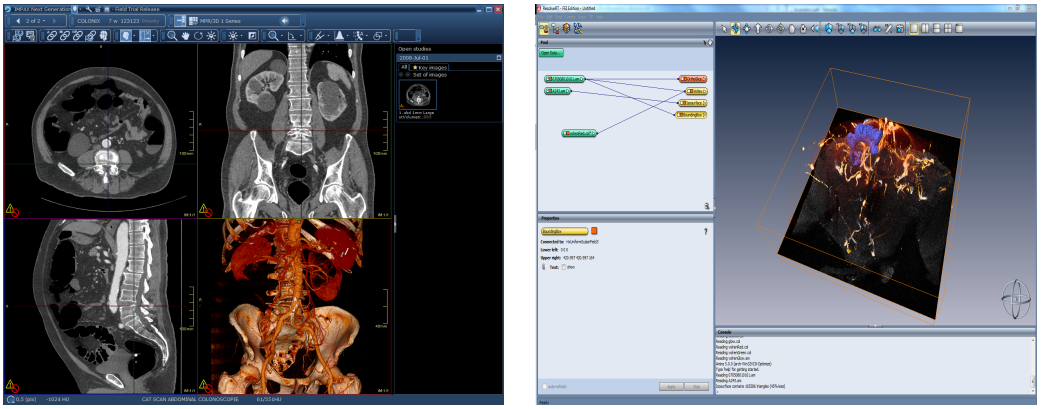
Industrial CT is another emerging field which uses volumetric imaging techniques for non destructive testing purposes. Important tasks in this field are quantification of material defects, automatic true to size assessment and the comparison of extracted shapes to CAD models [105].

1.4 Software Solutions for Volume Data Analysis and Exploration

The possibilities for exploration and analysis of volumetric datasets without the support of software systems is very limited. In the medical field it is still usual to use printed sheets with axial slices. This has primarily practical reasons, interactive visualization systems do not fit well into the sterile environment of an operation room. However in more complex cases exploration, diagnosis and planning has been performed beforehand.

In the medical field workstations for image analysis are part of the *picture archiving and communication system* (PACS) which provides access to the image archive through the DICOM protocol (Figure 1.8(a)). This standardized foundation allows software from different vendors to access the same data provided by a common archive. Examples for this kind of software are *IMAPX EE* from Agfa Healthcare, *syngo®.plaza* from Siemens and *IntelliSpace* from Philips.

A special case of medical visualization systems are navigation systems for *computer assisted interventions* (CAI). These systems are able to track surgical instruments and relate them to previously acquired image data of the patient. Examples for this are the *StealthStation®* products from Medtronic or the *ENT* system from Brainlab®.



(a) (b)

Figure 1.8: (a) PACS image viewer providing MPR and 3D volume rendering views. (b) Amira® visualization software using one combined 3D view showing volume rendering together with a volume slice and geometry.

A general purpose solution for volume data visualization is *Amira*® [126] (Figure 1.8(b)). It provides advanced functionality for volume data visualization intermixed with classic geometry based rendering. A plugin interface makes it adaptable to almost any volume analysis task.

Other examples are *ImageJ* [112] and *MeVisLab* [11]. Both have the focus on a very adaptable image (pre-)processing with integrated visualization capabilities. However both solutions are rather frameworks than actual applications which enable a rapid creation of analysis and exploration tools.

Except of the computer assisted interventions systems, which have a very well defined use case, most solutions are multipurpose systems. They are designed to process and visualize any kind of (medical) volume data but provide only general analysis tools. Different strategies exist to realize systems supporting special tasks and workflows.

- The development of specialized single purpose applications. CAI system often fall into this category since they combine software with a hardware setup (cart and navigation system) which is build for a specific type of intervention.

- General purpose diagnosis or visualization systems often utilize plugin systems for extendability. Through plugins special purpose analysis tools can be easily added to an existing installation. In some cases the plugin interfaces are open which enables users to develop extensions tailored to the own needs.

- Some frameworks offer scripting interfaces which allow the user to automatize the processing task and to initialize the visualization. A special kind of this is visual scripting as used in *MeVisLab* and *Amira*® which allow to define data processing and flow within a graph representation.

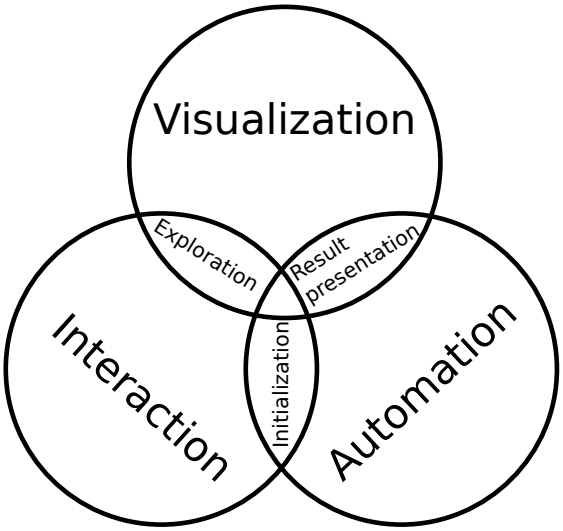


Figure 1.9: The diagram visualizes the relationship between the three main components of an application for volume data analysis and exploration.

Key Capabilities

Systems for volume data analysis and exploration provide three key capabilities: visualization, interaction and automation. All applications offer this capabilities to a varying extend. Active support for a specific analysis or exploration task manifests itself in the combination of these three capabilities:

1. **Visualization:** The visualization system is the key feature which enables the user to explore the data. Visualization software is responsible for presenting the dataset in a way which optimally exposes the feature of interest. Which kind of visualization method can be used depends entirely on the specific problem.

2. **Interaction:** User interaction defines how the user controls the visualization and the analysis tool. Well designed interaction patterns can improve speed, quality, and efficiency of the exploration and analysis task.

3. **Automation:** Automation is used to reduce the the manual workload of the analysis task. Automated analysis can in specific cases supersede manual interaction. In most cases however it is used to simplify the task and support the user.

The three components are not independent. In fact the combination of and the interface between these components governs the usability of the application. Figure 1.9 depicts these relationships. In this model exploration is located at the interface between visualization and interaction. Efficient exploration relies on a suitable visual representation of the dataset, but becomes only effective if the user interface allows efficient control. Automated methods interface with the interaction because these algorithm often need to be manually initialized with

parameters or seed points, or at least need to be started. The finally computed results have to be presented by the visualization system because in the end the user has to confirm, reject or adjust the result and needs to be supported by an effective presentation of those.

Problem Area

The design of algorithms, tools and applications for analysis and exploration tasks has to fulfill at least two requirement. First, the approach needs to result in a viable and correct solution. Second, the approach needs to fit into the environment and workflow. While the first requirement is obvious, the second requirement can become a serious problem as soon as the method should be applied under real-world conditions.

The projects presented in this thesis were all governed by this *real-world requirement*. Therefore it is particularly interesting to investigate how the environmental factors influence the design of this software of applications. To answer this question the problem area is constructed by the four dimensions interaction, time, quantity and quality:

- **Interaction:** The analysis task can be solved in many ways ranging from completely manual to fully automatic. Manual methods rely on the right tool set in combination with a visualization system which effectively reveals the features of interest and an effective user interface. Semi-automatic or supervised approaches need a good interface for initialization, efficient algorithms which provide fast feedback and the possibility to either re-parametrize the algorithm or to manually adjust the result if it is not acceptable for the user. Whereas fully automatic solutions must be able to produce results with high accuracy and low failure rate.

- **Time:** Algorithms, which support analysis and exploration tasks often have hard time constraints. The visualization system is expected to react in real time and semi-automatic analysis methods need to provide feedback in an acceptable time interval (usually not more than a couple of seconds). Otherwise interactive browsing, exploring, and analyzing the volume gets inefficient and unattractive. Fully automatic computations might not have that hard time constraints since they can be performed offline beforehand or the user can get back to the result some time later. In this case the time bound is defined by computational resources and the number of datasets which have to be processed.

- **Quantity:** Quantity refers to the amount of data which has to be processed for the task. Properties like size of the volume, number of channels, number of time steps or number of datasets determine memory consumption. Based on quantity it must be decided if the algorithm can perform on data located in RAM or if out of core methods have to be performed. Also quantity makes a huge difference in execution time. Therefore this property has a direct influence on the time constraint.

- **Quality:** The demand for highest quality of computed analysis results or visualizations is often in direct contradiction to any of the former properties. Little advances in quality or accuracy are often accompanied by a major growth of the execution time and eventually more memory consumption or require much more interaction.

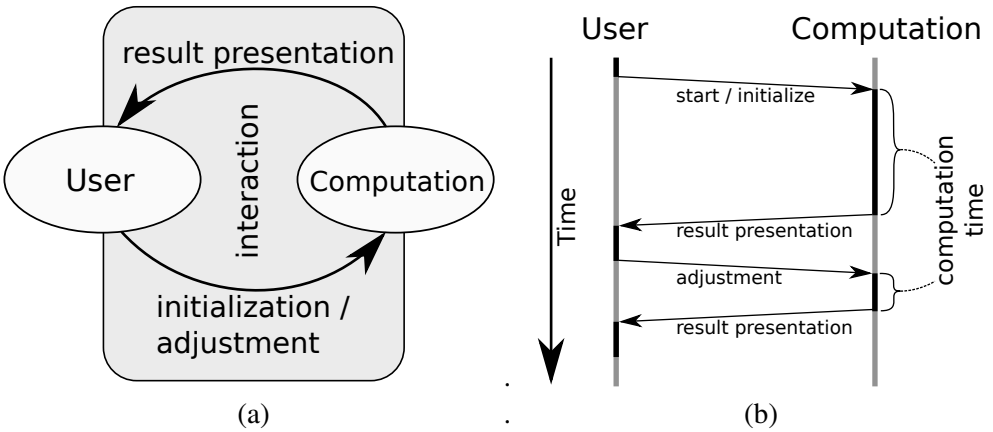


Figure 1.10: Interaction pattern in interactive applications.

The previous discussion shows that the four dimensions of the problem area are tightly coupled. Changing one requirement has always a direct effect on some of the other dimensions. Optimal solutions for volume data analysis and exploration tasks require to find the right balance between these properties.

Interactivity

Interactivity plays a major role because analysis and exploration is a user driven process. The user starts the process, actively issues adjustments and finally finishes the process if he finds the result acceptable or found the insight he was looking for. Therefore the center of concern for interactive applications is the time requirement, defined in the previous section.

The concept of the interaction loop is depicted in Figure 1.10 (a). In this illustration it becomes obvious that the depicted process alternates between user and computation. From the user point of view the latency induced to by computation has a major impact on interactivity. Figure 1.10 (b) depicts the time dependent interaction between user and algorithm. Therefore a characteristic property for the type of interactivity is the computation time.

Based on the computation time, which introduces latency into each loop, different characteristic interaction patterns can be defined.

- **Real time interaction**: The latency introduced by the computation is very low which enables multiple iterations per second. In user perception the system reacts instantly and without any delay. This kind of time complexity is needed if the interaction loop is iterated very often respectively if a high number of adjustments is necessary to produce the expected result. Examples for this are any kind of visualization system where the user interactively navigates the view.

- **Near real time interaction**: Computation in this class of interactivity takes considerably longer but just as long as the user is willing to wait. If the introduced delay is acceptable

depends on the quality of the result. Algorithms which produce better results need fewer adjustments and therefore fewer iterations until the user is satisfied. In this case the user can accept a longer delay. Lower quality must be compensated with more guidance which requires a faster turn around to be efficient.

- **Offline interaction**: If the computation time gets to high the system is no longer interactive. In this case it is not possible to perform the task in a serial way. Instead the computation will be issued and the result will be examined at a later point of time. This imposes high requirements on the quality of the result since adjustment and re-computation is not desired.

These basic interaction patterns depend on suitable run time and quality characteristics of the used algorithms. However not in all cases these requirements can be fulfilled, often the complexity of the problem is too high to find suitable solutions. Therefore different strategies are possible to break the monolithic problem into several smaller blocks which are simpler to solve.

- **Multi stage processing**: The problem is divided into several smaller problems where the time and quality requirements are easier to meet. For example a segmentation system can first use an automatic method to roughly segment the object. The missing segmentation quality is then improved in a second step by a semi automatic method or manual adjustments.

- **Pre-processing/indexing**: The problem is divided into a pre-processing or indexing stage, which is performed offline and an analysis and exploration stage which allows at least near realtime interactivity. The first stage is responsible for the creation of certain structures which then can be used in the second step to accelerate the computation so that the requirements are met. This method is used in any kind of retrieval system, where first an index is build from a large collection of datasets. This index is later used to find entries without the need to inspect the whole datasets.

- **Latency hiding**: In some cases the latency of an algorithm can be hidden from the user by delivering a fast partial result concerning the portion of the data which is in the focus of the user, and subsequently complete the result after a longer delay. This is for example a common practice in volumetric image processing. An expensive image filter is first applied to the data which is visible to the user (e.g., a planar section of the volume) and the remaining part of the volume is updated.

Algorithms which enable interactivity in analysis and exploration of volumetric image data must balance two requirements. First, the introduced latency must be small enough so that the system remains interactive. Second, the quality must be high enough to limit the number of interactive iterations to a minimum.

Additionally there are other parameters which greatly affect interactivity. The user interface is responsible for providing the algorithms with initializations and adjustments. More precise control over this parameters can improve the outcome. Therefore simpler and more effective user interfaces can improve this task. Furthermore the presentation of computed results determines

how the user perceives the solution. Adequate presentation can therefore simplify the evaluation of the result which can lead to more effective user interactions.

1.5 Structure of the Thesis

In this thesis four research project are presented which use different methods to enable interactivity in applications for volumetric image analysis and exploration.

Chapter 2 presents a system for intra-operative virtual endoscopy. The main goal of this system is to provide a meaningful visualization during all stages of the intervention. However an intra-operative setting requires to limit the user interaction to a minimum. The system uses intra-operative navigation to interactively control the system while meaningful visualizations are provided using robust rendering methods.

Chapter 3 discusses a method for feature localization on CT images of the spine. The algorithm was designed as a building block for a larger automated spine labeling tool and had to satisfy hard run time and memory consumption requirements. This chapter documents the effort of finding a viable trade off between speed and accuracy while keeping the algorithm interactively usable on middle class workstations.

Chapter 4 describes a similarity based retrieval method for neuronal structures. A large collection of Drosophila fly brain volume images with annotated neurons is the basis for extensive exploration which should be supported by interactive retrieval methods for the annotated structures. A main consideration for the implementation was the huge amount of data as well as the quality of the results. Fast retrieval feedback is achieved by using adequate indexing structures.

Chapter 5 presents a system for fast spatial lookup of image and object data. It works on a large collection of co-registered bio-image data and allows the user to interactively explore the image and object space by spatial distances and local image content. Designed to be used for active exploration, the system is expected to deliver results in a short period of time. Furthermore the amount of data prohibits the use of in memory index structures.

Chapter 6 concludes the thesis by summarizing the presented work and by a categorization of the presented strategies for enabling interactivity.

CHAPTER 2

Intra-Operative Virtual Endoscopy for Image Guided Endonasal Transsphenoidal Pituitary Surgery

This chapter is based on the publications:

> Florian Schulze, Katja Bühler, Andre Neubauer, Leslie Holton, and Stefan Wolfsberger. *Intraoperative navigation virtueller endoskopie bei endoskopischen transsphenoidalen eingriffen an der hypophyse.* In Proceedings of the Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC), 2008. [114]

and

> Florian Schulze, Katja Bühler, Andre Neubauer, Armin Kanitsar, Leslie Holton, and Stefan Wolfsberger. *Intraoperative virtual endoscopy for image guided endonasal transsphenoidal pituitary surgery.* International Journal of Computer Assisted Radiology and Surgery, 5(2):143–154, 2010. [115]

Virtual endoscopy has already proven its benefit for pre-operative planning of endoscopic pituitary surgery. The translation of such a system into the operating room is a logical consequence, but only a few general intra-operative image guided systems providing virtual endoscopic images have been proposed so far. A discussion of related visualization and interaction problems occurring during sinus and pituitary surgery is still missing.
This paper aims at filling this gap and proposes a system that integrates an existing virtual endoscopy system originally designed for pre-operative planning of pituitary surgery with a professional intra-operative navigation system. Visualization and interaction possibilities of the pre-operative planning system have been extended to fulfill the special requirements to the system if used for intra-operative navigation of endonasal transsphenoidal pituitary surgery.
The feasibility of the system has been successfully tested on one cadaver and 12 patients. The virtual endoscopic images were found useful (1) during the endonasal transsphenoidal approach in cases of anatomic variations and for the individually tailored opening of the sellar floor, and (2) during tumor resection for respecting the internal carotid artery. The visualization of hidden anatomical structures behind the bony walls of the sphenoid sinus during the sellar phase of the surgery has been found most beneficial.
According to our data, intra-operative virtual endoscopy provides additional anatomical information to the surgeon. By depicting individual anatomical variations in advance, it may add to the safety of this frequent neurosurgical procedure.

2.1 Introduction

Endoscopic surgery of the pituitary gland is a well established minimally invasive technique for the treatment of pituitary adenomas. The endonasal transsphenoidal approach using a rigid endoscope [27] (see Figure 2.1) requires a high experience level of the surgeon due to the limited field of view of the endoscope and the complex patient individual anatomical structure of the paranasal sinuses that make orientation difficult. The presence of critical structures that might not be harmed under any circumstances in close proximity to the adenoma, like the brain, optical nerves, eyeballs, internal carotid arteries and the pituitary gland itself, induce an additional level of complexity to the procedure.

Careful planning of the intervention based on pre-operative CT and/or MR data of the patient helps to reduce the risk during the intervention by identifying pathologies, anatomical variations and critical structures. Basic planning systems provide tools for image registration, segmentation, and MPR (Multi-Planar Reformation) viewing. Additional computational support is provided by virtual endoscopic views mimicking the real endoscopic image based on pre-operative image data, and a camera model reproducing the optical properties of the endoscope. Virtual endoscopy allows pre-operative simulation of the approach by virtual navigation through the paranasal sinuses and additional visualization of hidden critical structures supporting the surgeon to decide pre-operatively on the optimal surgical strategy.

The second option for risk reduction is the use of intra-operative navigation systems provid-

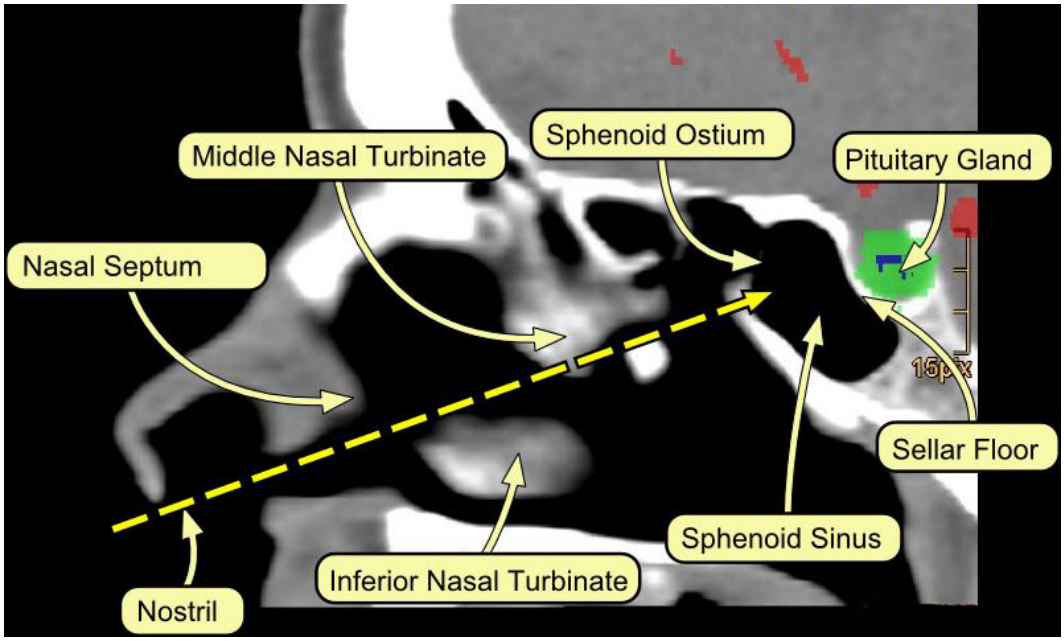


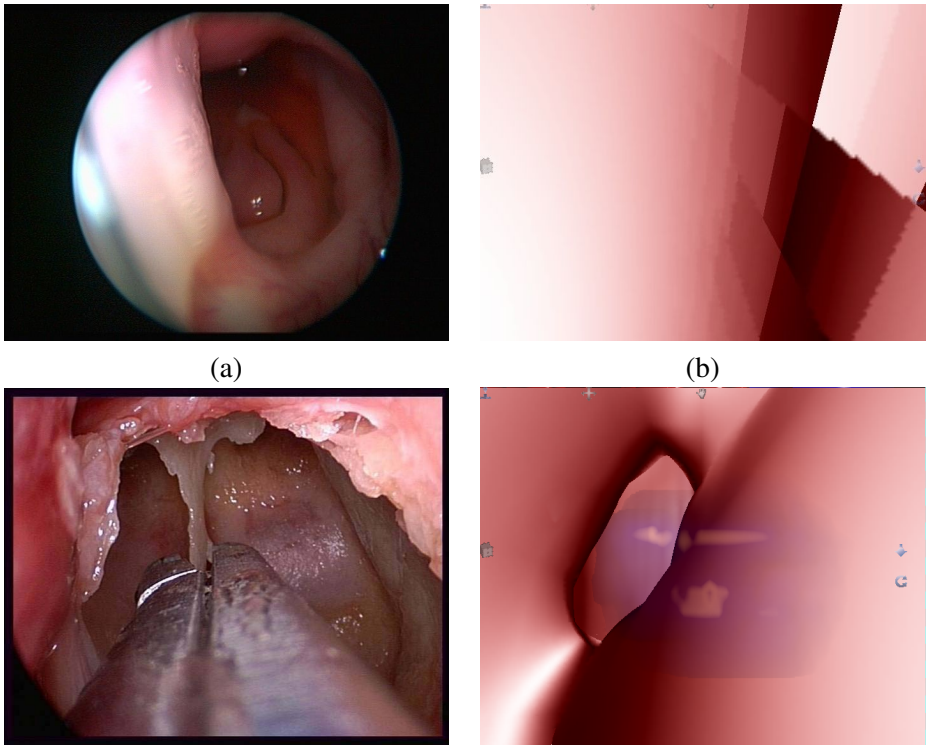
Figure 2.1: The endonasal transsphenoidal approach: The endoscope is inserted from the nostril towards the sphenoid sinus passing the inferior and middle nasal turbinates and the sphenoid ostium.

ing reliable information on the actual location of the tip of the endoscope and other instruments in relation to the patient based on pre-operative CT and/or MR data. These systems generally provide tracked MPR views, display of segmented structures, markers indicating an available pre-operative plan and eventually a non-perspective 3D view of the data set. However, none of the listed conventional visualization methods allows the direct visual comparison of endoscopic image and pre-operative data.

The combination of intra-operative navigation with virtual endoscopy is a logical consequence from this observation. The main benefit of the connection of tracking information with virtual endoscopic images is the possibility to establish a direct local relation of virtual and real endoscopic images and to visualize the pre-operatively defined critical structures not visible in the real image. Especially in non-standard cases this additional information can support the decision process of the surgeon during the procedure.

While potential benefits of intra-operative virtual endoscopy for pituitary surgery are obvious, and a high correspondence of virtual and real endoscopic images has been reported [108], virtual endoscopy has not yet been established for endonasal approaches in the operating room (OR).

Careful analysis of existing virtual endoscopy systems reveals a possible explanation for this observation: General virtual endoscopy systems provide images based on the given pre-operative data without taking the specific approach and the current intra-operative situation into account. In practice, such systems often fail to generate stable and meaningful virtual images during all



(a) (b)

(c) (d)

Figure 2.2: Typical situations in endonasal pituitary surgery where corresponding real and virtual endoscopic images are not comparable.

phases of endonasal transsphenoidal approaches which substantially limits the usefulness and acceptance of the system.

The main reason for these visualization problems are the narrow cavities the endoscope has to cross and slight differences between the available pre-operative images and the current intra-operative situation.

These differences can be caused by inaccurate tracking information induced by a slight flexibility of the endoscope, but they are present even if patient and pre-operative data are perfectly registered and the position of real and virtual endoscope match with high precision:

- During the nasal phase of the surgery tissue might be slightly deformed in comparison to the pre-operative data: it might be more or less swollen, compressed or shifted by the inserted instruments (Figure 2.2(a)).

- During the sphenoid phase structures are less deformable and mainly modified by resection: the sphenoid ostium is opened, septa and bone are removed (Figure 2.2(c)).

The first situation might result in a complete loss of sight in the virtual endoscopic image if the virtual camera is located inside virtual tissue while the real endoscope shows a clear image

of the nasal pathway (Figure 2.2(b)). In the second situation the removed tissue is still present in the pre-operative data and occludes in the virtual image the sight on structures already visible in the real endoscopic image (Figure 2.2(d)).

This paper proposes a virtual endsocopy system specialized on the intra-operative guidance of endonasal transsphenoidal approaches to the pituitary gland. The system is based on an existing planning system for endonasal transsphenoidal pituitary surgery and a professional intra-operative navigation system. Thus, this work focusses mainly on the intra-operative use of the virtual endoscopy system. The approach-specific core-visualization problems described above are addressed and several additional features facilitating the intra-operative use of the system are proposed. The feasibility and benefits of the extended virtual endoscopy system have been assessed and are discussed based on 12 clinical cases.

2.2 Related Work

An excellent general introduction to virtual endoscopy, including an overview on different rendering techniques, academic and commercial systems, and application fields has been published by Bartz [6].

The first proposed virtual endoscopy systems for visualization of paranasal sinuses based on patient image data used specialized hardware [140] or generated off-line fly-throughs along a pre-defined path [108] due to the limited computational power. The visualization of more recent systems is based on highly optimized software [92] or GPU based [63] raycasting algorithms that allow to add more realism, like realistic lightening or wetness, without loosing interactivity. The dominating render mode today is direct iso-surface rendering, a techniques that directly generates a virtual endoscopic image based on individual patient data without pre-processing of the images and generation of intermediate geometry. Variations of this technique are spatially restricted direct volume rendering [92] and combinations of direct iso-surface and direct volume rendering. The geometric representation of the paranasal sinuses plays a secondary role in the pre- and intra-operative setting as exact segmentation of the paranasal sinuses in computed tomography images is an extremely time consuming task [111, 128], and flexibility and quality of the direct representation is higher. A comparison of direct and indirect rendering techniques of paranasal sinuses has been given by Rogalla [107].

In addition to the correct simulation of the endoscopic view, a clear visualization of critical structures as background objects is the most important feature for pre-operative planning systems for sinus and pituitary surgery and is part of most cited systems. Other described features of planning systems include linked MPR views, navigation aids, annotation possibilities and PACS (Picture Archiving and Communication System) integration. A detailed discussion of virtual endoscopy for paranasal sinuses has been recently published by Kettenbach et al. [58].

Intra-operative guidance of endoscopic procedures has already been proposed with the availability of the first intra-operative navigation systems. For sinus and pituitary surgery different strategies to fuse pre-operative data and intra-operative images have been published: fusion of real endoscopic images with outlines [26, 94] or 3D representations of pre-segmented objects [55]; the combination of non-perspective 3D rendering of the whole head with cutting planes following the tip of the endoscope and visualization of the viewing frustum [141]; record-

ing and registration of intra-operative images to provide reconstructed views in case of total loss of sight due to hemorrhage [61,113]. Systems using virtual endoscopic images directly for intra-operative guidance of sinus surgery are rare. Lapeer et al. [67] proposed a system, that enhances a real endoscopic image directly by fusing it with the corresponding virtual view. Deformation of tissue and the related discrepancy between real and endoscopic images has been discussed only in connection with endoscopy simulation systems for training purposes [140].

Mayberg et al. [81] describe the use of the commercially available intra-operative navigation system Cbyon IGS for image guided pituitary surgery. Cbyon IGS provides parallel display of virtual and real endoscopic images. The system is designed for the intracranial approach and uses a direct volume rendering method which is not capable of visualizing surface anatomy of nasal cavities.

To the best of our knowledge there exists no paper discussing the complete set-up of a virtual endoscopy system for intra-operative guidance of endonasal pituitary surgery including a detailed discussion of related visualization, interaction, and hardware problems.

2.3 Materials

This section shortly describes the two systems that have been combined, adapted and extended to build navSTEPS - the system for virtual endoscopy based image guided endonasal pituitary surgery being subject of this chapter.

STEPS - Pre-operative Planning of Endonasal Transsphenoidal Pituitary Surgery

STEPS [92, 139] has been developed for pre-operative planning and simulation of endonasal transsphenoidal pituitary surgery. It provides the software basis of navSTEPS: Surgery planning and pre-operative exploration of patient data is still done using the original system, whereas additional visualization and interaction paradigms have been developed for its intra-operative use.

STEPS has been implemented as PlugIn of a professional radiological workstation (ImpaxEE, AGFA Healthcare) inherently providing PACS integration, slice and MPR views of image data, and professional registration and segmentation tools. It covers the whole planning workflow including automatic and semi-automatic mutual information based rigid registration of available MR and CT data; manual and semi-automatic segmentation of critical structures on MR and/or CT data; interactive data exploration based on enhanced MPR views and a virtual endoscopic view of nasal cavities and paranasal sinuses including background visualization of pre-segmented structures; data annotation; planning of the optimal access to the sphenoid sinus and the location for the opening of the sellar floor (Figure 2.3).

The virtual endoscopic view is rendered directly and in real time from the CT data, i.e., no segmentation of the nasal cavities and paranasal sinuses and generation of intermediate geometry is required. Two different render modes are available: A direct first-hit iso-surface rendering based on a pre-defined, but adjustable, iso-value that displays the shape of the rendered cavities, and a constrained direct volume rendering that allows the direct visualization of tissue characteristics (i.e., the difference between mucosa and bone) encoded in the CT image. The fusion of

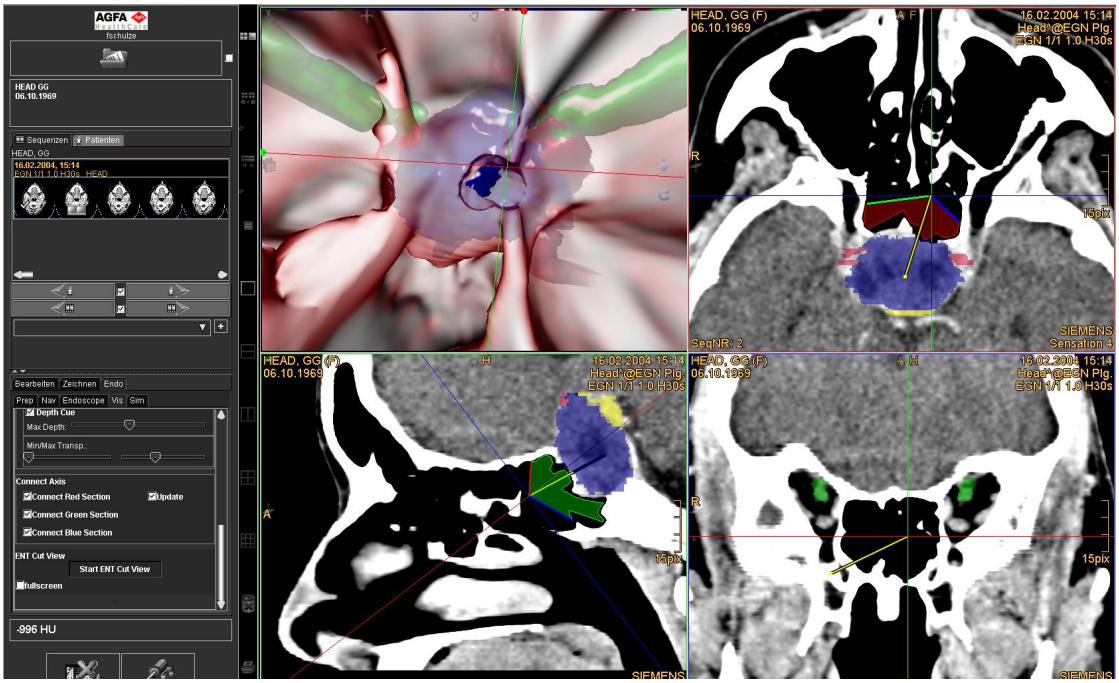


Figure 2.3: Screenshot of STEPS showing the planning and simulation of the opening of the sellar floor. The tumor and the internal carotid arteries are shown as background objects in the virtual view and appear color coded in the MPR. The tip of the virtual endoscope is located at the center of the MPR, a projection of the line of sight is indicated on each section by a yellow arrow. The color coded frustrum on the MPRs restricted by the rendered iso-surface provides a tool to directly compare the shown virtual endoscopic image with the original data.

virtual endoscopic views and rendered critical structures shown as background objects is done on the fly and provides highly interactive and flexible 3D images.

Optical parameters of the virtual endoscope like opening angle, viewing angle, barrel distortion and roll can be chosen to obtain comparable images of the virtual endoscopic view and the real endoscopic image. A set of orientation and navigation aids helps finding the optimal path to the sphenoid sinus and allows the validation of the chosen iso-value. The system also provides a simulation of the opening of the sellar floor including color coded indication if critical structures would be damaged or not.

A detailed description of STEPS can be found in the works of Neubauer et al. [91, 92]. The system is commercially not available, but under clinical evaluation. Results have been published by Wolfsberger et al. [139].

StealthStation®- Intra-Operative Navigation

The second basic component of navSTEPS is the image-guided navigation system StealthStation® by Medtronic. A StealthStation® delivers real-time feedback on patient anatomy, instrument

position, and therapy placement for both procedure planning and intra-operative confirmation. The system optically tracks the endoscope and up to four additional tools during the intervention. A software interface to the StealthStation® can be realized using the StealthLink® research portal that provides access to information like the patient space/image registration transformation and the tool positions.

2.4 navSTEPS - System Overview

navSTEPS combines the visualization, simulation and planning capabilities of STEPS with the tracking information of the navigation system.

To integrate both systems an additional communication layer has been implemented using StealthLink®. Customized calibration procedures for external and internal camera parameters guarantee the comparability of real and virtual endoscopic images. Visualization, planning and interaction possibilities have been extended for intra-operative use.

navSTEPS supports the whole work flow consisting of four phases: acquisition, planning, setup, and intervention. The user interface is designed in a way that each phase is mapped to its own specialized working environment providing computational assistance for all tasks of the respective phase. The following paragraphs give a workflow oriented overview of navSTEPS.

Data Acquisition

For pre-operative planning and conventional image guided navigation of endoscopic pituitary surgery CT (512×512 axial, 100-200 slices), a T1-weighted MR and a MR angiography data is acquired.

The CT data provides, besides information on the location of bony structures, clear air-tissue boundaries required to generate high quality virtual endoscopic images. The MR datasets contain information on tumor location and extend, pituitary gland, optical nerves, and cysts (T1 weighted MR) and about the internal carotid arteries (MR angiography).

Data Pre-Processing and Parameter Adjustment

navSTEPS integrates the whole functionality of the original STEPS to prepare the available image data for later generation of patient specific virtual endoscopic images: MR and CT data are registered, and critical structures are segmented and stored as binary masks. The pre-selected iso-value of -500 HU for surface reconstruction can be adjusted with respect to important thin structures affected by the partial volume effect. Endoscope parameters like the opening angle can be chosen according to the real setup during the intervention (see Section 2.3 for further details).

Intervention Planning

navSTEPSs main tool for patient individual pre-operative planning of the approach to the pituitary gland is the virtual endoscopic view enhanced by color coded critical structures that is also

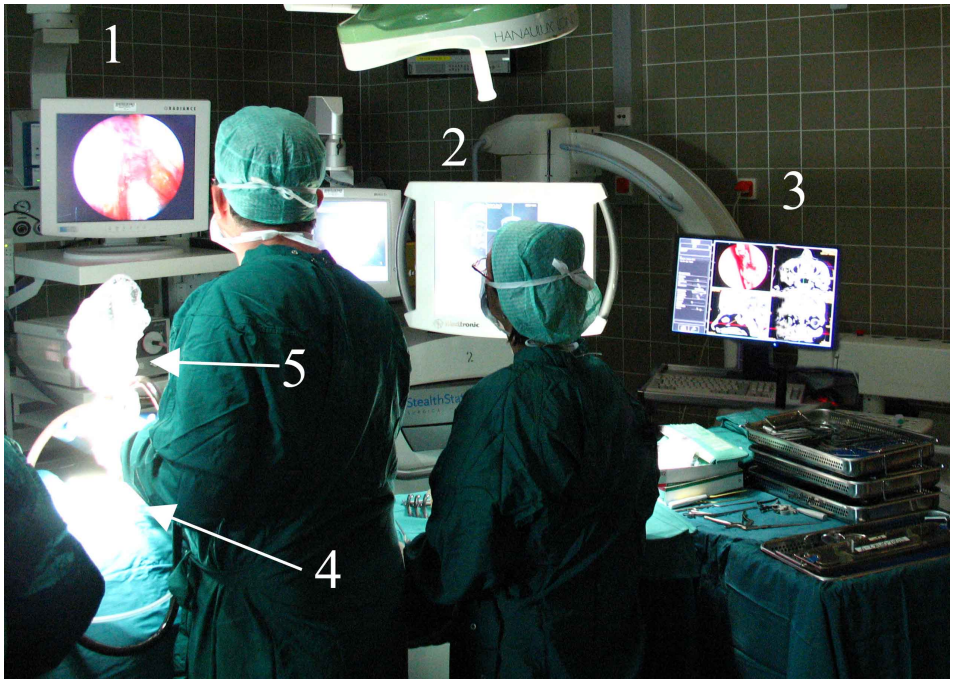


Figure 2.4: Intra-operative setup: Endoscope monitor (1), intra-operative navigation system (2), navSTEPS monitor (3), patient (4), endoscope (5).

part of the original STEPS. It allows the surgeon a pre-operative simulation of intra-operative situations that can help to avoid or to be prepared to critical situations (see Section 2.3 and related literature). The exact procedure for pre-operative planning of the intervention highly depends on the experience level of the surgeon, the individual anatomical situation and pathological findings of the patient.

For the intra-operative use, navSTEPS extends the visual guidance functionality of STEPS by so called waypoints that can be placed directly in the MPR of the pre-operative data to mark important landmarks like the sphenoid sinus ostium along the path.

To assure a free view to the sellar floor, which is needed during the most important phase of the intervention, a clipping plane (Section 2.5) can be defined pre-operatively that can be enabled during the intervention.

Setup

The intra-operative setup consists of an endoscope, an intra-operative navigation system, and a PC which runs the extended virtual endoscopy software navSTEPS. Figure 2.4 shows how the system is assembled in the OR.

At the beginning of the intervention the intra-operative navigation system is initialized by registration of the physical patient space to the pre-operative CT and/or MR images. Passive tracking targets are clamped on the endoscope and a second tool (e.g., the suction) which acts

as a pointer. Both tools are calibrated using the standard procedure provided by the navigation system.

After completion of the setup of the navigation system, the same CT data is loaded into navSTEPS together with the pre-segmented background objects and pre-operatively defined additional parameters.

A calibration step ensuring the comparability between real and virtual endoscopic images concludes the setup phase (see Section 2.5 for implementation details).

Intervention

All visualization capabilities of STEPS are fully available during the intervention. navSTEPS extends this functionality by providing additional features to handle the intra-operative visualization and interaction requirements for the application.

Reduced User Interaction. While STEPS was designed as highly interactive system, navSTEPS takes the interaction constraints during surgery into account like reduced moving space of the surgeon, sterility, and time constraints. The intra-operative system is almost interaction free: Currently only an eventually pre-operatively defined clipping plane has to be activated by one single mouse click or by using a foot switch.

Global Orientation Through Enhanced MPR Views. navSTEPS constantly acquires the current position of the endoscope and the pointer and updates all tracked views of the application accordingly. Global orientation is supported by the available MPR view that is steered per default by the tip of the endoscope but automatically switches to the tip of the pointer if present. This feature allows an intuitive exploration of structures close to the pointer directly on the original image data.

Local Orientation by Relation of Virtual and Real Endoscopic Images. The virtual endoscope follows the tip of the real endoscope and provides the corresponding virtual view based on the pre-operative data. The navigation system acquires the position of the endoscope with high precision, but tracking errors introduced by the technical properties of the endoscope (see Section 2.5) and the differences between the real intra-operative situation and the pre-operative images (see Section 2.5) make the exact match of the virtual and the real endoscopic image virtually impossible. Nevertheless, the high quality of the virtual images allows the surgeon to establish the relationship between the virtual endoscopic image and the intra-operative situation by direct visual comparison of anatomical landmarks. navSTEPS additionally enables the surgeon to directly control spatial relationships by using the pointer: The tool is visible in reality in the real endoscopic image and is displayed as cross hair in the virtual image.

To generate meaningful virtual endoscopic images during all stages of the surgery navSTEPS provides two mechanism to overcome the visualization problems associated with the differences between pre-operative data and intra-operative situation: (a) Especially during the nasal phase, where tissue is flexible and mucosa might be more or less swollen, it can easily happen that the virtual endoscope is placed within virtual tissue. navSTEPS automatically recognizes such a situation and applies a modified rendering algorithm that still generates a meaningful image. (b) Pre-operatively defined clipping plane(s) can be activated by mouse click or foot switch to easily handle tissue removal mainly during the sphenoid phase of the surgery (see Section 2.5 for implementation details).

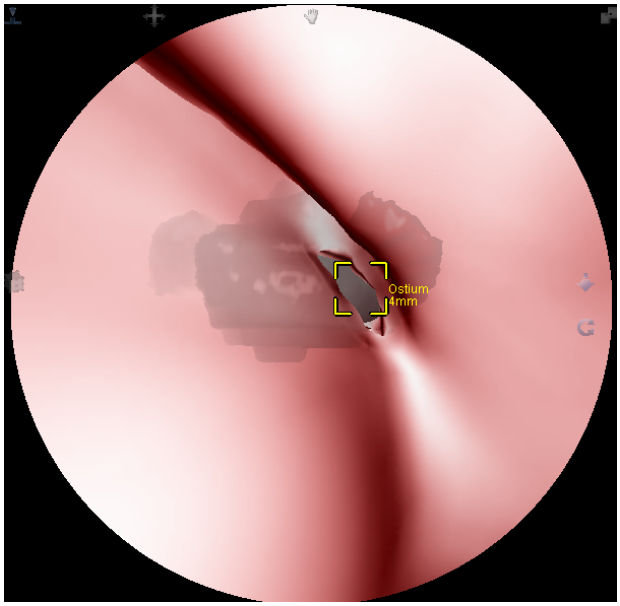


Figure 2.5: Waypoint marker at the ostium. Segmented structures, adenoma, pituitary gland and carotid artery are already visible in the backround.

Visual Guidance by Waypoint Marker. Pre-operatively defined waypoints are visualized as square overlays (see Figure 2.5) and guide the surgeon towards relevant landmarks. Additionally the name of the waypoint and the distance to the endoscope tip is displayed next to the marker.

Localization of Critical Structures During the Sphenoid and Sellar Phase. STEPS visualizes previously segmented structures that are generally hidden in the real endoscopic image as colored overlays on the MPR views and as background objects directly in the virtual endoscopic image (see Figure 2.3). navSTEPS combines these images with the navigation information of the real endoscope and the pointer. In this way it effectively supports the surgeon in localizing critical structures behind the occluding tissue and/or bone. This feature is especially important during the critical sphenoid and sellar phases of the surgery when the surgeon has to determine the optimal position for the opening of the sellar floor. An example is given in Figure 2.6.

2.5 Implementation Details

Camera Model and Calibration

The endoscope and additional tools are calibrated to the navigation system by measuring two positions, the tip p_{tip}^{tool} and the rear part (hind) p_{hind}^{tool} with a special calibration device. This positions are defined in tool space, i.e., in the coordinate system of the tracking target which is attached to the tool. The navigation system updates continuously a matrix T_{tool}^{image} which allows to transform from tool to image (CT data) space.

$$p^{image} = T_{tool}^{image}\, p^{tool} \tag{2.1}$$

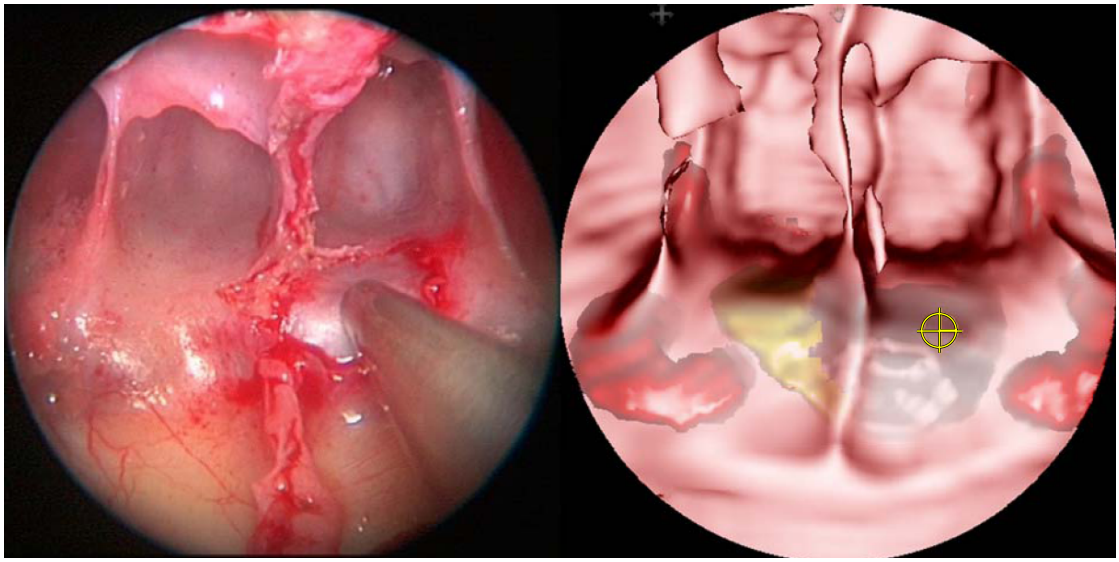


Figure 2.6: Left: Real endoscopic view on the sellar floor. The suction tool visible in front is used as pointer. Right: Corresponding virtual endoscopic view with background objects and crosshair indicating the position of the tip of the pointer above the adenoma (gray). The pituitary gland is displayed as yellow structure and the carotid arteries are indicated in red.

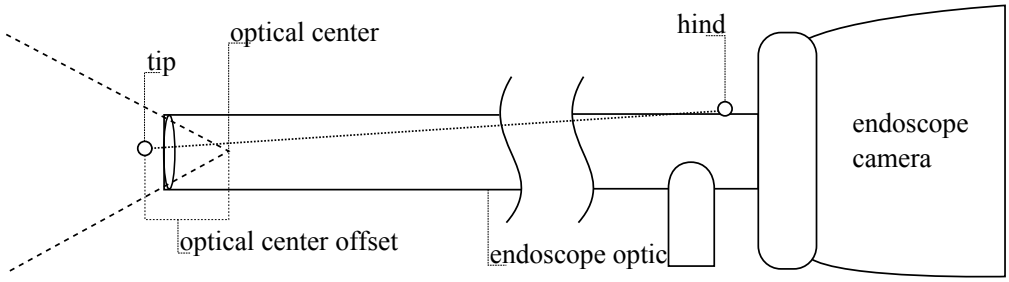


Figure 2.7: Endoscope Assembly

For simple pointing tools the two resulting positions in image space are sufficient, however driving a virtual endoscope requires more attention. The endoscope used for endonasal transsphenoidal surgery consists of two parts: A long and thin monocular rigid optic and the video camera (see Figure 2.7), neither the position of the optical center nor the orientation can directly be derived from the measured points. Thus, a camera calibration has to be performed in two ways. First, camera position and orientation (extrinsic parameters) have to be corrected. Second, the intrinsic camera parameters have to correspond with the parameters of the real endoscope.

Extrinsic Camera Parameters

Based on the tip and hind position a virtual camera is defined in tool space such that the optical center is located at the tip and the viewing direction is aligned to the axis defined by tip and hind. However since the tip position is measured in front of the lens and the hind position is most likely off the optical axis (see Figure 2.7) further corrections are required to mimic the real endoscope. The roll angle (rotation around the optical axis) is an additional parameter that must be adjusted.

Therefore a correction transformation C is inserted into Equation 2.1 which can be used to transform the virtual camera from tool to image space:

$$p^{image} = T_{tool}^{image}\, C\, p^{tool} \tag{2.2}$$

The correction transformation C is assembled from a translative part which moves the tip into the optical center and a rotational part which corrects viewing direction and roll angle. Scaling or shear transformations are not allowed.

Intrinsic Camera Parameters

For intrinsic camera calibration only the field of view v_{fov} is estimated so far. More complex visual behavior like radial distortion can be visualized by the rendering system but is not used since these parameters are not captured at the moment.

Calibration Procedure

To define the final camera model, the correction transformation C and the field of view v_{fov} have to be estimated. Currently a simple manual calibration is used that requires the definition of the field of view, which is generally known for the used endoscope, and the alignment of the roll angle. The pre-defined position for the optical center is 3 mm behind the measured tip (see Figure 2.7) and generally does not need further adjustment.

In our clinical study we used the inferior nasal turbinate as a reference to define and verify the chosen roll angle (see Figure 2.8), because this structure can be reached in an unproblematic way and has a characteristical appearance.

Endoscope Related Errors

The connection between the optic and the camera is designed to make a fast interchange of optics possible and allows the rotation of both parts against each other. This construction imposes problems building a virtual camera model because the intra-operative navigation system is only able to track the camera body or the endoscope lens. The tracking target is attached onto the endoscope optic in our setup. If the camera is rotated against the lens, the calibration (at least the rotational part along the viewing axis) is invalidated.

Another source of errors is the optics itself. Especially long and thin build types can be bend easily applying little force which can result in positional errors of several millimeters.

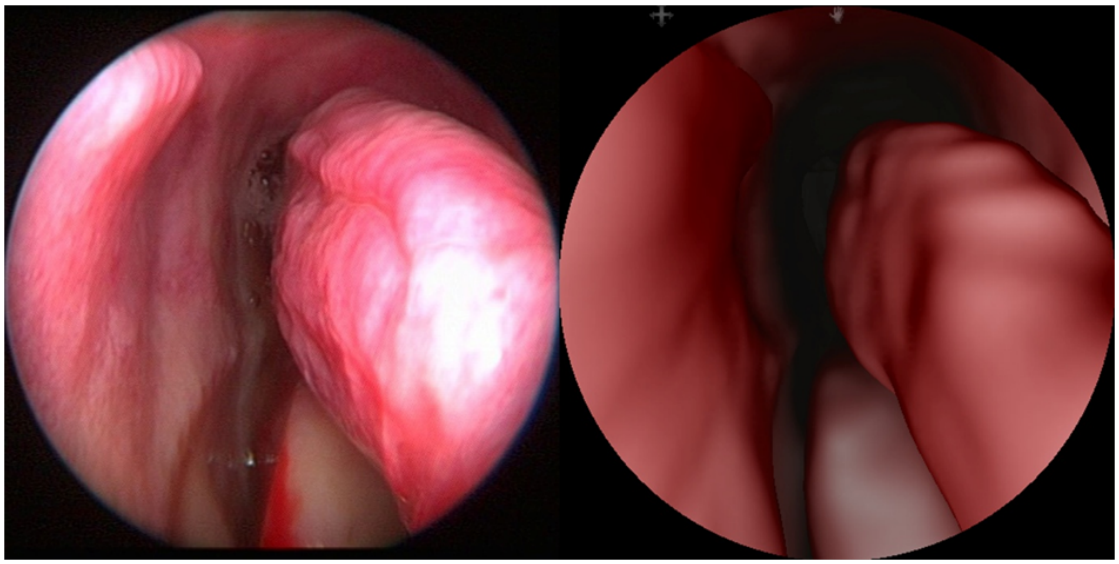


Figure 2.8: Intra-operative and virtual image of the inferior turbinate.

Occlusion Handling

Besides of the extrinsic tracking errors described in the previous section, there are mainly three situations in which the real patient anatomy differs from the anatomy captured pre-operatively:

1. The CT images were acquired hours or days before the surgery and do not represent the current anatomical state which likely has changed due to natural swelling.
2. Deformation of mucosa.
3. During surgery tissue and bony structures are resected.

Differing real and virtual anatomical structures pose several problems. There are two situations where the virtual view can be highly different to the endoscopic view. First, if large displacements of soft structures occur during surgery due to (de-)swelling, compression or shift, the tip of the endoscope is placed inside the virtual tissue and the default first-hit-raycasting algorithm is not capable to produce a meaningful image. Second, if the surgeon has removed a layer of tissue, it will still be visible in the virtual view and may block the view.

Both classes of problems cannot be avoided. Nevertheless the system must provide a meaningful view especially on critical structures in any situation. To achieve this we propose two different strategies:

- The use of a robust rendering algorithm which provides meaningful images even if the conventional rendering approach would be occluded or show artifacts (see Section “Transparent Backface Rendering”).
- A simple clipping system to effectively handle occlusion problems related to tissue removal (see Section “Clipping”).

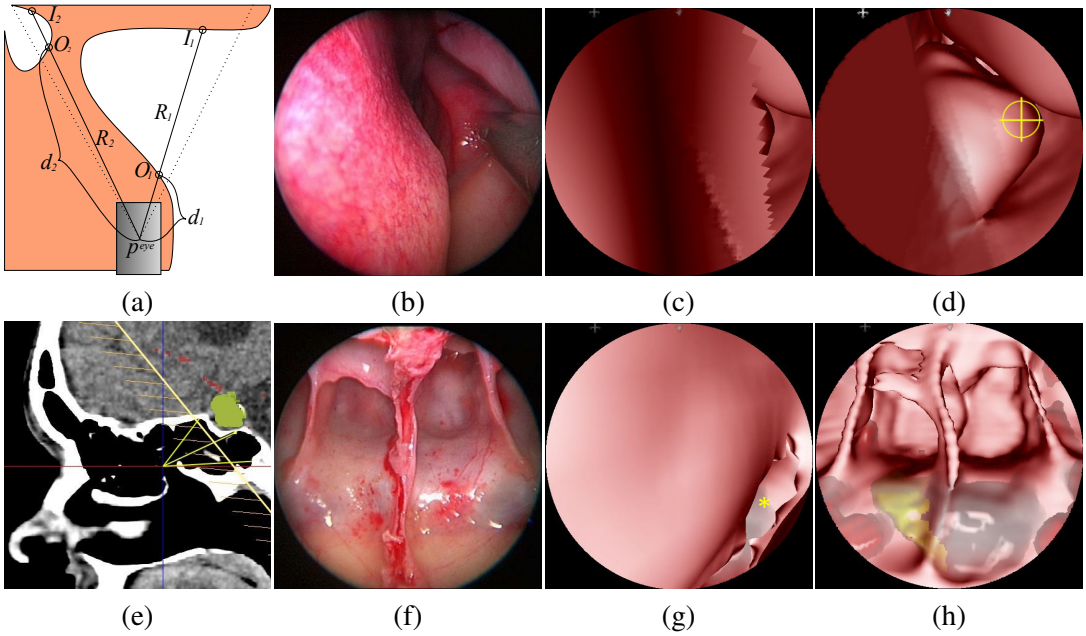


Figure 2.9: Transparent backface rendering: (a) Algorithm. (b) Intra-operative endoscopic view. The endoscope touches the nasal septum on the left. The middle nasal turbinate is visible on the right side (deformed by the suction). (c) Pure iso-surface raycasting resulting in a complete loss of sight because the virtual camera is located inside the nasal septum. (d) Transparent backface rendering provides sight on most of the anatomy. The middle nasal turbinate is clearly visible (however the turbinate is not deformed as in the real endoscopic view). Clipping: (e) Sagittal CT, the endoscope is located in front of the sphenoid ostium facing towards the sellar floor. The clipping plane is depicted as a yellow line running through the sphenoid sinus. (f) Intra-operative endoscopic image with a clear view on the sellar floor because the septum around the sphenoid ostium has been resected. (g) Virtual image without clipping. The sphenoid ostium is visible in the lower right corner (marked with *). (h) Clipping is activated. The rendering provides a clear view on the sellar floor, matching the real endoscopic view.

Transparent Backface Rendering

The virtual endoscopic images are generated by a first-hit raycasting algorithm. The algorithm computes the color for each pixel by tracing a ray through the volume until the underlying value exceeds the chosen iso-value. Starting from this position the exact intersection with the iso-surface and the corresponding normal is computed which is used to shade the pixel. This basic algorithm works fine as long as the tracing starts at a position with a value below the given iso-value (which is the usual case if the virtual endoscope is located outside of any tissue). If the optical center of the virtual endoscope is located inside virtual tissue, the tracing procedure will stop immediately and the algorithm does no longer deliver useful images.

Transparent Backface Rendering extends the original first-hit raycasting algorithm in a way that it still generates meaningful images if the virtual camera is placed slightly inside the tissue:

If the value at the beginning of the ray already overshoots the threshold, the modified casting procedure searches for the first point along the ray with an iso-value below the given threshold (the "exit point"). From this point on the basic direct iso-surface raycasting is performed. Ray R_1 in figure 2.9(a) illustrates how the algorithm finds the intersection with the correct iso-surface I_1 by first searching for the exit point O_1.

The limitation of this straightforward approach is depicted by tracing ray R_2: If the search for the exit point is not restricted, the hidden cavity (I_2) not accessible by the real endoscope might become visible. This problem has been solved by introducing a maximal search distance d_{max} that limits for each ray the tolerated distance from p^{eye} to O_i. d_{max} has been chosen $\leq 10mm$. This range generally covers the small differences between real and pre-operative data caused by swelling, tissue removal, slight tissue shifts and/or tracking errors.

To clearly indicate that the virtual endoscope lies inside virtual tissue, and that the virtual images differ from the current real situation, the pixel color is dimmed proportional to the distance d_i.

Figures 2.9(b)-(d) show a typical situation. The endoscope is pressed against the nasal septum at the left side 2.9(b). This results in a deformation of the tissue which allows the virtual camera to be located inside the virtual nasal septum. Figure 2.9(c) shows the result of the default first-hit raycaster rendering which generates no usable result. Figure 2.9(d) depicts the the improved algorithm. The middle nasal turbinate is clearly visible however not deformed as in the real view.

Clipping

Clipping is one way to guarantee visibility of important structures in the virtual endoscopic image if tissue and/or bone have been removed during surgery to advance the endoscope (e.g., opening of the sphenoid ostium to access the sphenoid sinus) or to get a clear view on important structures (e.g. removal of septa in front of the sellar floor).

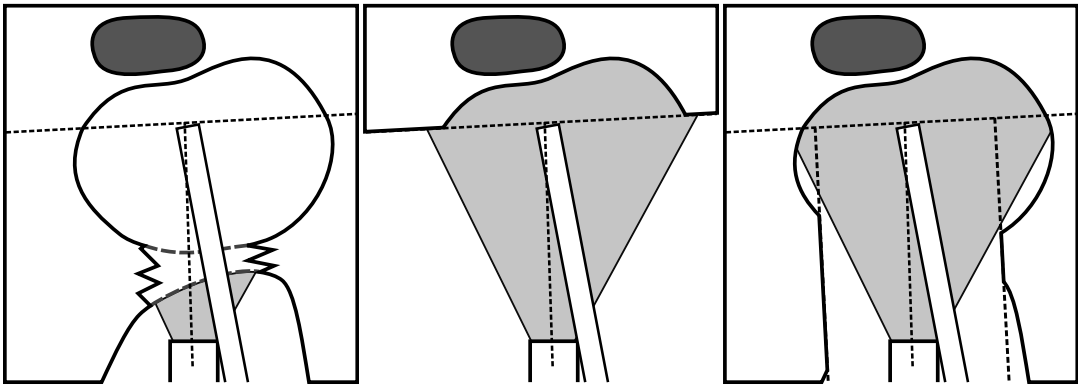
The original STEPS provides sophisticated tools to simulate tissue and bone removal, but for intra-operative use the required user interaction is far too complex. To reduce user-interaction to a minimum, navSTEPS implements virtual removal of tissue as a simple clipping plane. Two interaction metaphors have been realized: pre-defined and interactive clipping.

Pre-defined clipping is setup in the pre-operative planning phase by defining a fixed clipping plane. During intervention only enabling/disabling of this clipping plane is possible.

Figure 2.9(e)-(h) shows an example where the clipping plane has been placed in front of the sellar floor. If activated, the virtual endoscopy generates comparable images to the real endoscopic view and allows the localization of the tumor and other critical structures.

Interactive intra-operative clipping has been realized by using the pointer (e.g., suction) tool to define the clipping plane (see Figure 2.10(a)). The clipping plane is constructed at the tip position of the suction perpendicular to the connection vector between eye position and pointer tip (Figure 2.10(b)) and is activated by using a footswitch.

Additional *cylindrical clipping* provides a clipping mechanism that emulates opening of sinuses, removal of septa and bone (Figure 2.10(c)).



(a) (b) (c)

Figure 2.10: Interactively controlled clipping. The surgeon has opened a paranasal sinus. The view of the virtual endoscope is still blocked because the image is generated based on pre-operative CT data (a). With the tip of the suction a clipping plane is defined to enable a view into the cavity (b). A cylindrical clipping shape can be used to simulate the opening more realistically (c).

2.6 Results

The system has been tested so far in one cadaver study and twelve clinical cases.

Preparation Time. Pre-operative planning using the original STEPS needs about 13 minutes including image registration, segmentation and anatomy exploration. The additionally required planning steps for intra-operative use provided by navSTEPS (placing of the clipping plane and waypoints) takes at maximum two minutes.

The intra-operative setup of navSTEPS (connection of PC and StealthStation, start of the system, loading of pre-operative images and pre-defined parameters) and the calibration of the virtual camera requires maximally 5 minutes of additional time.

Feasibility. The feasibility of navSTEPS has been assessed by the performing neurosurgeon (S. Wolfsberger) in relation to the clinical features of the individual case. Figures 2.11 - 2.14 shows examples of the use of navSTEPS in all stages of an endonasal transsphenoidal approach to the sellar floor.

The results of the evaluation based on the twelve clinical cases are listed in Table 2.1. The virtual endoscopic images were found useful:

1. During the approach.

    a) In cases of anatomic variations: The transnasal approach to the sphenoid sinus can be hindered by various individual anatomic variations such as septal deviations and spurrs, bullous turbinates and postsurgical scarring all narrowing the nasal cavities. The virtual endoscopy system was able to navigate the surgeon to the sphenoid ostium even in the case of a narrow nasal corridor by means of the waypoint system

no.	*pathology*	*size*	*PS extension*	*invasive*	*anatomic variation*	*resection*	*benefits of navSTEPS*
1	NF	2.5	yes	yes	firm tumor consistence	subtotal	Position of ICA during opening of sellar floor. Position of ICA during parasellar tumor removal.
2	ACTH	1.0	no	no	presellar sinus	complete	Projection of PG, tumor and ICA on thick bony sellar floor during drilling.
3	PRL	1.0	no	no	cystic tumor	complete	Localization of cyst for tailored sellar floor opening.
4	NF	2.0	no	no	firm tumor consistence	complete	Position of ICA during opening of sellar floor.
5	RCC	2.5	no	no	cystic	cyst drainage	Localization of cyst for tailored sellar floor opening and cyst puncture.
6	PRL	1.5	yes	no	parasellar extension	complete	Localization of PG and tumor for tailored sellar floor opening. Localization of ICA during parasellar tumor removal.
7	NF	3.0	no	no	multiseptated sphenoid sinus	complete	Position of ICA during opening of sellar floor.
8	CPG	3.0	no	yes	cystic	subtotal	Position of ICA during opening of sellar floor.
9	NF	3.0	no	no	narrow nasal corridor	complete	Nasal anatomy for maximum opening of sphenoid sinus. Position of ICA during opening of sellar floor.
10	NF	2.5	no	no	-	complete	Position of ICA during opening of sellar floor.
11	GH	1.5	no	no	nasal septum deviation, bullous concha	complete	Nasal anatomy for maximum opening of sphenoid sinus. Position of ICA during opening of sellar floor.
12	HP	1.0	no	no	-	biopsy	Localization of PG and lesion for tailored sellar floor opening and biopsy.

Table 2.1: ACTH: adrenocorticotroph-cell adenoma, GH: growth-hormone-cell adenoma, ICA: internal carotid artery, NF: endocrine non-functioning adenoma, PG: pituitary gland, PRL: prolactinoma, RCC: Rathke's cleft cyst, CPG: craniopharyngeoma, HP: hypophysitis

and the visibility of anatomical structures through the extended rendering algorithm (see Table 2.1, case number 9, Figure 2.11).

b) For individually tailored opening of the sellar floor: During the sphenoid phase of the procedure it is crucial for complete tumor resections to maximize the opening of the sellar floor in cases of large adenomas. The lateral extension of such an opening is defined by the siphon of the internal carotid arteries. The available clipping mechanisms provided in all cases a clear view to the sellar floor. In combination with the visualization of the internal carotid artery hidden behind the bony walls of the sphenoid sinus navSTEPS was able to safely guide the surgeon during sellar floor opening (see table 2.1 cases 1, 4, 6-11, Figure 2.13). In cases of small sellar lesions, an individually tailored opening of the sellar floor is necessary to optimally expose the tumor while maintaining the function of the normal pituitary gland. The virtual endoscopy system was able to visualize the pathology in relation to the pituitary gland behind the bony walls of the sphenoid sinus. With this information, the surgeon was able to place his approach through the sellar floor in an optimal position (see Table 2.1, cases 3, 6, 12, Figure 2.14).

2. During tumor resection for respecting the internal carotid artery.

   In cases of tumor extension lateral to the internal carotid artery (parasellar extension), the virtual endoscopy system was able to provide the surgeon with the information about the position of the artery in relation to the tumor, thereby increasing the safety of this challenging task and maximizing tumor resection (see Table 2.1, cases 1 and 6, Figure 2.14).

Accuracy. The accuracy as been checked by correlation of the nasal and sphenoid septation. The position of the structures has been compared between the live view, the virtual endoscopic view and the sectional view presented by the navigation system. The deviation was always in the standard range of the navigation system i.e., 2 mm both at the beginning and during later stages of the procedure.

Usability. The usability of the system was analyzed for the intra-operative situation. Well received by the surgeons was the fact that the system requires almost no additional interaction (activating the clipping plane is the only exception) after the setup. The direct coupling of the virtual camera to the real endoscope makes it possible for the surgeons to do their work as usual and just go back to the new visualization if additional information is needed without any additional effort.

2.7 Discussion and Future Work

The potential benefit of intra-operative virtual endoscopy for image guided endonasal transsphenoidal pituitary has been discussed in many publications and is mainly undisputed. The limited acceptance of such systems in the OR is basically based on the visualization problems induced by the specific anatomy of the nasal cavities, correspondence problems, and tracking errors.

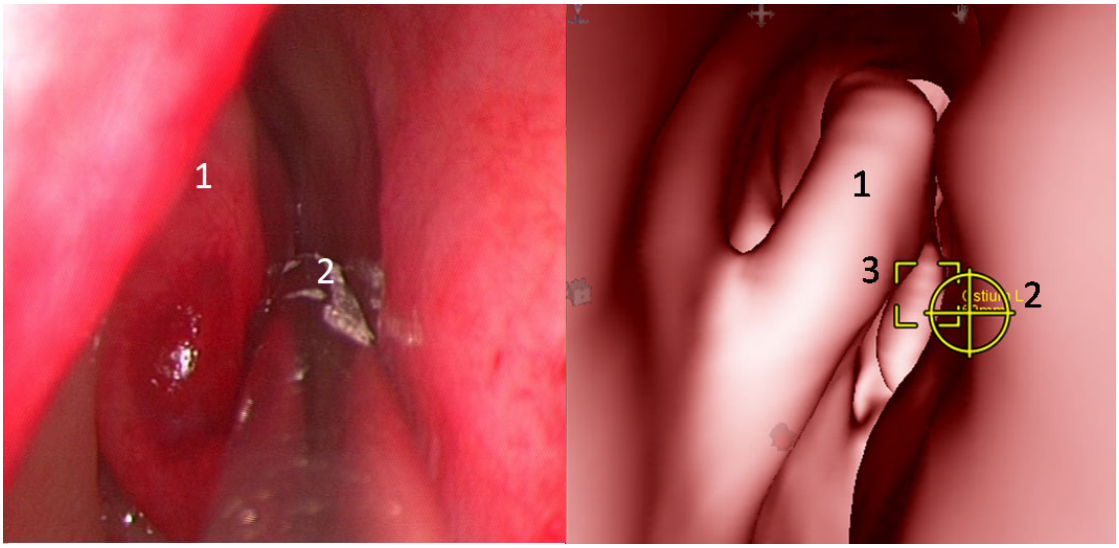


Figure 2.11: Virtual and real image of the tip of middle turbinate (1), suction (2), trajectory to sphenoid ostium (3). Note that the middle turbinate is deformed to the left.

navSTEPS, the system presented and tested in this paper, is a highly specialized solution addressing the two main visualization problems that hamper the acceptance and use of virtual endoscopy for image-guided endonasal transsphenoidal pituitary surgery in the OR. The analysis of the twelve clinical cases performed with the system have shown that it is possible to create meaningful virtual endoscopic images through all stages of the surgery and that the system can add to the safety of the procedure.

Some currently existing technical problems may be solved in the future with the availability of new innovations regarding the technical equipment: Upon the next generation of endoscopes the weaknesses of current endoscopes might be improved and it will become possible to integrate a reliable automatic camera calibration method into navSTEPS.

In the future the use of navSTEPS for other paranasal sinus interventions will be evaluated. In this case interactive clipping techniques and other methods which allow the surgeon to control visibility become even more important and need extensive evaluation.

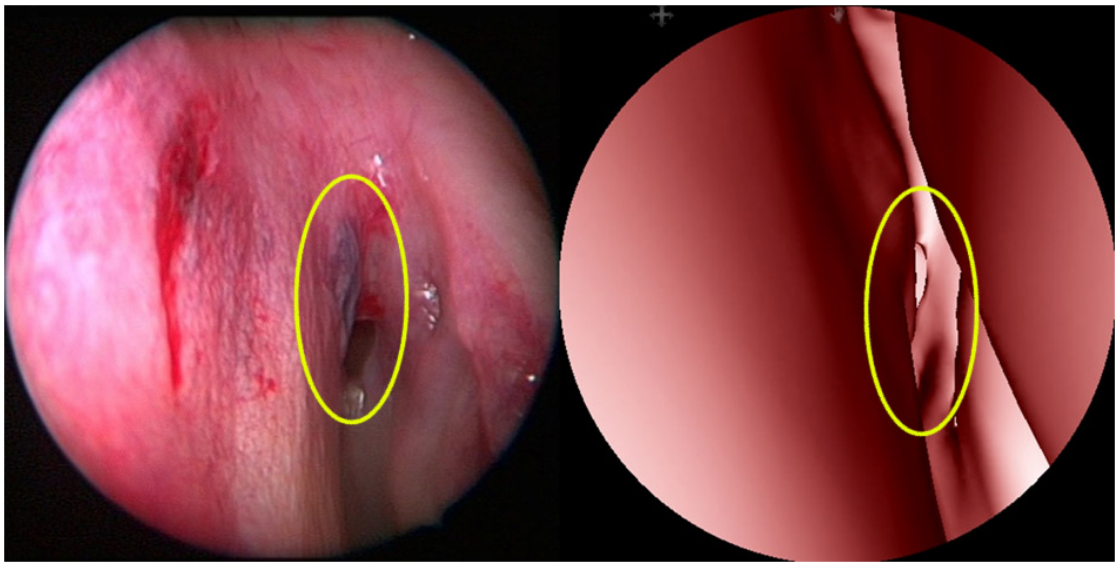


Figure 2.12: Corresponding intra-operative and virtual image of the sphenoid ostium (marked in both views).

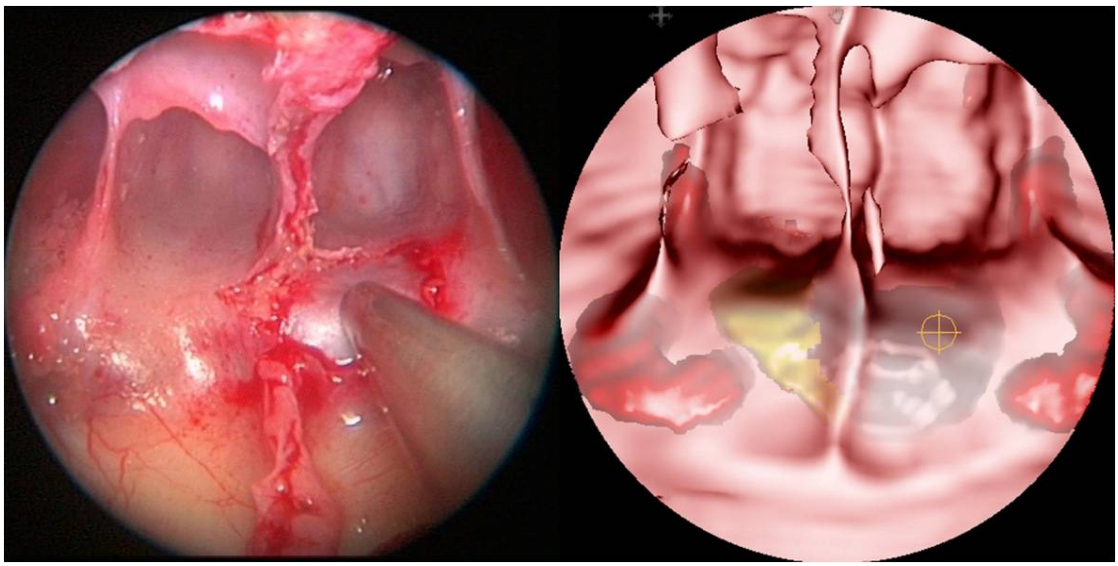


Figure 2.13: The activated clipping plane provides free view to the sellar floor with suction.

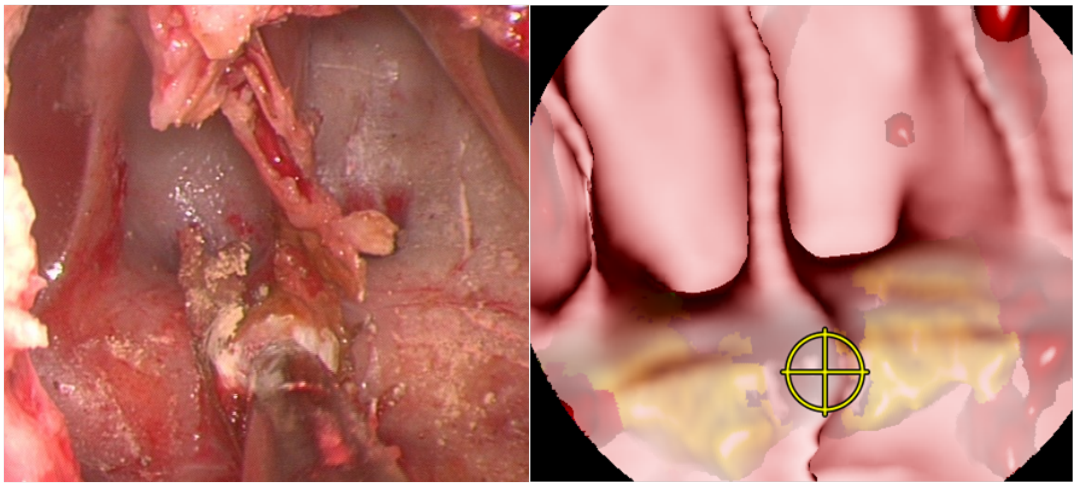


Figure 2.14: Tailored opening of the sellar floor superficial to the adenoma with a high speed drill. The crosshaires in the virtual image indicate the position of the suction tool placed on the drilled opening.

CHAPTER 3

Fast and Memory Efficient Feature Detection using Multiresolution Probabilistic Boosting Trees

This chapter is based on the publication:

> Florian Schulze, David Major, and Katja Bühler. *Fast and memory efficient feature detection using multiresolution probabilistic boosting trees.* In Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), volume 19, pages pp. 33–40, 2011. [117]

This paper presents a highly optimized algorithm for fast feature detection in 3D volumes. Rapid detection of structures and landmarks in medical 3D image data is a key component for many medical applications. To obtain a fast and memory efficient classifier, we introduce probabilistic boosting trees (PBT) with partial cascading and classifier sorting. The extended PBT is integrated into a multiresolution scheme, in order to improve performance and works on a block cache data structure which optimizes the memory footprint. We tested our framework on real world clinical datasets and showed that classical PBT can be significantly speed up even in an environment with limited memory resources using the proposed optimizations.

3.1 Introduction

In the past years various methods for automatic processing and understanding of medical 3D image data have been developed. One important building block is the automatic detection of anatomical landmarks. Detection of these features stands often at the beginning of the processing pipeline: it transforms the dense volume representation into a sparse set of possible landmark locations, allowing a significant acceleration of subsequent high level segmentation methods.

However, making the transition from pure research algorithms which focus often solely on detection performance to real world radiology applications brings a number of additional requirements into consideration. The algorithm has to be able to deal with possibly limited technical resources - not all workstations in a hospital might be equipped with the newest hardware, and the algorithm shall run in the context of radiology workstation software which already occupies resources. Excellent time performance is required because automatic algorithms often substitute manual workflows while the result must be authorized and/or adjusted by the radiologist. In this case an automatic algorithm will only be used if the execution time of the algorithm is considerably shorter than the manual approach would be.

This work presents a highly optimized general purpose feature detection framework for the effective reduction of possible feature candidate positions in 3D image data as preprocessing step for more expensive object detection methods. Referring to the clinical application context, we designed our method according to the following requirements:

1. Time Performance: The result must be calculated in a relatively short time (e.g., within seconds) in order to be usable in a clinical environment.

2. Memory Performance: The algorithm must also execute on standard PCs with limited technical resources.

Thus, the focus of the proposed algorithm and its implementation is on a small adaptable memory footprint while retaining as much execution speed as possible.

3.2 Related Work

Object recognition, and local feature detection as a subdiscipline of it, are for many years core topics of computer vision research.

Point based methods beginning with the Harris corner detector [43] try to automatically extract points of interest from an image. Exact control of which points are extracted is not supported, therefore recognition of complex structures/areas is done by combining sets of feature points. The most prominent point detector is the SIFT algorithm [75] which overcomes the limitations of previous solutions by being scale, rotation and perspective invariant. However, translating SIFT, which is aimed for 2D images, to 3D volumes suffers from dramatic performance problems. Niemeijer et al. report [93] that SIFT feature extraction on a $200 \times 200 \times 1024$ volume downscaled by 50% takes 10 minutes to compute.

Machine learning based approaches use a (learned) classifier to decide if a specific region of an image belongs to an object. A prominent example for this class of algorithms is the method for real-time face-detection presented by Viola and Jones [134] that uses a cascade of boosted weak classifiers. A more general approach has been proposed by Tu et al. [130] by introducing Probabilistic Boosting Trees (PBT). PBTs are decision trees which use boosted learners as classifiers in each tree node. Viola's boosted classifier cascades are a special case of a PBT. An alternative to PBTs is the popular d-tree forests method [77] which produces higher detection rates, but with the drawback of much higher execution costs [66].

PBTs have been successfully applied on tissue classification in medical images: Militzer and Vega-Higura [82] use PBT for bone removal in CT angiography. The volume is first split into segments using the watershed algorithm, then each segment is classified with PBT.

Fast preselection of feature candidates for more expensive high level methods is the topic of the paper of Langer and Kuhnert [66]. They integrate classical decision trees with simple color based features and a multiresolution scheme for candidate computation for the expensive SIFT feature detection.

The problem of the large memory footprint of volume data is often discussed in the context of volume rendering. LaMar et al. [65] use an octree structure with blocks containing different resolutions, where only the needed subvolume is downloaded to graphics hardware. However, the whole volume data still has to fit into main memory. This has been improved by Guthe et al. [40] who proposed to hold the data 30:1 wavelet compressed in memory and extract needed data on demand block-wise and cache the data as long as possible.

The purpose of our feature detection method is similar to that of Langer and Kuhnert [66] since we also aim to reduce the list of possible candidate positions as much as possible for later more expensive methods. Langer and Kuhnert tailored their algorithm especially for prefiltering for SIFT feature computation. In difference to them we decided to use the more general PBT [130]. This has several advantages: first, it is independent from SIFT features and easily adaptable to any kind of landmark/structure. Second, decision tree methods can capture large image variabilities captured by a large number of weak classifiers while they only need to execute a relatively small subset of them to perform a concrete classification. Third, they are robust against over-fitting unlike classic decision-tree algorithms.

Our contribution.

To satisfy the high performance requirements of the algorithm in a clinical environment, we extend the original PBT by integrating cascading tree nodes into normal tree building and introduce the concept of classifier sorting (Section 3.3). Both result in higher execution speed of the

classifier. A second performance optimization is achieved by integrating the PBT into a multiresolution classification scheme (Section 3.3). An effective postprocessing step is introduced that applies particle filters to compute probability maps for candidate features for outlier detection (Section 3.3). The memory footprint of our feature detection framework is optimized by the introduction of a multiresolution, multi-derivative block cache data structure (Section 3.4). The performance of our method has been evaluated on a real world clinical use case (Section 3.5).

3.3 Algorithm

In the following we explain in detail the classifier and our extensions on it, the multiresolution feature detection framework and the postprocessing step based on candidate probability.

Probabilistic Boosting Tree with Partial Cascading and Classifier Sorting

Probabilistic Boosting Trees

A PBT [130] is a special kind of decision tree which holds at each tree node a boosted classifier. PBTs are trained top down. Based on a set of positive and negative samples a boosted classifier with a limited number of weak classifiers is trained for each tree node. On each recursion level the sample set is split using the generated classifier and the new subsets are used to train positive and negative child branches. Although multi-class classifiers are possible, we limited our implementation to the simple two-class model.

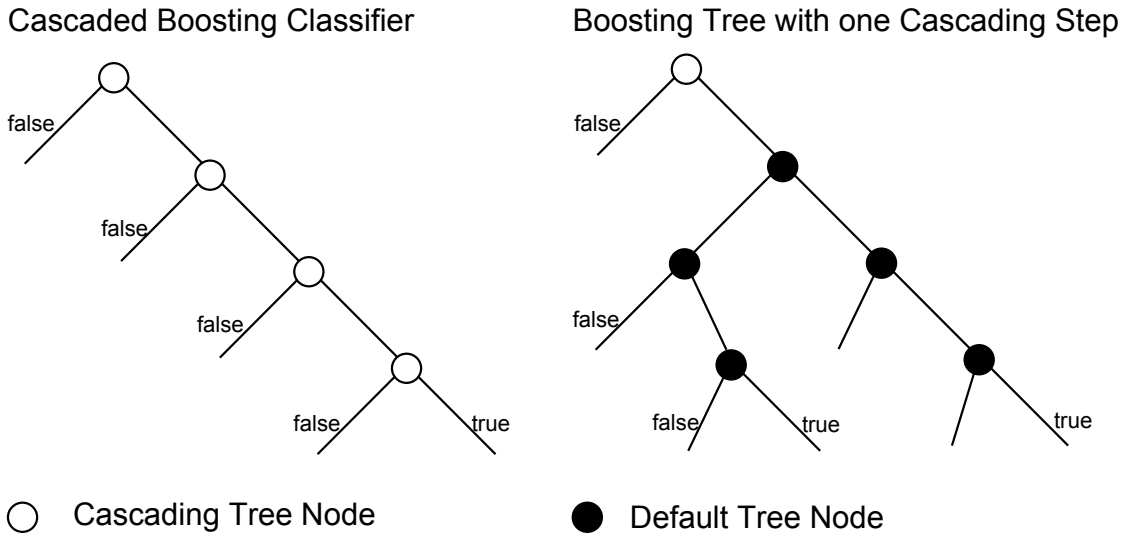


Figure 3.1: Probabilistic Boosting Tree. Left, tree with cascading nodes only. Right, one cascading node at the tree root followed by a default PBT.

Classical Cascading.

If the boosted classifier in each tree node is trained in a way that it does not produce false negative results, the resulting decision tree consists of positive child nodes only. Traversing this tree has only one sequential path and degenerates to the cascade of boosted classifiers of Viola and Jones [134] (see figure 3.1 (left)). Cascading improves execution speed. It allows the classifier to early terminate and reduces in this way the number of classification tests, but it reduces also the flexibility of the original PBT to capture a high variability of features.

PBT with Partial Cascading.

We observed that a high number of samples can be classified as false by executing only one boosted classifier (see section 3.5). This allows to combine the speed-up of cascading with the flexibilty of the PBT by placing one cascaded classifier in front of the PBT: Our tree model contains one cascading node at the root level. A negative outcome stops the classification immediately, a positive outcome is further processed using the full PBT (figure 3.1 (right)).

Classifier Sorting.

We also observed that a high number of samples can be early terminated with a cheap and fast performing classifier (see section 3.5) and that it is advantageous to use expensive classifiers only in places which are executed less often. In our model the most visited place is the cascading node at the root of the tree which can discard a large number of samples as false. The rest of the tree is visited less frequently. Hence, we sort the expensive classifiers into the later tree nodes while the first node can only use fast executing classifiers.

Image Features.

The classifier decides on a per voxel basis if the current voxel belongs to the searched structure or not. Since PBT is a so called ensemble classifier, basically every possible classification method can be integrated. However, the selection of image features has influence on detection performance and execution speed.

In the current work we integrated classifiers which make decisions based on five different image features.

1. Haar-like features with different patterns and sizes.
2. Image intensity
3. Gradients and principal curvatures
4. Region histograms based on image intensity and derivatives with different sampling resolutions and sizes.
5. Structure tensors

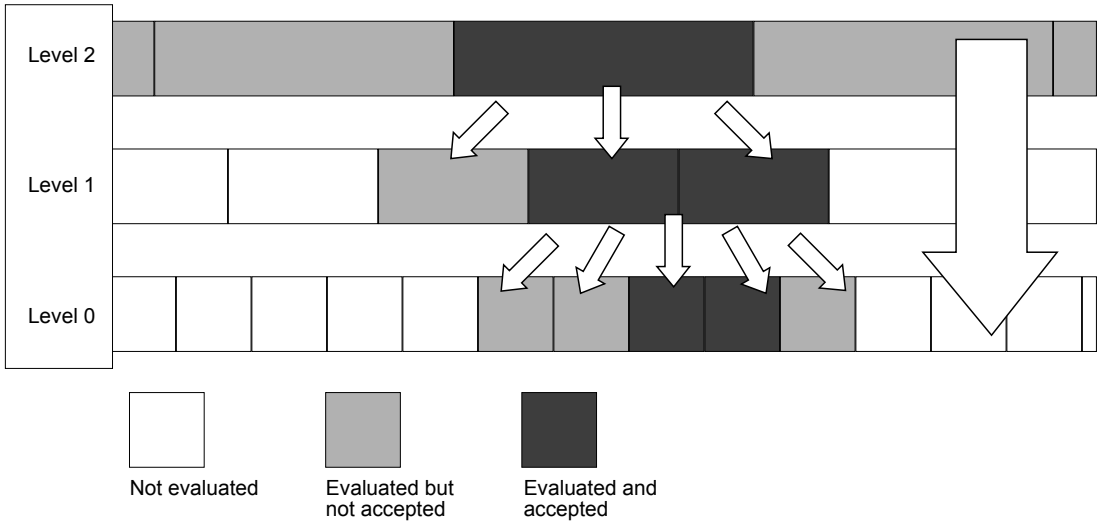


Figure 3.2: Multiresolution Algorithm

Haar-like features and image intensities are the features with the lowest computational costs and are therefore used for building the cascaded root. Gradients need to be computed by filtering as well as principal curvatures which need an additional Hessian analysis step. Region histogram classification multiplies the cost by the number of samples. Structure Tensors require the convolution of the gradient image with a Gaussian kernel and subsequent eigenanalysis of the structure tensor matrix. These three types of classifiers are exclusively used for the non-cascaded part of our PBT.

The chosen weak classifiers are scale variant which is adequate for our application scenario because we expect anatomical structures to have a specific size (small variations in size should be accepted anyway, larger variations because of age or gender can be covered with different detectors and pre-classification based on patient background data).

Multiresolution Feature Detection

The PBT with Partial Cascading is embedded into a multiresolution scheme based on a power of two Gaussian image pyramid [2] to further reduce the number of voxels to be processed.

A separate classifier C_i is trained for each resolution level. Multiresolution classification starts at the lowest resolution level n by applying classificator C_n on image I_n. Classification results in a set of positively marked voxels $(p_0^{+,n}, ..., p_m^{+,n})$. These voxels are propagated into the next higher resolution level $n-1$ where each positive lower resolution voxel marks the voxels within the corresponding filter kernel in level $n-1$ as candidates. Classification of the current level is only computed on the remaining candidate voxel. The propagation is repeated until the original resolution (level 0) is reached. Figure 3.2 depicts the algorithm with a 1D example. Note that most of the high resolution voxels do not need to be checked using this scheme.

In the case of overlapping kernels some higher resolution voxels have two or more parent

voxels and it can happen that a voxel is marked as positive and negative. In this case the positive mark is kept. This leads to a slight over-segmentation, but on the other hand the effect of false negative samples might be reduced, which is a wanted effect.

Filtering of Results Using Fast Probability Computation

The direct result of our feature detection algorithm is a bit mask of candidates which still might contain false positives. One method to reduce the number of false positives is to assign a probability to each candidate that reflects the confidence in its classification. The resulting probability map can then be further processed by thresholding which effectively removes outliers and/or non-maximum-suppression which only leaves the candidates which are at the center of the expected shape.

Probabilistic boosting trees can deliver such a probabilistic classification. The drawback of this straightforward approach is the low time performance.

We observed that the result of feature detection form clusters at the feature location resembling already the searched structure (e.g., the intervertebral discs in figure 3.9). Thus, we propose to assign probabilities to candidates by comparing the shape of its surrounding cluster with the searched shape.

A fast option to compute this are shape particle filters which are applied in our framework. The likelihood that a candidate belongs to the searched structure is computed by applying a shape approximating the structure of interest around each candidate and by measuring the ratio of overlap of neighborhood and shape.

3.4 Implementation

Data Preprocessing

The spatial resolution of medical 3D images in a clinical environment is generally highly anisotropic. Especially the slice distances show high variability from modality to modality, from scanner to scanner depending on the used imaging protocol. The scale variant nature of the image features described in section 3.3 requires the same spatial resolution of all images to be processed.

Thus, training data as well as unseen data is preprocessed by resampling the original volume data to an isotropic voxel size that is selected based on the targeted anatomical landmark. The current implementation uses bilinear interpolation for resampling.

The resampled volume (in the following denoted as “base volume”) is the basis for all successive computations and the original data can be discarded at this point.

Data Management and Derivative Computation

The data management component is responsible for efficiently providing the necessary data to compute the requested weak classifiers on all resolution levels while keeping the memory footprint small and flexible.

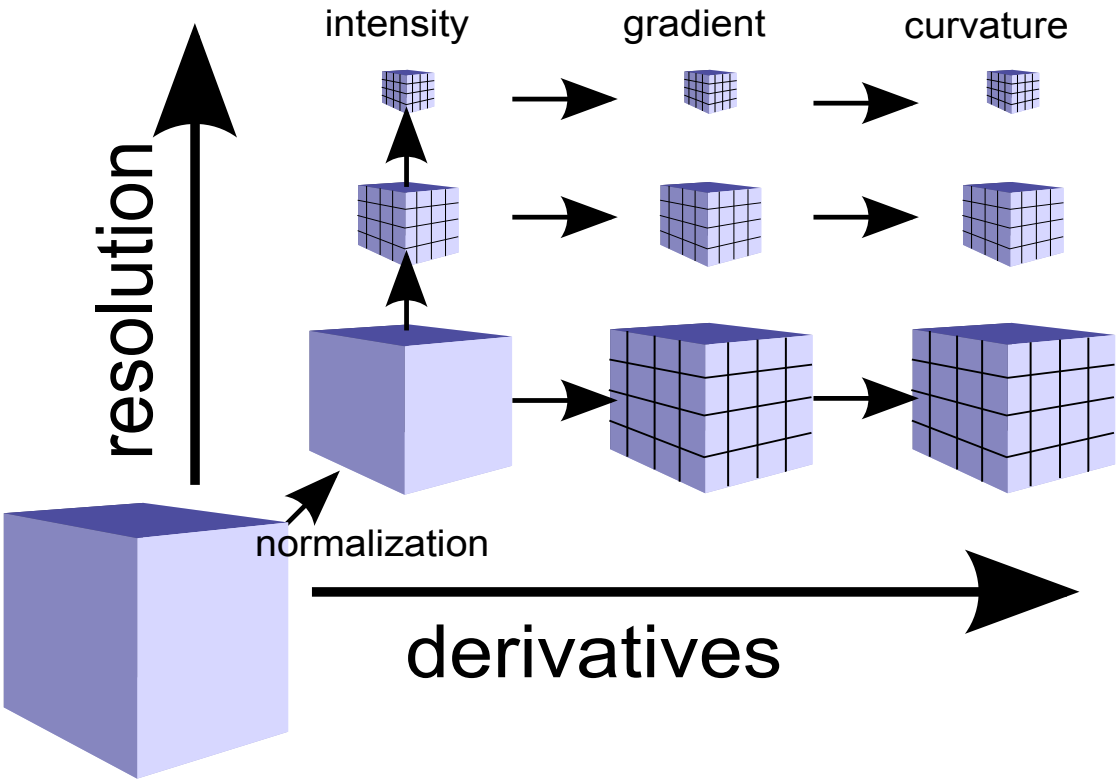


Figure 3.3: Datastructure: Only the base intensity volume is kept completely in memory. All other data, derivative and lower resolution volumes are computed block wise on demand.

The supported weak classifiers require intensity, gradient and principal curvature data for all positively marked voxel positions on the different levels of resolution. It is obvious that the performance of the weak classifiers decides on the performance of the whole PBT.

It is well known that filtering volume data with separated filters for derivative computation is much faster than applying a three dimensional filters per voxel individually. We currently use a $3 \times 3 \times 3$ Sobel filter for gradient computation, which can be replaced by any other appropriate separable filter. However, applying a separable filter for derivative computation requires to keep the whole filtered volume in memory, which might be problematic having our initial requirements in mind.

To overcome this limitation and to make the memory footprint manageable also in an environment with limited resources, we introduce a cached block structure (see Figure 3.3). The intensity base volume is entirely located in memory. Lower resolution volumes, gradients, structure tensors and principal curvature are organized into smaller blocks that are only computed on request. After computation, block data remains cached in memory. If the memory for allocation of new blocks gets low, the cache is partially cleaned by removing data which was accessed the longest time ago.

For fast computation of Haar-like features an additional data structure, an integral volume, is

needed. This data is currently computed as a whole and kept in memory. This is due to the more complicated generation method of this data which makes it hard to compute the value block-wise on demand.

Optimized Classifier Execution

Generally each voxel can be classified individually by executing the whole boosting tree starting from the lowest resolution. However, having in mind that one voxel in a lower resolution volume has influence on a number of voxels in the higher resolution and that the data is arranged in a cached block structure, it is worth to consider a optimal execution order.

Detection of features in the whole volume or in a sub volume follows two strategies. First, feature detection is done in resolution level order. This means that the PBT for one level is executed on the whole region of interest and then all positively classified voxels are propagated to the next higher level.

Second, all per level classification is performed block wise. In this way only a small number of data blocks must be in cache. Any other execution order (for example line wise) would cause a lot of cache misses and would likely lead to frequent re-computations of block data. If multiple classifiers must be applied on the same volume all classifiers are executed on each block sequentially. After the first classifier is executed the block cache remains in a (partially) filled state. Data which is already cached must not be computed if the next classifier tries to access this data. Parallelization is implemented using a worker thread-pool. Classification of one block is fed into a job queue which distributes the work to the worker threads.

3.5 Experiments

Our multiresolution PBT framework was tested in a real world scenario as preprocessing part for a semi-automatic annotation algorithm for the vertebral column. The task was to preselect appropriate candidates for the location of the intervertebral discs and the spinal canal.

For the intervertebral discs, three different detectors were trained to cover the different appearance of lumbar, thoracic, and cervical disks. The spinal canal could be detected by using only one detector.

Setup and Training

The algorithm has been trained and evaluated on 19 CT datasets (13 for training 6 for evaluation only) containing different parts of the vertebral column. The datasets have up to 1112 axial slices with a slice resolution of 512×512 and a slice distance between 0.62 mm and 3.0 mm. Some of the data contains pathologies (broken vertebrae, collapsed disc, scoliotic spines) as well as one cervical dataset from a child.

Experiments have shown that the thinnest intervertebral discs in the cervical section can still be distinguished if the slice distance is at least $1.5\ mm$. We therefore fixed the base volume voxel scale for this experiment as $1.5\ mm$ isotropic and the datasets were resampled accordingly.

In all datasets position and location of the intervertebral discs and the spinal column have been manually labeled. Based on the given annotation, positive samples have been generated

Volume	original size	normalized size
1	$512 \times 512 \times 202$	$106 \times 106 \times 134$
2	$512 \times 512 \times 163$	$113 \times 113 \times 108$
3	$512 \times 512 \times 361$	$144 \times 144 \times 168$
4	$512 \times 512 \times 222$	$89 \times 89 \times 148$
5	$512 \times 512 \times 249$	$170 \times 170 \times 166$
6	$512 \times 512 \times 152$	$91 \times 91 \times 101$
7	$512 \times 512 \times 277$	$245 \times 245 \times 184$
8	$512 \times 512 \times 260$	$244 \times 244 \times 179$
9	$512 \times 512 \times 1112$	$274 \times 274 \times 370$
10	$512 \times 512 \times 228$	$176 \times 176 \times 228$
11	$512 \times 512 \times 945$	$244 \times 244 \times 630$

Table 3.1: Properties of volumes for performance evaluation.

randomly inside the intervertebral disc and the spinal column. Negative samples have been generated randomly all over the volume with the constraint to have a minimal distance to positive samples of $10\ mm$.

Performance Evaluation

Time performance of the algorithm has been assessed based on a set of eleven CT volumes (six evaluation and five training datasets). The properties of the data, its original and normalized size is listed in Table 3.1. The classifier is trained using one cascading step and allow only intensity and Haar-like features in the cascade node. The influence of the different optimizations is measured against this default. Detection performance was measured based on 8 datasets containing the 6 evaluation datasets.

Limited Memory

The data structure is designed to cope with limited resources. However, reaching the bounds of memory provokes clearance of cache blocks that might have to be recomputed at a later stage of the algorithm. Figure 3.4 illustrates the time performance over different cache memory bounds for datasets $4 - 7$ and show a clear threshold ($\sim$ 25 MB) for all four datasets where the performance/memory ratio changes dramatically. This memory limit is slightly different for each dataset and depends on the dataset size. If the available memory falls below that threshold computation time rises heavily whereas performance remains stable if enough memory is available. The threshold marks the point where data blocks need to be frequently recomputed. As long as enough memory is available deletion of block data from the cache and occasional re-computation has almost no influence on performance.

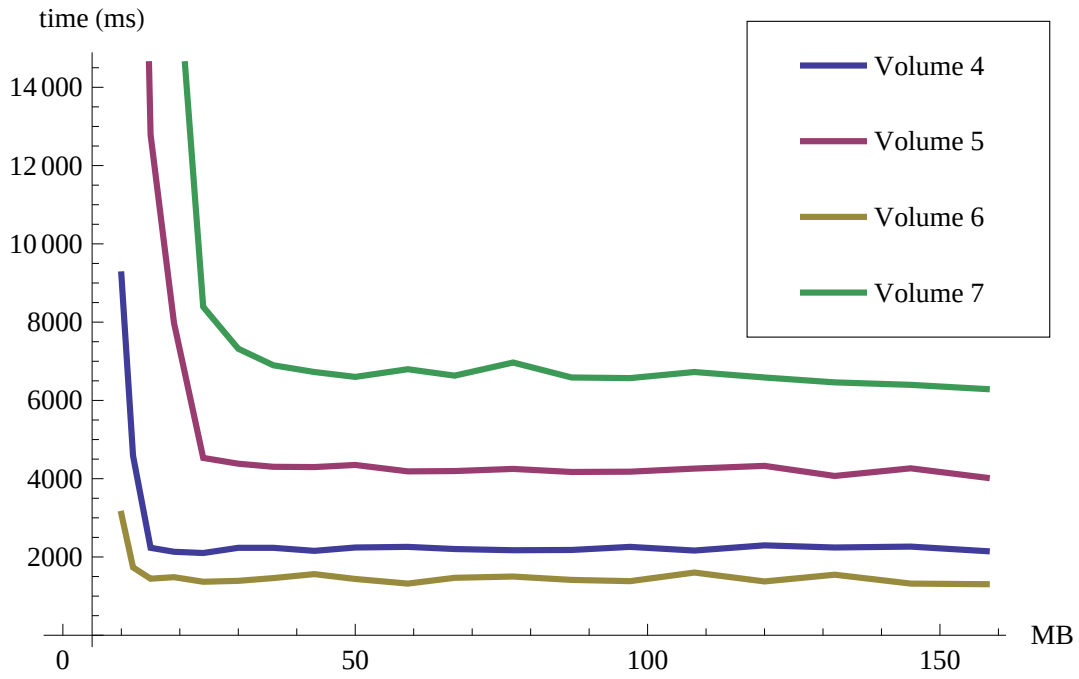


Figure 3.4: Memory Limits

Multithreading.

We tested the multithreading performance of our algorithm on an Intel quad core CPU with 2.4 Ghz and hyperthreading. Figure 3.5 plots the computation speed over the number of threads again on datasets $4-7$. Time drops until 4 threads are used. For more threads no significant speed-up (but also no significant slowdown) can be monitored.

The scaling with the number of threads below four is not linear. This is caused by the current locking strategy that prohibits accessing one block if it is currently computed by another thread. This situation mainly occurs if 2nd derivatives have to be computed that require accessing also neighboring first derivative blocks. If another thread is classifying one of these neighboring blocks at the same time it has to wait until the lock is released. This kind of collision happens more frequently as more threads are used. We expect therefore a logarithmic scaling of time performance with the number of cores as long as the locking behavior is not improved.

Cascading Speed-up.

The impact of cascading on detection speed has been measured by comparing the time performance of our default detectors with detectors which are trained without including a cascading step. The result is plotted in figure 3.6. Over eleven datasets we measured a speed-up of $1.45-2.67$ for detectors including a cascading step.

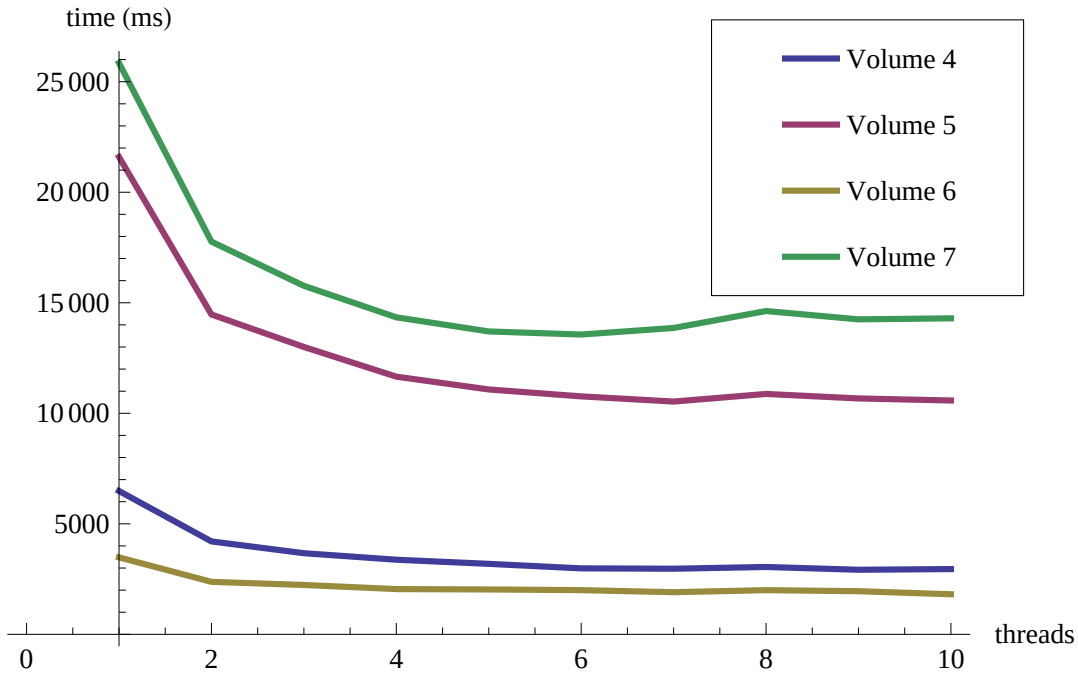


Figure 3.5: Plot computation time against number of threads. Tested on a quad-core with Hyperthreading.

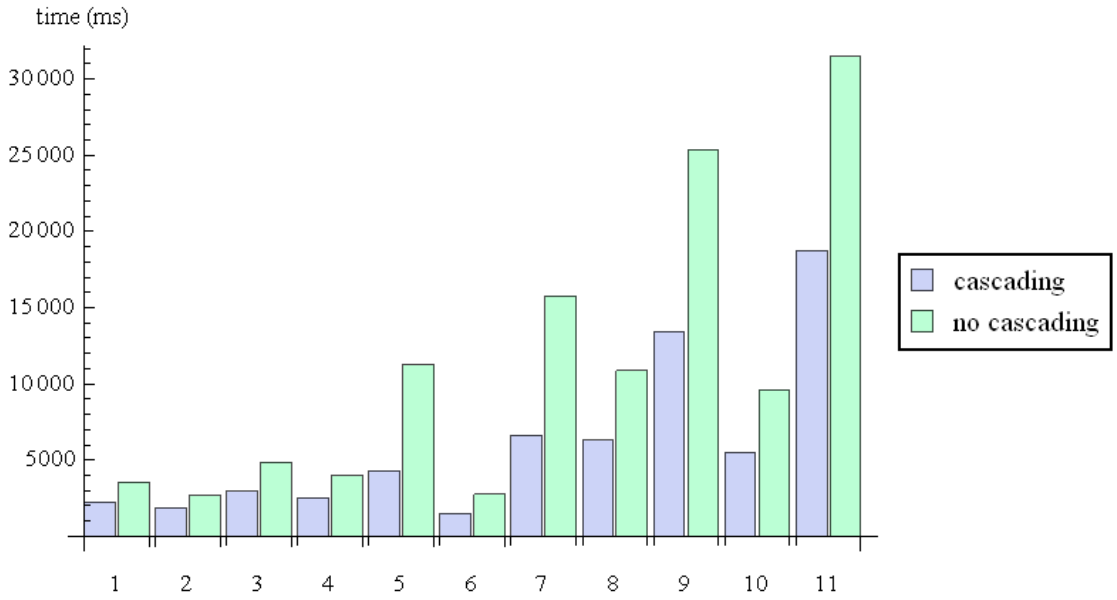


Figure 3.6: Detection speed comparison between PBT with (blue) and without (green) cascading on eleven different datasets.

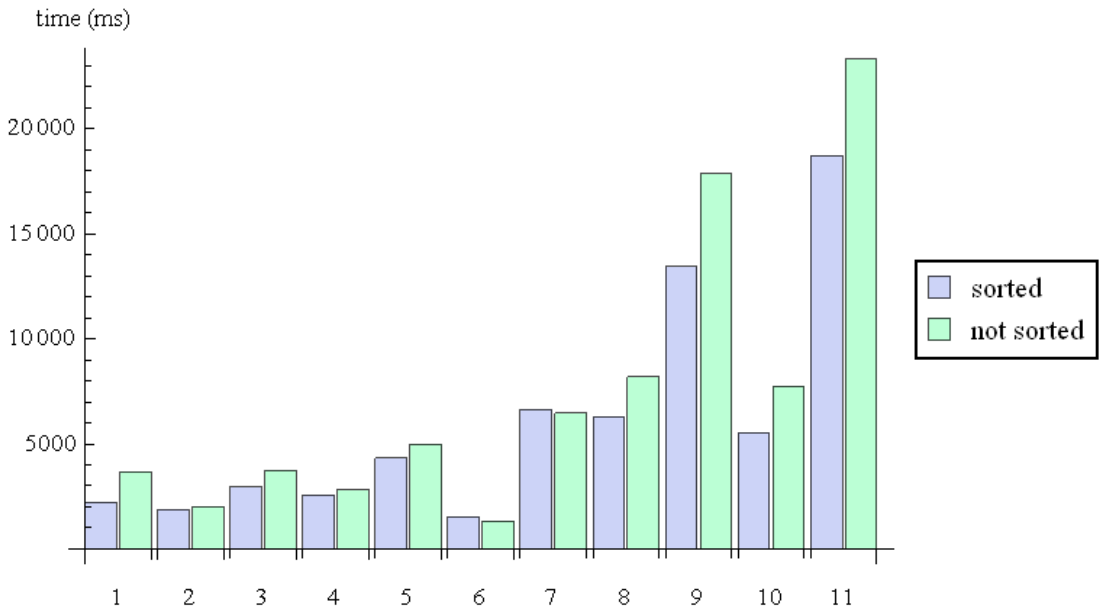


Figure 3.7: Detection speed comparison between PBT with (blue) and without (green) classifier sorting on eleven different datasets.

Classifier Sorting Speed-up.

The impact of classifier sorting is plotted in figure 3.7. We compare the time performance of our default detector including cascading and sorting with detectors which are allowed to use all classifiers in the cascading node. Classifier sorting results in a speed-up up to 1.65 for detectors which use sorting.

Multiresolution Speed-up.

To measure the impact of multiresolution feature detection we compared detectors using three levels of resolution against detectors using only one level. The results are plotted in figure 3.8. The measured speed-up ranges between 3.44 and 17.51.

Detection Performance.

Two feature detection results are depicted in Figure 3.9. The first row shows the detection of intervertebral discs in the lumbar section of the spine, the second row shows the detection of the spinal canal on a whole spine. The detection progress from lowest to highest resolution level is depicted from left to right. The images illustrate well the effectiveness of the multiresolution scheme since already at the lowest resolution level the major part of the volume is excluded from a higher resolution analysis.

The selected voxels (blue) reproduce the shape of the searched anatomical parts to a large extend. However outliers can be observed, for example inside the vertebral body (first row) or at

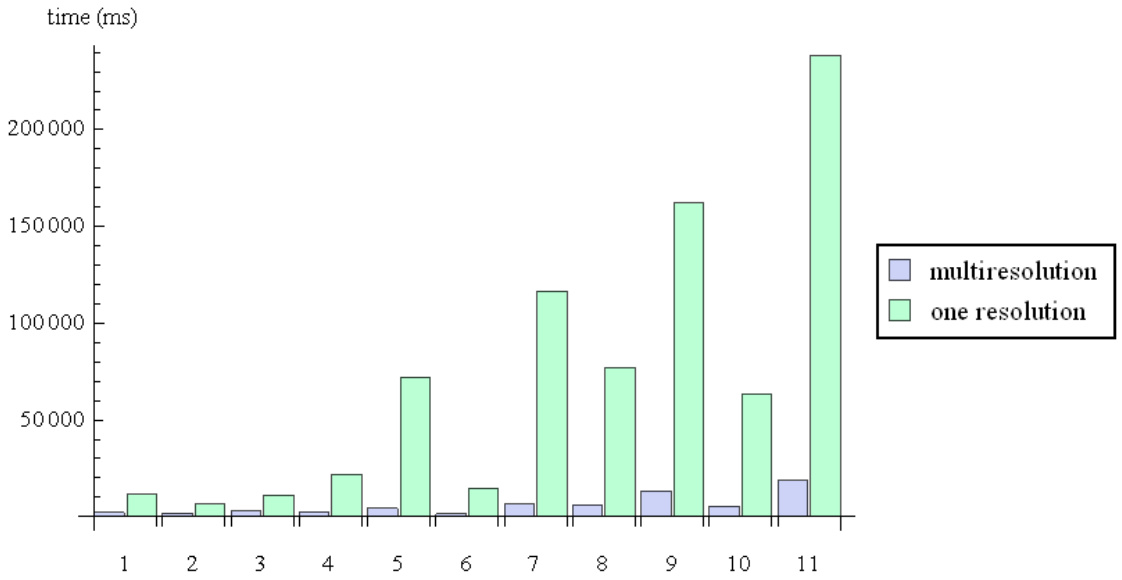


Figure 3.8: Detection speed comparison between multiresolution vs. one resolution.

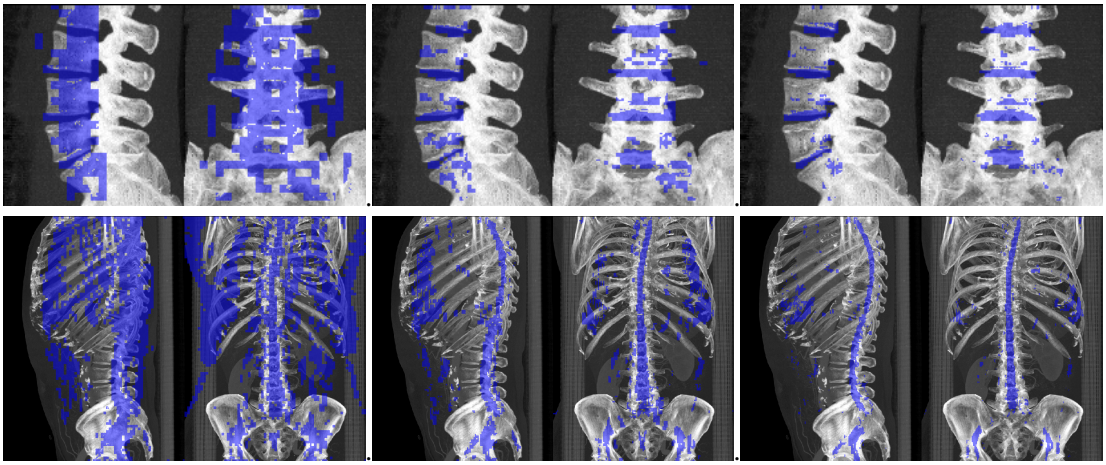


Figure 3.9: Coronal and sagittal images of detection results for the intervertebral disc (first row) and the spinal column (second row). Three levels of resolution document the detection process, lowest resolution left to highest resolution right.

the ventral side of the ribcage (second row). Moreover missing features can be observed as well (first row, ventral side of the topmost disc).

This observation is also reflected in recall and 1-precision plots (Figure 3.10). Recall denotes the ratio between selected voxels within the ground truth and all possible ground truth voxels. 1-precision describes selected voxels outside the ground truth divided by all the voxels which were selected by the feature detection (also the falsely selected ones). The evaluated data involves

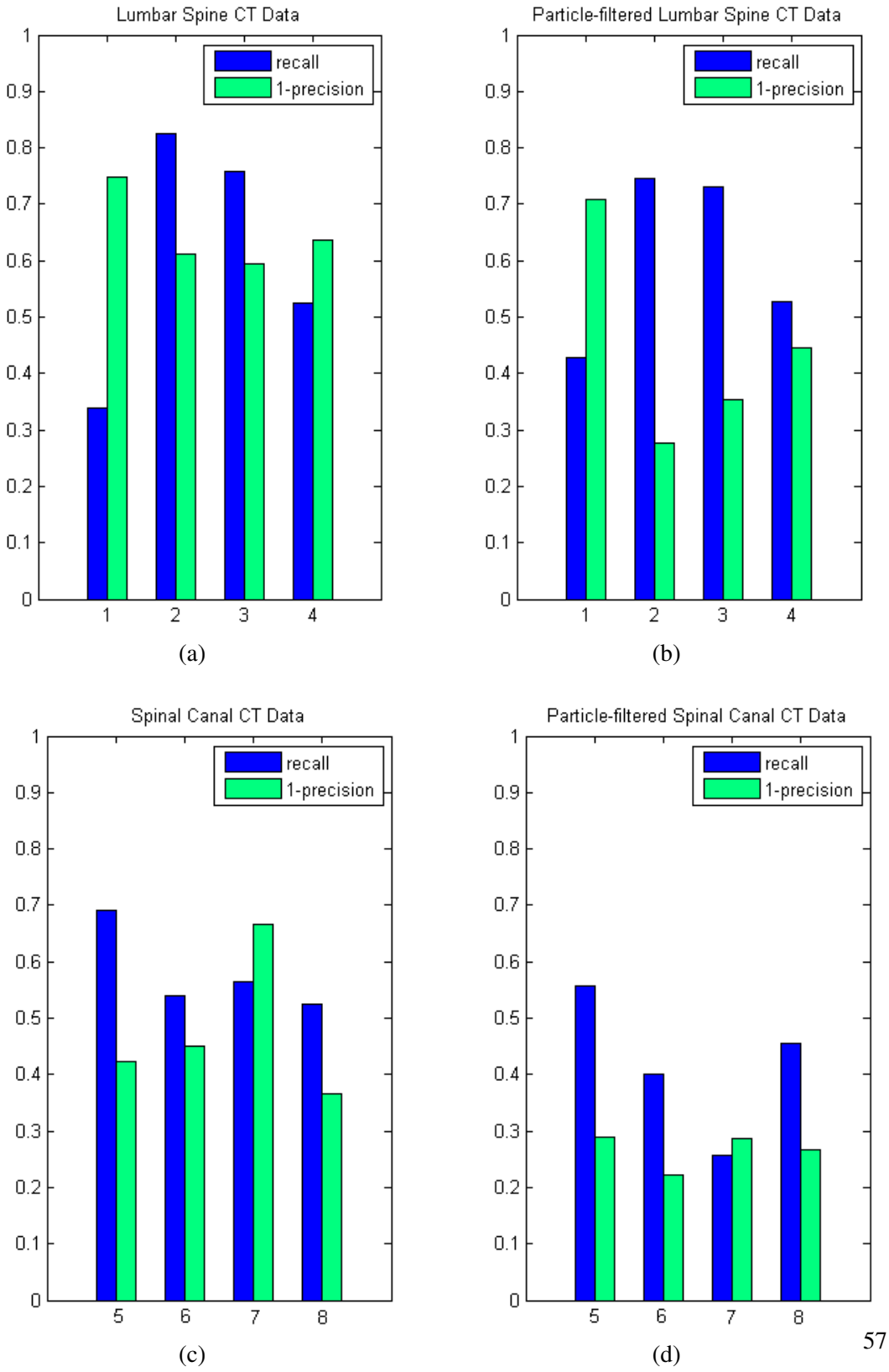


Figure 3.10: Recall and 1-Precision plots showing results lumbar spine dataset ((a) and (b)) and the spinal canal ((c) and (d)). Plots ((a) and (c)) show results without and plots ((b) and (d)) show results with post processing through particle filtering.

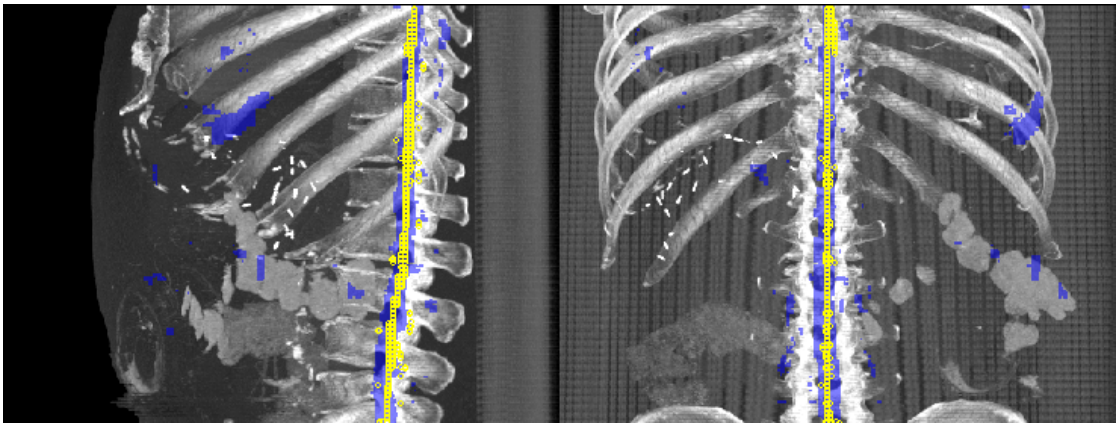


Figure 3.11: Feature mask (blue) post processed with particle filtering, non-maximum suppression and thresholding (yellow).

healthy spines (1, 2, 3, 6, 7 in Figure 3.10) and spines with diseases like scoliosis and broken vertebrae (4, 5, 8 in Figure 3.10).

The Figures 3.10 (a) and (c) show results after feature detection without any postprocessing where the high recall rates give information about good detection results of structures of interest (discs and spinal canal). However, besides the high recall rates there are also high rates of 1-precisions because of the occurrence of outliers (e.g., spongy bone within vertebrae with similar features to discs). The high 1-precision rates can be reduced by postprocessing steps by using the particle filter to assign probabilities to all detected voxels and removing all candidates with a probability value below a threshold $t = 0.15$. The results are depicted in Figure 3.10 (b) and (d). A significant drop of the 1-precision values can be observer due to a reduction of outliers. However, this action reduces the recall rate as well albeit much less significant. The threshold value t was manually defined, we found that $t = 0.15$ gives in most cases good results. However, since t has a direct influence on recall and precision, it can be used to tune the algorithm for specific use cases.

An example for postprocessing of the resulting feature mask is depicted in figure 3.11. First probabilities are computed by applying a box shape particle filter with the dimensions $9 \times 9 \times 60$ mm^3. The box approximates the elongated shape of the spinal canal. Second, the feature points are reduced by non-maximum suppression of the probabilities. Third, outliers are removed by thresholding the probability.

3.6 Discussion and Conclusion

We have presented a method for time and memory efficient feature detection on medical 3D volume data. The goals and requirements formulated at the end of Section 3.1 have been reached by selecting a classification based approach based on a Probabilistic Boosting Tree classifier. The classification method was improved by combining the decision tree with one cascading step and the introduction of classifier sorting. This classifier was embedded into a multiresolution

framework. We could show that all optimizations together result in a huge time performance gain with an approximated speed-up factor of 20.

Multithread performance was measured to scale sub linearly (almost logarithmically) which is due to internal data locking. The speed-up is for state of the art quad core CPUs still significant. But to benefit from more parallelism, improvements have to be done in this section. However it is likely that more sophisticated access patterns and locking schemes can help to overcome this problem.

The behavior of the block cache data structure was evaluated in section 3.5. It is noticeable that even larger datasets require only ~ 25 MB for the block cache to run almost unhindered. However even under circumstances where less memory is available the algorithm will just perform slower.

CHAPTER 4

Similarity Based Object Retrieval of Composite Neuronal Structures

This chapter is based on the publication:

> Florian Schulze, Martin Trapp, Katja Bühler, Tianxiao Liu and Barry Dickson. *Similarity Based Object Retrieval of Composite Neuronal Structures*. Proceedings of Eurographics Workshop on 3D Object Retrieval (EG3DOR), 2012 [118]

Circuit Neuroscience tries to solve one of the most challenging questions in biology: How does the brain work? An important step towards an answer to this question is to gather detailed knowledge about the neuronal circuits of the model organism Drosophila melanogaster. Geometric representations of neuronal objects of the Drosophila are acquired using molecular genetic methods, confocal microscopy, non-rigid registration and segmentation. These objects are integrated into a constantly growing common atlas. The comparison of new segmented neurons to already known neurons is a frequent task which evolves with a growing amount of data into a bottleneck of the knowledge discovery process. Thus, the exploration of the atlas by means of domain specific similarity measures becomes a pressing need. To enable similarity based retrieval of neuronal objects we defined together with domain experts tailored dissimilarity measures for each of the three typical neuronal sub structures cell body, projection, arborization. The dissimilarity measure for composite neurons has been defined as domain specific combination of the sub structure dissimilarities. According to domain experts the developed system has big advantages for all tasks which involve extensive data exploration.

4.1 Introduction

A mechanistic understanding of brain function must ultimately be built upon a detailed account of how individual neurons are organised into functional circuits, and how information processing within these circuits generates perception and behaviour. Genetic model organisms offer the possibility of applying powerful genetic methods to identify, characterise, and manipulate specific neurons in the brain. In particular, Drosophila melanogaster, the fruit fly, has emerged as one of the leading model systems for exploring how information processing in defined neural circuits generates complex behavioural patterns [96]. Central to these approaches are methods to reproducibly label and identify cells of a given type, and to construct digital atlases that ideally would include representations of each neuronal type on a common frame of reference. Molecular genetic methods make it possible to express transgenic markers in various neuronal subsets. In some cases, individual types of neurons can be labelled in this manner, though more often multiple cell types are labelled in each brain. Neurons marked in this manner can be visualized using confocal microscopy, resulting in multi-channel volumetric images. To be able to combine images of different fruit flies, i.e., to overcome slight anatomical variations and distortions and to provide a common reference frame, all images are co-registered to a template brain using non-linear registration [109]. Interesting neuronal structures are segmented on the registered images and their geometric representations are stored together with their source images and other meta information in a database.

Given the number and diversity of neurons in the fly brain, any systematic mapping of the individual cell types necessarily involves the acquisition, registration, and analysis of many thousands of images. Such constantly growing collections of interrelated spatial data build the basis for further knowledge generation and reasoning, creating the need for effective tools enabling the scientist to explore these large data sets. One urgent need is for a method for efficient similarity searches and 3D object retrieval, as well as robust measures for the classification of neuronal

morphologies. Given the representation of a specific type of neuron, or a component thereof, the scientist frequently needs to interrogate the entire database to identify other instances of the same neuron, or distinct neuronal types that share some but not all of its features. Such similarity measures can therefore also form the basis for automatic classification systems that could sort individual representations into distinct morphological classes.

We present a similarity based shape retrieval method tailored to the specific requirements of neuronal structures in the fly brain. The main contribution of this work is the definition of appropriate similarity measures for neuronal (sub)structures. These methods should be equally applicable to brain atlases for other species.

4.2 Related Work

Neuroscience is a data intense field requiring specialized and scalable data management, data mining and exploration methods. Data collections and studies in neuroscience are often inter subject, i.e., aim at fusing information from data retrieved from different individuals to a common atlas. A good introduction to this specific kind of image and object data collections and related challenges has been given by Van Essen [132] and Hanchuang Peng [101]. Van Essen described the emerging role of databases and atlases for neuroscience research, while Hanchuang Peng listed the main challenges of the new field of bioimage informatics as *clustering*, *classification*, *indexing* and *retrieval* of the data base contents.

Location or Euclidean distance based search for neuronal structures have been addressed by Bruckner et al. [15]. They described a system which allows to retrieve neuronal objects from an atlas by visual queries. The user marks a location or object of interest with a brush gesture in space and the system immediately returns a list of close (minimal distance) or overlapping objects. A similar method for exploration of pathways and connectivity of neurons has been presented by Lin et al. [72]. The framework offers a variety of tools which allow to combine several location based queries to retrieve connected objects or to identify neurons sharing the same pathways through the brain. Fly Circuit [89] is a web based database for Drosophila image and object data. It offers the possibility to search for neurons or cell bodies by similarity. Similarity is defined either by spatial distance in case of cell bodies or by a spatial distribution matrix in case of whole neurons. Non of the three mentioned methods addresses a shape or similarity based search of neurons.

A method for interactive exploration of neuronal pathways in diffusion tensor imaging (DTI) of the human brain has been presented by Sherbody et al. [121]. A set of regions of interest can be interactively defined and manipulated while the algorithm returns all fiber tracts connecting these regions. Besides such manual exploration of fiber tract data, clustering methods have been used to automatically identify bundles of similar fibers. Similarity between fiber tracts is often defined by their euclidean distance. Demiralp and Laidlaw [25] describe a weighted mean distance metric which favors the middle section of the fiber tract and use it for similarity coloring of fiber tract bundles. Moberts et al. [85] evaluated different clustering methods and reported that *hierarchical clustering using single-link* and *mean distance between fibers* gives the best results. These two methods explore the space of fiber tracts either by location, distance or connectivity. Appearance or shape is not directly considered.

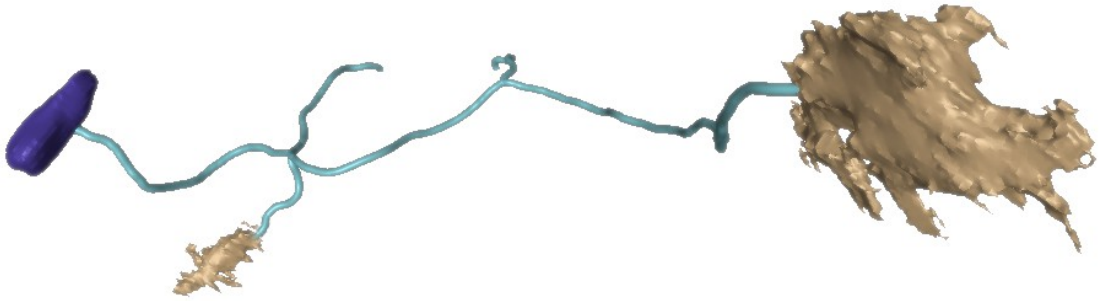


Figure 4.1: Rendering of an neuron as it is stored in the fly brain atlas. Blue = cell body, green = projection, brown = arborizations.

A different approach is proposed by Scorcioni et al. [119]. Neurons are characterized by more then 40 different metrics which describe the morphology of the structure. However, it becomes apparent that morphology is a very variable feature in our data. Algorithms which capture similarity in our data must be able to cope with variable morphology and partial matching shape.

Cardona et al. [18] presented a method which improves reconstruction of brain circuitry of the larval Drosophila by automatically assigning neurons to their respective lineages based on a shape based similarity measure and can be therefore considered as closest to our work. Unknown secondary axon tracts are automatically assigned to a lineage by matching them to previously labeled corresponding axon tracts. The proposed similarity measure for the tube like axon tracts is based on the curve morphing method of Jiang et al. [51] and relies on a combination of shape similarity, mean Euclidean distance and shape homogeneity.

Several general approaches for rigid and non-rigid shape retrieval have been proposed. Their discussion goes beyond the scope of this paper. For a detailed survey on shape retrieval we refer to Tangelder and Veltkamp [127]. To the best of our knowledge non of these methods has been applied up to now to realize a shape based retrieval method for similar neurons.

4.3 Data and Methods

The nervous system of the Drosophila as in any invertebrate organism consists mainly of unipolar neurons (for more informations we refer to Blankenship and Houck [10]). This means that from a nerve cell body only one process extends from it, which typically later bifurcates into a dendritic branch and an axonal branch. A typical example of a neuron as it is stored in the database is depicted in Figure 4.1.

The neuronal objects (cell bodies, projections and arborizations) are segmented on co-registered confocal microscopy images of the brain (see Figure 4.2 (left)). The volumetric image has a size of $420 \times 420 \times 165$ μm^3 and is sampled with a resolution of $768 \times 768 \times 165$ voxels.

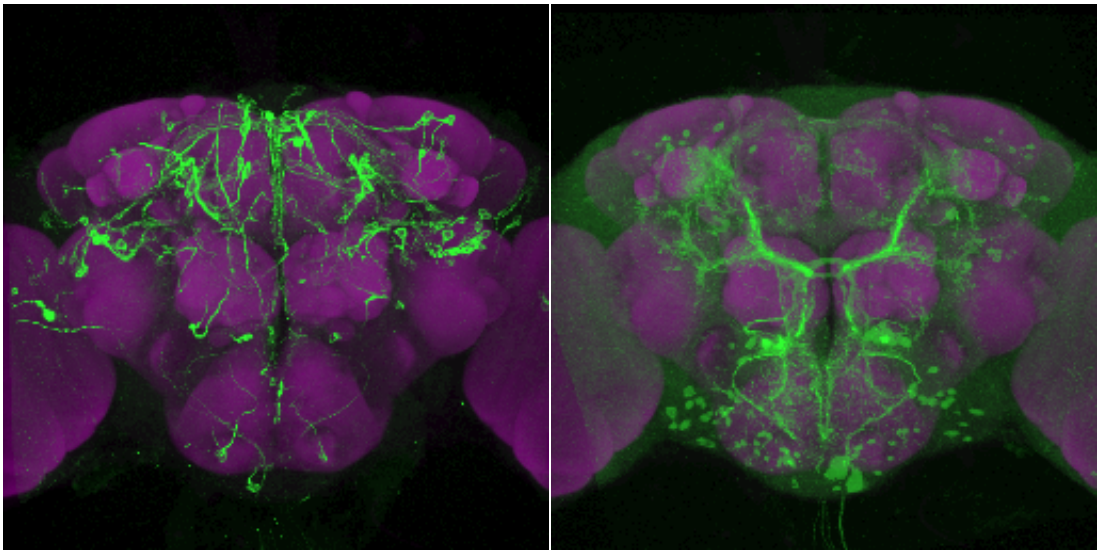


Figure 4.2: Maximum intensity projections of volume image data. Green channel: Fluorescence staining of neurons. Magenta channel: Brain template. Left, basic image. Right, averaged image based on five basic images.

Cell bodies and arborizations are annotated manually using a region growing tool on averaged image data (see Figure 4.2 (right)). The resulting binary masks are automatically converted into triangle meshes for rendering and further processing. Projections, semi-automatically traced [69], are thin elongated tree-like structures which are represented as skeleton graphs with radii.

The composition of one cell body, one projection and any number of arborizations form a neuron. Each neuronal structure belongs to exactly one neuron, but a neuron stored in our database is not necessarily complete, i.e., only a subset of the three components might be available.

Since all structures are segmented based on co-registered image data, the objects share a common reference frame and form an atlas, i.e., they are directly comparable based on their location in space. However the locational invariance undergoes a significant uncertainty. This stems from an average registration error $\approx 5\ \mu m$ and biological variability between individual flies.

In the following we describe how dissimilarity between neuronal structures is modelled and how dissimilarity between whole neurons is defined.

Cell bodies

Cell bodies (soma) are blob-like structures located within the cortex of the fly brain (see Figure 4.1 (left)). Shape and size varies heavily on segmented data mainly because cell bodies are floating structures on the cortex. Therefore, the shape, gathered from averaged images captured from several genetically identical flies, represents the density of the cell body position.

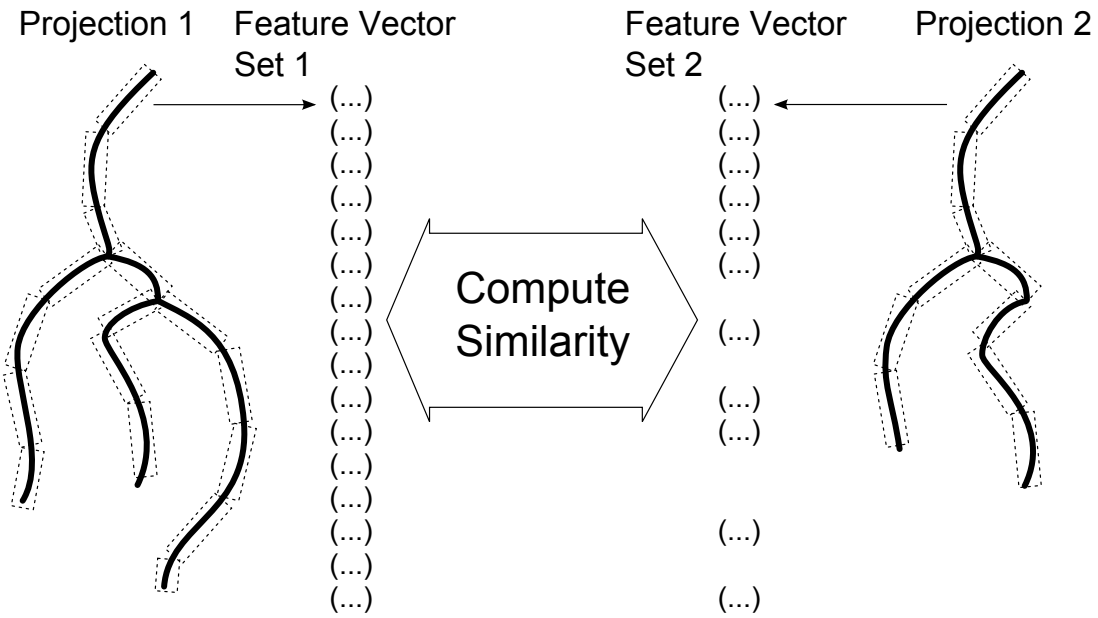


Figure 4.3: Transformation of a projection skeleton graph into a set of feature vectors.

The only discriminative feature of a cell body C is the location in space. Therefore dissimilarity can simply be computed using the Euclidean distance between cell body's center of mass $\mathbf{m}(C)$ and we define the dissimilarity function for two cell bodies C_a and C_b as follows:

$$D^c(C_a, C_b) = \|\mathbf{m}(C_a) - \mathbf{m}(C_b)\| \tag{4.1}$$

Projections

Projections (axons and dendrites) are thin elongated tree-like structures which are represented as skeleton graphs with radii. Discussions with domain experts revealed that the most characteristic features are location and shape, whereas other morphological features (e.g., number/ location of branches or terminals) are very variable and therefore misleading. Furthermore direct pairwise similarity computation is expected to be unrewarding and too expensive because of the expected growth of data. Therefore the presented approach aims to characterize projections by a small set of feature vectors which reflect the invariant properties of the structure and enable a fast dissimilarity computation.

The most descriptive feature of a projection in an atlas is the position and shape of its traces. The morphology on the other hand can be different between instances of the same neuron. Therefore we transform a projection into a set of feature vectors which describe the properties of subparts of the skeleton graph, but ignore the morphology. Projection skeleton graph P is split into n non branching sub-traces p_i. The length $|p_i|$ of the traces is a parameter of the algorithm and influences directly the number of feature vectors in the set. For each trace p_i a feature vector is

assembled consisting of two three dimensional vectors:

$$\mathbf{f}(p_i) = (\mathbf{m}(p_i); \omega_{\mathbf{d}} \cdot \mathbf{d}(p_i)) \tag{4.2}$$

where $\mathbf{m}(p_i)$ denotes the center of mass of p_i and vector $\mathbf{d}(p_i) = (d_x^i, d_y^i, d_z^i)$ denotes the main direction of trace p_i. In order to be invariant against the point order of the traces we demand that $d_x^i \geq 0$. If this is not the case the alignment is negated. $\mathbf{m}$ is defined relative to a normalized volume so that all components are in the interval $[0..1]$. The scalar $\omega_{\mathbf{d}}$ denotes a weighting parameter which defines the influence of the alignment on the feature vector. The whole projection is therefore described by a set of feature vectors:

$$\mathbf{F}(P) = (\mathbf{f}(p_1); ...; \mathbf{f}(p_n)) \tag{4.3}$$

Figure 4.3 shows an example where two different skeleton graphs should be compared. The comparison of both corresponding feature vector sets is not straightforward as the number of feature vectors varies and association between vectors is therefore undefined. Furthermore the distance measure should be able to detect partial matches.

Possible approaches to define a dissimilarity function for projections are for instance the *Bag of Words* [71] and the *Bag of Features* (BoF) [33] algorithms. We propose to use the *Pyramid Match Kernel* (PMK) [36] because we found that a multi-scale method adapts better to the variability of the domain specific data. The PMK method builds a histogram pyramid over the feature space. The resolution of each histogram starts by 1 for each dimension and is doubled on every higher level. The pyramid match kernel K for the feature vector sets $\mathbf{F}_a := \mathbf{F}(P_a)$ and $\mathbf{F}_b := \mathbf{F}(P_b)$ of projections P_a and P_b is computed as follows:

$$K(\mathbf{F}_a, \mathbf{F}_b) = \sum_{k=0}^{L} \frac{1}{2^k} N_k \tag{4.4}$$

with

$$N_k = I(H_k(\mathbf{F}_a), H_k(\mathbf{F}_b)) - I(H_{k-1}(\mathbf{F}_a), H_{k-1}(\mathbf{F}_b)) \tag{4.5}$$

, where H_k denotes the histogram at level k and I is a function computing the overlap between histograms. Finally the kernel function K is turned into the dissimilarity function:

$$D^p(\mathbf{P}_a, \mathbf{P}_b) = 1 - \frac{1}{\sqrt{c}} K(\mathbf{F}_a, \mathbf{F}_b) \tag{4.6}$$

where $c = K(\mathbf{F}_a, \mathbf{F}_a) \cdot K(\mathbf{F}_b, \mathbf{F}_b)$ normalizes the similarity value.

Arborizations

Arborizations are dense terminal branching structures which enable neurons to intercommunicate. It is important to determine the similarity of arborizations to figure out if they correspond to the same neurons. Similarity between arborizations can be defined by their shape. Because of the generation process similar arborizations tend to differ by small distortions or sometimes only parts are segmented. We decided to use shape context because it benefits from being insensitive

to small distortions and is easy to compute, yet still has a high accuracy. Originally proposed for 2D shape similarity [7], the shape context has also been generalized for 3D shapes [86].

For each vertex $\mathbf{v}_i, i = 1..,n$ of the mesh representation of an arborization A a coarse log-polar histograms H_i with $k = 1, ..., l$ bins of the connection vectors of $\mathbf{v}_i$ with all other vertices is computed. Thus, H_i describes implicitly the relative positions of all other vertices of the shape in respect to v_i.

$$H_i(k) = |\{\mathbf{v} \in A | \mathbf{v} \neq \mathbf{v}_i;\ (\mathbf{v} - \mathbf{v}_i) \in bin(k)\}| \tag{4.7}$$

The bins of the histograms $H_i, i = 1, ..., n$ used for the shape context of the shape are uniform in log-polar space to make the descriptor more sensitive to nearby sample points. The shape context descriptor is translation- and rotation-invariant, and can be made scale-invariant by an additional normalization step.

As arborizations consist of up to 800.000 vertices, and the runtime is dominated by the vertices, the general shape context is not appropriate for our application. Moreover, as sometimes only parts of arborizations are segmented, we also have to solve the problem of partial matching. Therefore, we propose to use the bag of features (BoF) based fast pruning algorithm using shape context (shapemes) by Mori et al. [86].

To realize BoF we have to obtain a vocabulary of geometric words $W = \{w_1, \ldots, w_m\}$ that is representative for the full set of shape contexts of the known shapes. We use vector quantization through k-means clustering for our purpose. The corresponding BoF histogram $H_W(A)$ will be obtained by counting the occurrences of the geometric words for the shape A.

Because the relation between the words is lost using BoF, we use a spatial-sensitive bag of features (SS-BoF) approach by Bronstein et al. [14] to improve the results. In the case of SS-BoF, the frequency of word pairs $f_{ij}(A)$ for spatially close shape context histograms of the shape A will be used as feature descriptor. We use diffusion distance [73] to measure the spatial distance. The resulting feature descriptor for the whole shape A is the $m \times m$ SS-BoF histogram $\mathcal{F}(A) = (f_{ij}(A))$.

In order to additionally strengthen discriminative word pairs, Bronstein et al. use the text retrieval inspired weighting proposed by Sivic and Zisserman [122]. Word pairs with a high frequency are less discriminative then those with low frequency, therefore spatially-close geometric words will be weighted by their *inverse document frequency*

$$\omega_{ij} = log\left(\frac{N}{n_{ij}}\right) \tag{4.8}$$

where N is the number of objects in the database and n_{ij} is the number of occurences of the word pair (w_i, w_j) over all objects.

In order to compute the dissimilarity function for two arborizations A_a and A_b, we simply compute the L_1 distance between the weighted SS-BoF histograms $\mathcal{F}(A_a)$ and $\mathcal{F}(A_b)$

$$D^A(A_a, A_b) = \sum_{i=1}^{m}\sum_{j=1}^{m} \omega_{ij}\,|f_{ij}(A_a) - f_{ij}(A_b)| \tag{4.9}$$

Neurons

As neurons comprise cell body, projection and arborizations, similarity between neurons is defined by the similarity of these components. Therefore, the similarity of each component has to be computed and the result set of each component has to be combined in a rank-aware manner to one single result set.

Rank-aware queries, also known as top-k queries [50], only retrieve the k objects that are highest ranked in the subqueries. For example, consider a top 5 similarity query on flags to a query flag in terms of color and texture. The top-k algorithm has to return those 5 flags that match best the criteria of the user in short computation time and determine a score value for each flag. An approach for fast processing of complex queries consisting of several subqueries has been presented by Güntzer et al. [39] based on the approach of Fagin [31].

To answer similarity queries on neurons our application has to return the top-k ranked neurons based on their subqueries. We use QuickCombine [39] for fast subquery combination based on an aggregating dissimilarity function.

The selection of a dissimilarity function $D(X, Y)$ that maps the distances of each of the subqueries to a single score value is crucial for good results. As mentioned by Grosser et al. [38] using domain knowledge for selecting a proper dissimilarity function improves the results of exploration algorithms.

Determining similarity between two neurons N_a and N_b based on their anatomy is a multi-step procedure. Neurobiologists first evaluate the similarity of the projections of the two neurons. If the projections are very similar it is very likely that the neurons also share an anatomical structure. In the next step, the similarity between the cell bodies and the arborizations will be determined. If those structures are also very similar it is very likely that both neurons are similar. Moreover if one of the two stages responds with a low similarity but the other one with a high similarity the neuron can still be of interest for the researchers.

Therefore we propose a structure-sensitive dissimilarity function between neuron $N_a = (C_a, P_a, A_a^1, ...A_a^m)$ and $N_b = (C_b, P_b, A_b^1, ...A_b^n)$.

$$D^{CA}(N_a, N_b) = \omega^C D^C(C_a, C_b) + \frac{\omega^A}{n+m} \sum_{i,j}^{m,n} D^A(A_a^i, A_b^j) \tag{4.10}$$

$$D(N_a, N_b) = \sqrt{\omega^P \; D^P(P_a, P_b)\, D^{CA}(N_a, N_b)} \tag{4.11}$$

The weighting parameters ω^C, ω^P and ω^A are used to define the influence of the respective sub structures.

4.4 Evaluation

The method was evaluated on a database containing 188 cell bodies, 451 arborizations and 346 projections. For evaluation we used the following parameters: Projection feature vectors are computed from $40\ \mu m$ long traces. The weight of the alignment fraction is set to $\omega_{\mathbf{d}} = 0.2$. The shape context of arborizations is described by $4 \times 4 \times 8$ (α, β, radius) log-polar histograms. The

vocabulary size is set to $m = 20$ geometric words and therefore each arborization is described by a 20×20 SS-BoF. For neuron similarity we used the following weighting parameters $\omega^C = 0.5$, $\omega^P = 1.0$ and $\omega^A = 1.0$. Cell body similarity is weighted by 0.5 because it is less discriminative than the other parts of a neuron.

Quantitative Evaluation: We asked the domain experts to select retrieval classes for 50 randomly chosen query objects from each of the projection, arborization and neuron collections. Cell bodies where not evaluated because dissimilarity is in this case defined only by Euclidean distance. Hence assembling a ground truth retrieval class would involve the definition of a distance threshold which recreates the ranking method and is therefore trivial.

The retrieval classes contain between one and 25 similar instances. The retrieval result was scored based on the manually composed ground truth data. Figure 4.4 shows the three resulting recall vs. precision plots.

We received unexpected good results for neuron retrieval which is due to the fact that the experts selected just very few neurons into the retrieval classes because of the high variability and the low number of currently available complete neurons. Fortunately these neurons could be retrieved with very high ranks. This demonstrates the performance of the sub-structure dissimilarity models as well as composition method. On the other hand the recall vs. precision curve of projection retrieval has relatively low precision values together with high recall rates. Despite that the retrieval performance is still sufficient, this shows that the emphasis for similarity rating from the expert's side goes beyond a pure geometrical definition, also knowledge about important pathways and anatomy plays a role.

Qualitative Evaluation. Domain experts evaluate the retrieval system and its performance in respect to the following use case: the assignment of new and unknown sub-structures (cell bodies, projections and arborizations) to already known neurons. The task usually either requires a very good knowledge of the data or involves lengthy manual search in the database. Therefore the problem becomes more and more complex as the database grows.

For this task the domain experts reported a substantial gain of efficiency compared to manual assignments. The similarity search narrows down the amount of data which has to be compared visually dramatically. Tests showed for all sub-structure types that appropriate results are almost always retrieved within the top 20 ranks.

The domain experts also assessed the performance of the neuron retrieval method. The biologists reported that the method retrieves and ranks neurons in a comprehensible way. Furthermore they expect that, with regards to the advancing growth of the database, neuron object retrieval will become an important tool which will help to keep the neuron database explorable.

4.5 Results

Results of different neuronal object similarity queries are depicted in Table 4.1. Query objects are in the first column followed by the top four result objects.

The first row shows an example query on cell bodies. As the only discriminative feature for cell bodies is the Euclidian distance between their center of mass, the query results are as expected.

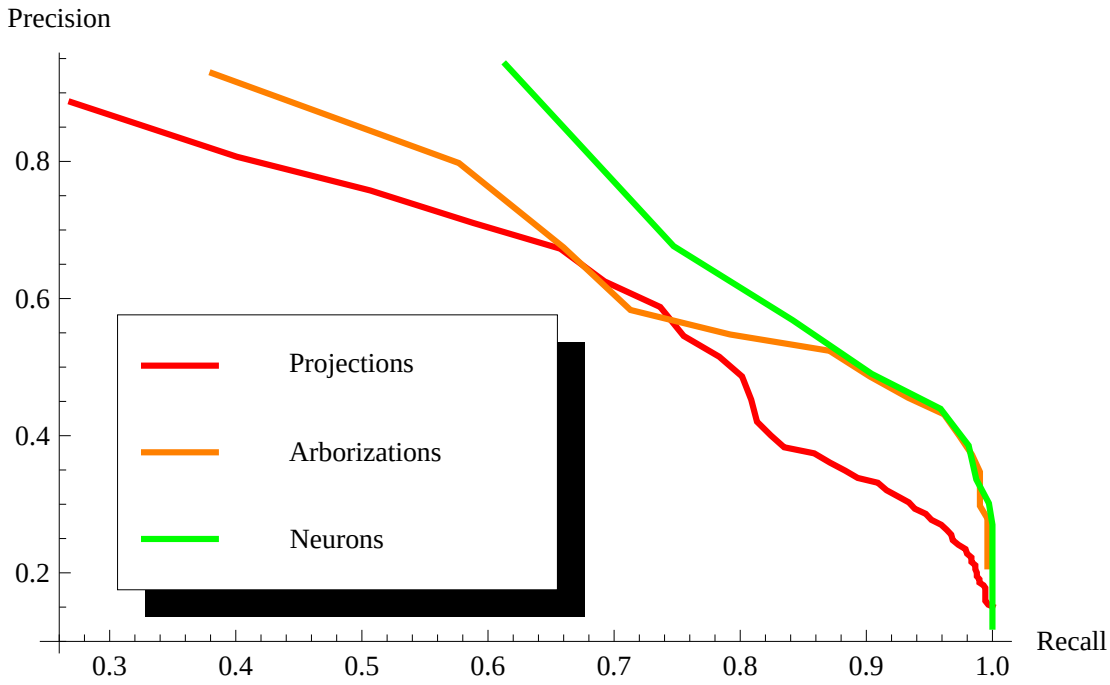


Figure 4.4: Recall vs. precision plots for projection, arborization and neuron retrieval.

Results for two projection retrieval cases are depicted in row two and three. The first case is relatively typical because the search results contain the three other instances of the same projection ranked as the top three results. In the second case the query is performed with an object that does not have any other instances but runs through a very common pathway. Rank one to three are set with completely unrelated projections and the rank four result is an instance of the same projection placed on rank one.

Results for two arborization retrieval cases are depicted in row four and five. The first case is based on unrelated geometrically very similar shapes, whilst the second case also contains partial matches in the top ranks.

The results for neuron retrieval are depicted in Table 4.1 row six and seven. The first case demonstrates that our approach retrieves neurons that are anatomically similar even if they are completely unrelated to each other. Moreover, in the second case the most similar neuron in the database is retrieved in the first place.

4.6 Conclusion

We have presented an effective object retrieval method for neuronal sub-structures as well as composite neurons. Our main contribution is the definition of dissimilarity functions for various neuronal structures which reflect the special needs in the domain of circuit neuroscience. Furthermore we defined a domain specific composition of sub-structure results which enables similarity based retrieval of complete neurons.

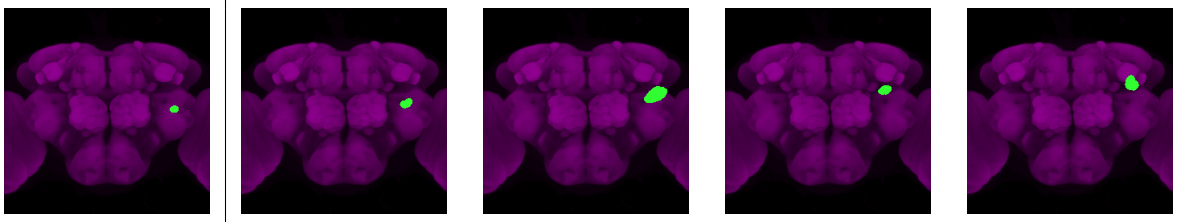
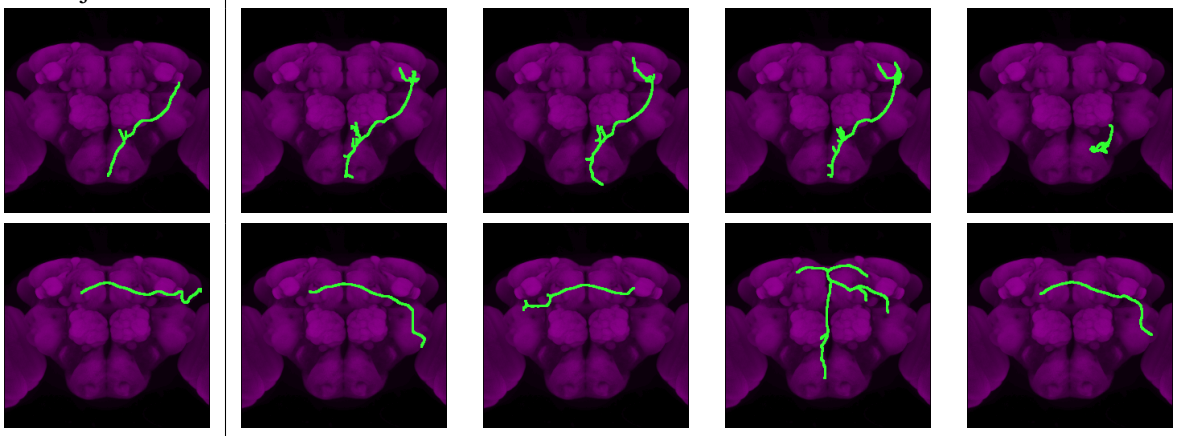
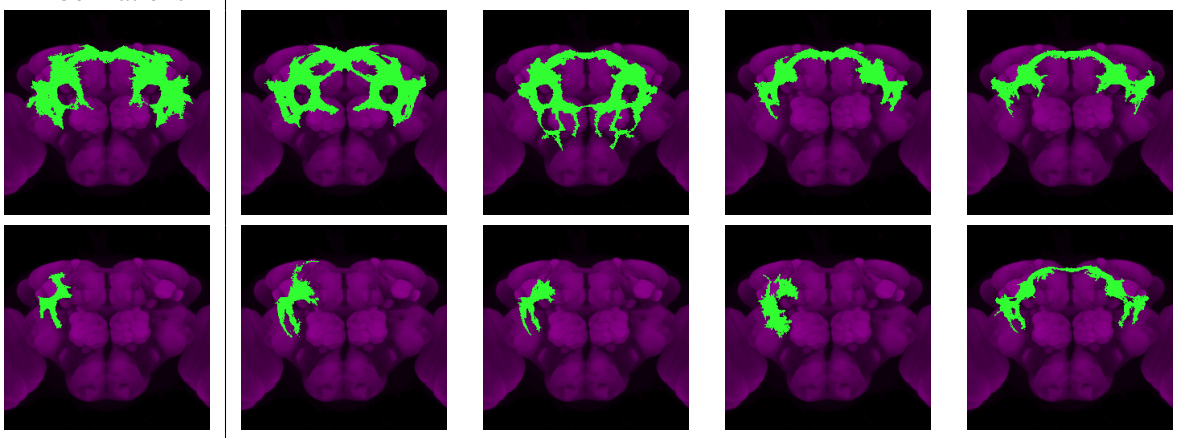
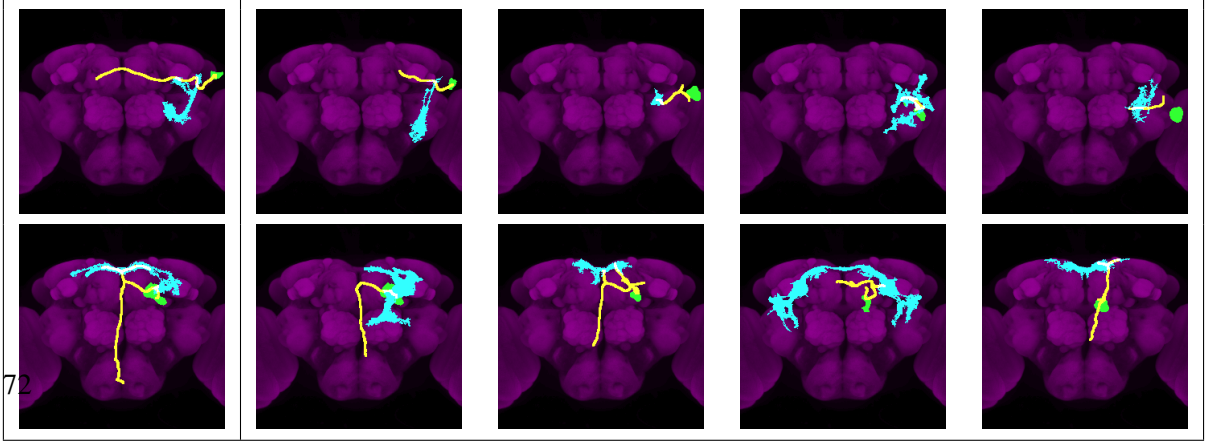
Query Object	1st rank	2nd rank	3rd rank	4th rank
Cell bodies				
Projections				
Arborizations				
Neurons				

Table 4.1: Result Images

As domain experts have reported, the retrieval results for neuronal structures are absolutely appropriate. Apart from the currently low number of admissible composite neurons this even applies to neuron retrieval.

Despite the already good performance, future work must be the further enhancements of retrieval performance. This involves further exploitation of domain knowledge for the discrimination process. Furthermore, the next logical step would be a semi- or fully-automatic labeling of neuronal structures based on the proposed retrieval system.

CHAPTER 5

Dynamic Spatial Indexes for Large Registered Image and Object Collections

This chapter is based on the technical report:

> Florian Schulze, Marcus Hansen, Katja Bühler, Tianxiao Liu, Barry Dickson. *Dynamic Spatial Indexes for Large Registered Image and Object Collections*. VRVis Research Center, 2012 [116]

Large collections of coregistered 3D images in combination with segmented objects forming an atlas are basis for research and knowledge discovery in circuit neuroscience. Modern imaging methods in combination with advanced genetic methods are able to visualize systematically the neuronal structure of Drosophila melanogaster.
The size of this collections however exceeds the number of datasets which can be explored manually. Especially the discovery of local structures either depicted in image data or as already segmented objects within the whole database poses a major problem.
We present a spatial data structure for indexing large amounts of three dimensional data and which fulfills the requirements of fast spatial coherent data access and incremental updatability. Based on this general method we develop two specialized implementations for our image and object data collections. Furthermore we demonstrate how the basic indexes can be used to build high level queries which allow to answer a multitude of data specific questions.
We show that the two spatial indexes enable efficient and interactive exploration of the image and object database which contains more than 22,000 entries.

5.1 Introduction

Understanding how the brain works is the question driving circuit neuroscience research. Using the Drosophila melanogaster as a model organism in combination with modern volumetric imaging methods permits a view into the flies brain and enables to study the neuronal structure. Advanced genetic methods are used to express single genes and therefore visualize subsets of the neurons. Registration of the images produced in this way enables direct local comparability and leads the way for atlas building and knowledge discovery. Current work focuses on systematically mapping tenth of thousands of gene expressions which result in large collections of high resolution 3D images. Based on this image data, structures can be segmented and semantically annotated to form an atlas of known neurons.

The resulting large image and object collection is the basis for further investigations. For the specific targeting of a certain neuron, it is crucial to know which genetic lines target which neuron. This can be learned by finding for a known neuron other images and objects which depict the same neuron but have a different genetic origin. Neurons have a unique distribution in space. Because of the spatial correspondence it is straightforward to find other images which have a similar signal at the same locations and objects whose location and shape are similar. For analyzing the function of neurons it is important to find possible interconnections between neurons. Since neuronal connectivity is established through synapses at the branching part of the neuron, a premise for connectivity is the spatial overlap between these parts of two neurons. Spatial comparability enables to search for objects which occupy the same space and therefore qualify as possibly connected neurons.

Solving the posed questions is theoretically simple because of the direct spatial comparability. Practically they constitute a huge problem because of the large number of datasets. Probing only a single region for images which provide an interesting expression is manually not feasible.

The implementation of automatic methods is therefore a logical step and straightforward to implement. However even automatic computation suffers from the large amount of data. Naive and sequential processing of the data collection to solve a certain location based question can easily take hours to compute on databases containing tenth of thousands of volumetric images. This can certainly be accepted if a limited number of queries can solve the questions. However it is definitely not feasible for exploration of the image and object space because this would require a high number of queries with big feedback delays.

Bio Image Databases and Altlases

The progress of imaging techniques suitable for biological research leads to a rising number of volumetric image data which has to be handled [101]. To take advantage of a large number of image data, it is inevitable to develop specialized databases where the data can be stored in. Especially databases for neuroscience are expected to be highly beneficial but also challenging [62]. A discussion of strength, weaknesses, and challenges of these systems is provided by the works of Toga [129]. Various efforts have been made to build databases for Drosophila melanogaster research. FlyBrain [4] combines schematic representations, image atlases and 3D object data. FlyBase [131] is a database for Drosophila research which incorporates image, genetic and molecular data while providing advanced semantic querying support. Interactive browsing and annotating 3D image stacks of FlyBase is realized with the Virtual Fly Brain project [99].

The combination of databases with atlases is considered to be of great importance [132]. For atlas building, image data must provide spatial correspondences of voxels which is established by non-linear image registration [109]. On the basis of this coherent image data, neuronal structures can be segmented to form an atlas. Specialized segmentation for neuron data have been presented by Lee et al. [69] and Peng at al. [102].

Spatial Database and 3D Datamining

Cellular expression in image data and therefore segmented objects are mainly classified by their location in space and their spatial relationships. Atlases make it possible that entities in the database are directly comparable by location. Therefore location based querying, browsing, and exploration of the database is an important feature of such database systems.

Spatial databases are a well known research topic in the field of geographic information systems (GIS) [41]. However GIS systems are designed for data which differs in many aspects from our data (e.g., it is not volumetric), which makes this methods not directly applicable.

An example for a specialized method enabling dynamic spatial queries is presented by Sherbody et al. [121]. In this method fiber tracts, segmented in a single diffusion tensor image of the human brain, can be retrieved by the definition of multiple regions where the structures must run through. Overlaps of fibers and querying regions are efficiently computed using a hierarchical oriented bounding box data structure. Herskovits and Chen [44] present a system which integrates image datamining support into a brain-image database. The system provides support for SQL queries which perform directly on image data. Therefore it is possible to query for images which have defined image characteristics in specific regions of interests. Another ap-

proach where the shape of the different brain regions is used to classify datasets in a brain atlas is described by Joshi et al. [52]. Each brain dataset undergoes a preprocessing step including skull stripping and sub-cortical segmentation, followed by feature analysis of the extracted entities. Based on this computations, the brain atlas can be visualized and clustered. In the field of circuit neuroscience Lin et al. [72] present a method for exploration and traversal of traced neuron data. Connectivity between neurons as well as a coarse spatial lookup table is stored in a relational database. This information is used to enable spatial exploration and traversal of neuron connectivity. Bruckner et al. [15] propose for a similar kind of data a visual query interface which enables the exploration of the object space by location and neighborhood. A space filling curve is used to optimize the access to a on disk lookup table containing location based distance informations to all objects within a given range.

Contribution of the Presented Work

In this chapter we present a method which dramatically improves the time performance of spatial queries to large co-registered image and object collections. The system presents results most of the time within seconds which makes it usable for interactive exploration of the data collection.

The presented method follows the same basic idea for fast spatial queries as presented by Bruckner et al. [15]. However the existing method is limited to object data and uses a static pre-computed index which is not suitable for constantly growing databases. The presented work aims to overcome this limitations. Our contributions to this topic are:

- The development of a spatial index data structure which permits fast spatially coherent data access and can be used for any kind of spatial data.
- The integration of a robust update mechanism which allows increment insertions of datasets.
- The application of the index for efficient exploration and analysis of object and image data.
- The implementation of a number of high level queries which solve use cases important to collections of spatially corresponding image and object collections.

5.2 Materials and Methods

In this section we describe a basic and general index structure for fast location based data access on a large database of co-registered images and related annotated objects. The approach centers around the idea that each entity in the collection has specific location based properties. The properties reflect the questions of the users to the data. On image data the users are primarily interested if a dataset has signal at a specific location because it shows that the gene which is expressed in the image is related with the neuronal structure at this position. On object data it is of interest whether the annotated structures are close to or if they overlap with a specific location. These properties are directly comparable between all datasets because all entries in the database are registered to a common frame of reference. An index providing immediate access to these

properties at a specific location for all datasets together, enables aggregation and exploration of spatial features.

This section describes the general structure of such a spatial index and how specialized instances for image and object data are constructed and which basic questions to the data we are able to answer with this indexes.

Input Data

The indexes and spatial query techniques described in this paper are optimized for data mining of co-registered 3D images and related segmented neuronal objects. The raw images have been generated using the GAL4/UAS system [12] to label and manipulate specific neurons in the fly brain and ventral nerve cord (VNC) [13] and confocal microscopy. They consist of two channels, one showing stained neurons and another one which contains brain morphology staining that is used to co-register all images to one common template image using non-rigid registration [109]. Examples for the use of this method are: Identification, visualization, and exploration of the olfactory memory [57], exploration of the neuronal circuit governing courtship behavior [142], investigation of sexual dimorphism in the fly brain [16] and reconstruction of a brain-wide wiring network consisting of up to 16.000 cells [19]. For a general explanation of the approach of investigating neuronal circuits using this method we refer to the work of Olsen and Wilson [96].

We currently work with two separate templates, one for the brain and one for the VNC. After registration the brain morphology channel is discarded. Only the channels showing neuron staining serve as input for our indexing and query pipeline. The image data cover a region of $420 \times 420 \times 165$ μm^3 and are stored with a resolution of $768 \times 768 \times 165$ voxels. The stained images also serve as a source for semi-automatic segmentation and labeling of neuronal structures. It is distinguished between three different types: cell bodies (blob-like structures located at the periphery of the brain), arborizations (dendrites, terminal branching structures which interconnect through synapses with other neurons), and projections (thin, elongated and branching structures which connect cell-bodies and arborizations). One cell body, one projection and several arborizations together form a neuron. These structures are depicted in the image data as regions with high staining. The segmented objects are stored as binary image containing the respective object mask for indexing purposes and as geometry for fast visual display. An additional object data type are the neuropils. Neuropils divide brain and VNC in a discrete set of functional compartments. The research on functional meaning of neuropils and the definition of the exact shape and location is an ongoing research topic. Currently 53 brain and seven VNC neuropils are defined. All object and image data is stored in a relational data base and features a unique id.

General Index Structure

Our spatial query space $V_T = \{\mathbf{v} = (x, y, z) | x = 0, ..., l; y = 0, ..., m; z = 0, ..., n\}$ corresponds to the discrete Euclidean 3D space defined by the voxel structure of the respective template image T, i.e., each template (in our use case brain and VNC) defines its own index. $N = (l+1)(m+1)(n+1)$ denotes in the following the cardinality of V_T. The set of spatial data on which we want to perform our queries is denoted with D. All datasets in D share the same

frame of reference V_T. M denotes the cardinality of D. D is in our application context either a set of images registered to template T or a set of objects segmented on previously registered images.

The index $I = (V_T, D, \phi)$ establishes a relation between space V_T and the query data D. I stores for each $\mathbf{v} \in V_T$ a list of relevant information on datasets in D found at location $\mathbf{v}$. The kind of information considered to be relevant is application specific and is defined by the *indexing function* $\phi(\mathbf{v}) := \{p_1, ..., p_k\}$, that extracts this information out of the datasets in D and compiles the list of results to be stored for each position $\mathbf{v} \in V_T$. The stored list of results is called in the following *payload* at index position $\mathbf{v}$.

Thus, specific index instances differ in the chosen template T, the query data D and the definition of the indexing function ϕ. ϕ has to been chosen in a way that the index delivers - in the best case - a direct answer to the question posed to the data.

Core Queries and Their Index Instances

We identified two core questions to our image and object data. The related indexing functions and index instances built the basis for the definition of a multitude of higher level queries in section 5.3.

Question one is "What objects are within a specified range r around a given location in space?" This results in a *Point Object Query* that delivers as result the set of all ids of objects within a specific range r around the input voxel. The related index instance is defined as $I^{obj} := (V_T, O, \phi^{obj})$. O contains binary masks of all available objects. The indexing function ϕ^{obj} generates for each position $\mathbf{v}$ a list of tuples containing an object id and the corresponding minimal point-to-object distance $\delta(\mathbf{v}, o)$. If a voxel is located inside an object $\delta(\mathbf{v}, o)$ is set to zero. We store only tuples within a predefined maximal range r around our input object.

The second question is "Which images have high staining at a specific location?" This is the so called *Point Staining Query* that expects as result all ids of images with high staining, i.e., intensity over a given threshold t, at this specific point. The related index instance is defined as $I^{img} := (V_T, G, \phi^{img})$ where G contains all indexed images. The indexing function $\phi^{img}(\mathbf{v})$ delivers the list of ids of all images which have a significant staining (Section 5.4) at position $\mathbf{v}$.

Figure 5.1 illustrates the use and the results of the core queries in our application context.

5.3 Higher Level Queries on Neurobiological Data

The basic data specific index instances deliver for each single voxel the information if there are any images or objects of interest and delivers the image id or the object id (and object-voxel distance) respectively. This section presents a collection of more elaborate queries based on the core queries that accept as input a whole set of voxels, a so called region of interest (ROI). The resulting set of core query results is fused by each of the defined queries in different ways into one single higher level query result. These queries allow to answer fundamental data specific questions which are presented as use cases for each of the introduced high level queries.

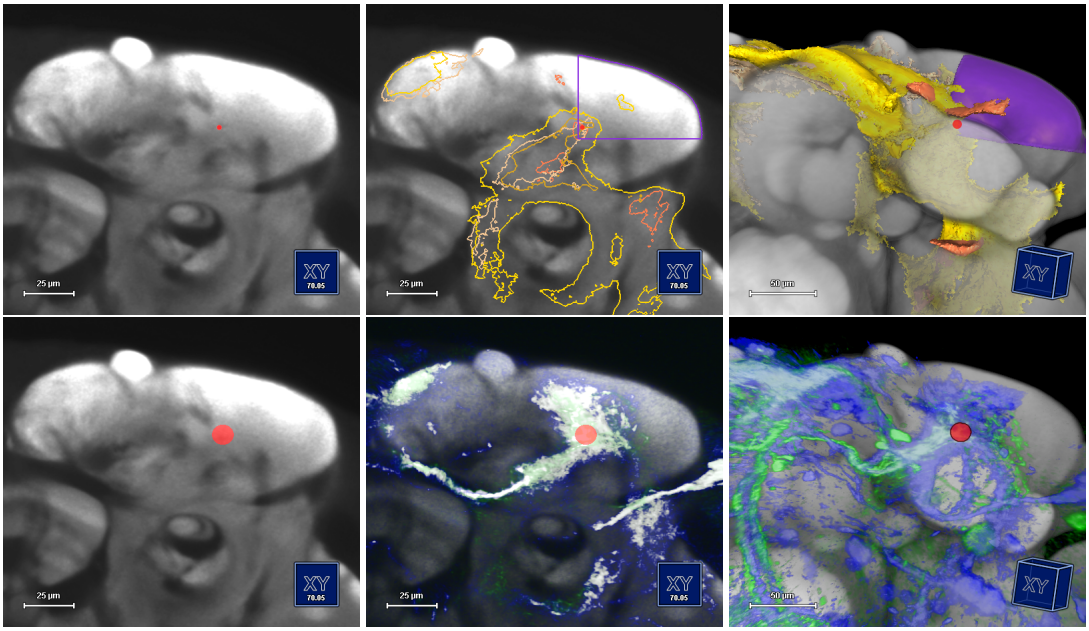


Figure 5.1: Examples of core queries. The upper row shows a *Point Object Query*. The query point has been defined on a 2D slice (left). A selection of objects ranked as top query results are shown as outlines in 2D and as surface renderings in 3D (middle and right). The second row shows a *Point Staining Query*. The query point has been defined on a 2D slice (left). The two top ranked images (rendered in green and blue) are shown in 2D and as volume rendering in 3D (middle and right).

Queries using the Object-Distance Index I^{obj}

The object queries allow to explore the object space of the data collection. The identity of neuronal objects is defined by their position in space and shape. Occupation of the same space by different objects can be evidence of an interconnection between neurons and therefore functional relationship.

Region Object Query

This query accepts as input an arbitrary region of interest R. It applies the basic Point Object Query to each voxel in the ROI and combines all tuples $(id_o, \delta(\mathbf{v}, o))$ in the resulting set of payloads to a global result set. The result set contains all retrieved object ids id_o associated with either the shortest found distance to object o or the number of overlapping voxels

The Region Object Query allows to answer a multitude of data specific questions depending on the definition of the input ROI and the filtering of the output objects. The following use cases illustrate application scenarios of the query tailored to our neurobiological data and related scientific questions.

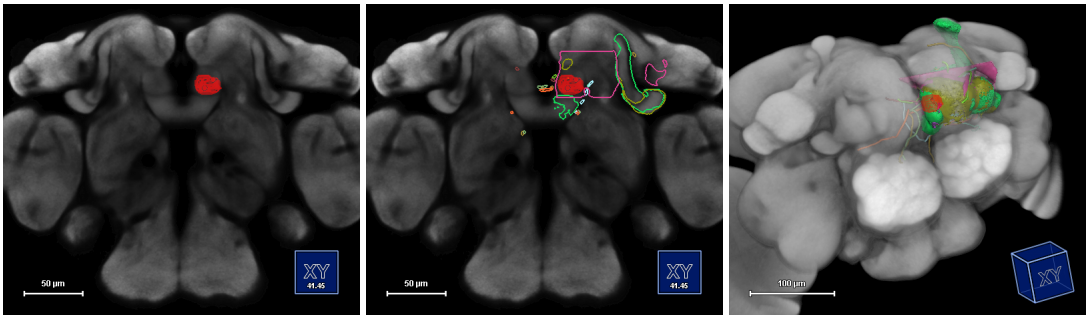


Figure 5.2: Use Case 1: Anatomically motivated region query. A query region has been marked on a 2D section showing the template image using a brushing tool (left). A selection of objects ranked as top query results are shown as outlines in 2D and as surface renderings in 3D (middle and right)

.

Use Case 1: Anatomically motivated spatial exploration of available objects. "Which cell bodies, projections, arborizations or neuropils (or any subset of them) are close to an arbitrary user defined region of interest?" is a common and frequent question to the data. Figure 5.2 shows an example of how the straightforward Region Object Query can be used to spatially explore the available objects in the database.

Use Case 2: Finding already annotated objects related to unknown structures in a given channel image. Regions of interest can not only be defined on the template image, but also in stained images. Marking regions of interest on staining images can be used to find for instance already known and annotated projections in the database which are close to a stained projection in a given channel image. Figure 5.3 shows an example for such a query, implemented as *Path Object Query* which is a special instance of the Region-Object Query that accepts as ROI an ordered sequence of points in 3D.

Use Case 3: Object based exploration of available objects. The question "Which Objects are close to another object?" is posed to explore the spatial neighborhood of objects or to realize for example a simple distance based similarity measure for cell bodies. This *Object Object Query* is an instance of the Region Object Query that takes as ROI the binary mask of the reference object and delivers neighboring objects and their minimal distances or overlapping objects and there overlapping voxel count (see Figure 5.4).

Use Case 4: Deriving functional information about available neurons. The overlap of arborizations and neuropils is an important indicator for deriving high level information on the function of the stored neurons. More specifically we have to effectively answer questions like "Which neuropils are significantly innervated by this neuron?" and "Which arborizations are overlapping, what is the extent of overlap and where does this overlap happen?" The *Object*

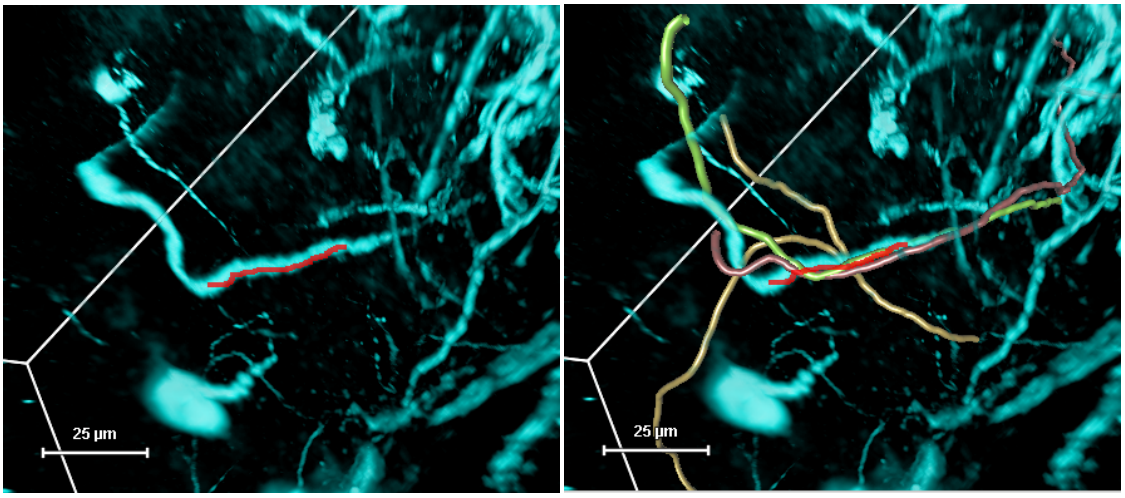


Figure 5.3: Path Object Query on staining image. The user has marked a structure on the 3D rendering of the image data by painting a path on it (left image). As a result the algorithm fetches all structures from the database within a small neighborhood to the path (right image).

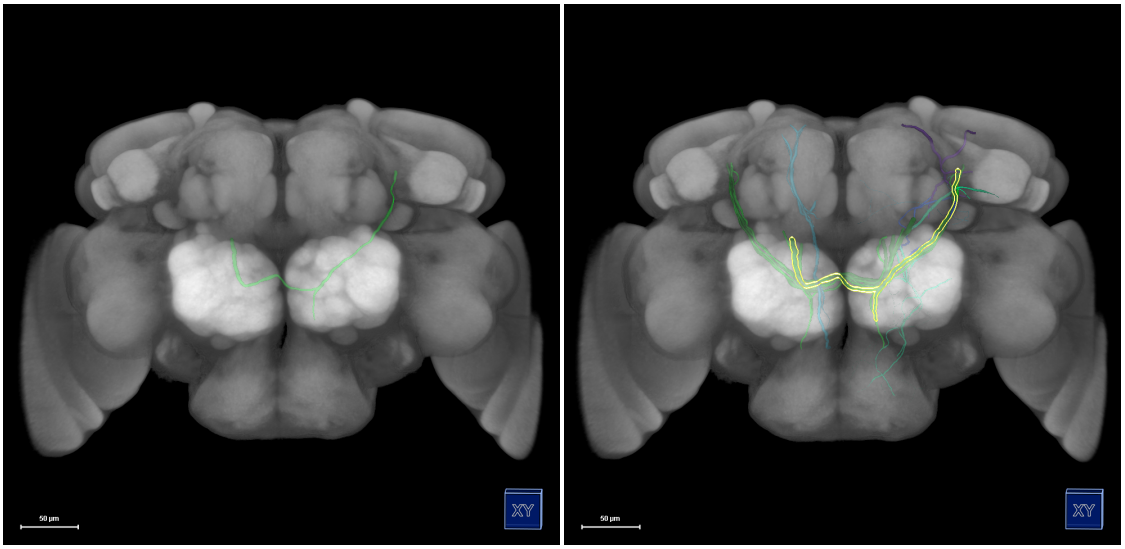


Figure 5.4: Use Case 3: We are using the projection shown in the left image as input region for the region-object query which results in an object-object-distance query. The results shown in the right images have been filtered to show only projections.

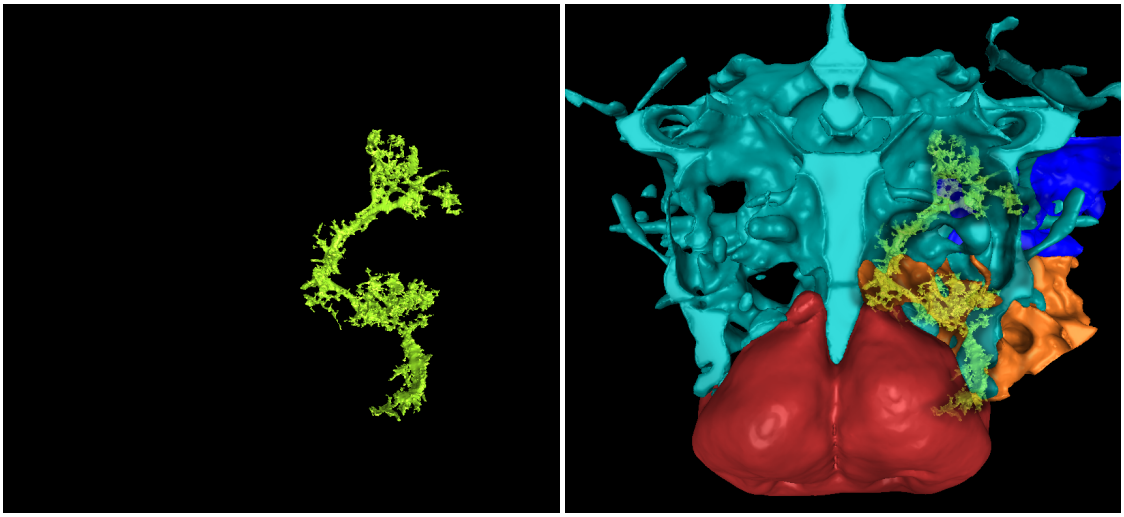


Figure 5.5: Use Case 4: An arborization (left) is used to query all neuropils (right) which are innervated by this neuron.

Object Overlap Query is a Region Object Query that accepts as input the binary mask of an object and delivers only objects which overlap with the query object. Figure 5.5 shows an example where all by an arborization innervated neuropils are retrieved.

Overlap in Region Query

For certain data analysis tasks it is necessary to quantify how much of the overlap between two objects resides inside another region. Let A and B be two overlapping objects and R be the region. The fraction $r^R_{A\cap B}$ denotes the overlap $|A \cap B|$ which is inside R:

$$r^R_{A\cap B} = \frac{|A \cap B \cap R|}{|A \cap B|} \tag{5.1}$$

Let ρ be a function which computes if position $\mathbf{v}$ is inside object o.

$$\rho(o, \mathbf{v}) = \begin{cases} 1 & \text{if } \phi^{\text{obj}}(\mathbf{v}) \text{ contains a tuple } (id_o, d) \text{ with } d \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

The overlap fraction r can be computed using the indexing function ϕ^{obj} with:

$$r^R_{A\cap B} = \frac{\sum\limits_{\mathbf{v}\in A} \rho(B, \mathbf{v})\rho(R, \mathbf{v})}{\sum\limits_{\mathbf{v}\in A} \rho(B, \mathbf{v})} \tag{5.3}$$

Use Case 5: Arborization overlap in neuropil. Overlap between arborizations give rise to the existence of synapses. Functional meaning can be inferred if it is known in which functional

compartments (neuropils) the overlap resides. For an arborization of interest overlap values to all other arborizations can be computed using the Region Object Query. For each overlapping object the distribution of the overlap with neuropils is computed using the *Overlap in Region Query*.

Queries using the Staining Index I^{img}

The staining queries allow to explore and mine the large amount of images stored in a database. Most of these images are not annotated. Mining this collection for images which show a similar global or local staining profile or specific structures is the main focus of the queries presented in this section. As for the object index we present several use cases to emphasize the practical value of our approach.

High Staining Query

The high staining query enables the user to search for images in the index which have high staining in a region of interest. Its principle is similar to the Region-Object Query presented in the previous section. The algorithm counts the number of appearances of each image ID in the given ROI and stores the result in a list containing tuples of image id and the related number of voxels with high staining in the given ROI. The amount of staining h_i^R of image i in a region of interest R is defined as:

$$h_i^R = \sum_{\mathbf{v} \in R} \sigma(i, \mathbf{v}) \tag{5.4}$$

Where $\sigma(i, \mathbf{v})$ decides if image i has high staining at position $\mathbf{v}$:

$$\sigma(i, \mathbf{v}) = \begin{cases} 1 & \text{if } id_i \in \phi^{\text{img}}(\mathbf{v}) \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Use Case 6: Search for channel images containing a marked structure in a reference image. One of the core questions to the collection of mostly not annotated images is "Is a projection, cell body, or arborization I see in this image also visible in other images?". The *Staining Brush Query* takes as input a ROI that has been defined by interactively brushing a region on the reference image and produces a list of all images in the database which have staining in that region. This region might be a region with high staining, but can be also any other region, e.g., on the template image. Figure 5.6 shows an example.

Use Case 7: Search for channel images showing a specific object. Another important question to the image data is "Is this known object present in other images of the collection?" This *Object-Staining Query* is similar to the previous one, but takes as input ROI the binary mask of a known object.

Use Case 8: Search for images with a certain level of high staining in a list of neuropils. The distribution of image signal over the set of brain regions (neuropils) can be used as a low

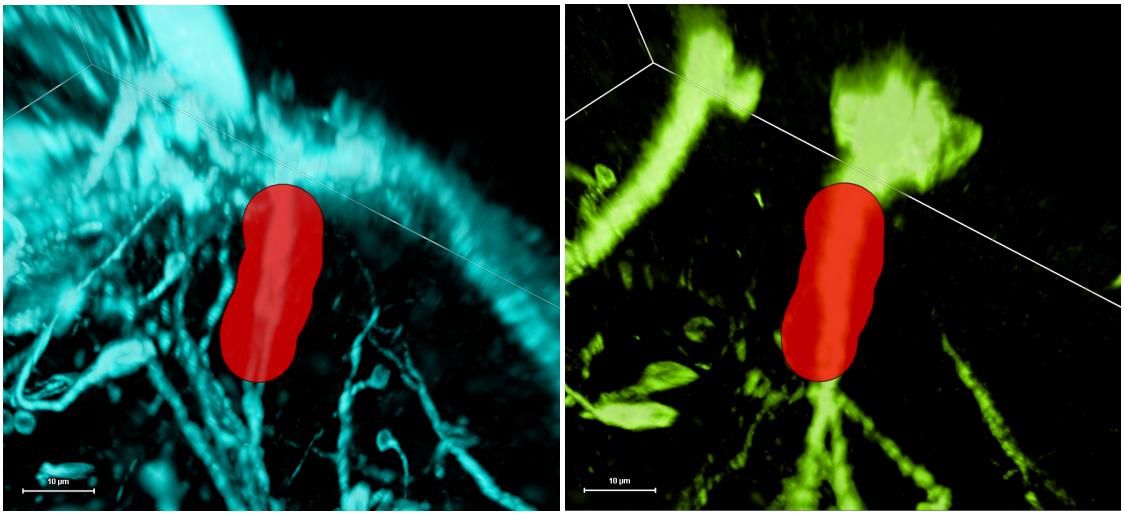


Figure 5.6: High Staining query: With brush tool a tubular structure in the center of the left image has been selected. The right image shows the dataset from the database with the highest amount of staining in the selected region.

level descriptor of the image. Querying images which exceed a certain threshold of staining in a set of neuropils is therefore a common image retrieval method. Staining ratios are computed by querying the number of stained voxels in the neuropils for each image using the high staining query. Subsequently the ratios are computed either by dividing the volume of stained voxels in the neuropil by the volume of the neuropil or by division with the volume of stained voxels in the whole image.

Staining Similarity Query

The High Staining Query presented in the previous section counts only the number of voxels with high staining in the specified region of interest. However having an interesting dataset already loaded, it can be of great interest to search the data for other images which show a similar staining expression in a ROI. We define similarity as the similarity between staining masks of image datasets. Therefore we can use Dice's coefficient to quantify similarity s between masks A and B:

$$s = \frac{2|A \cap B|}{|A| + |B|} \tag{5.6}$$

Applied to our index structure we can formulate the similarity function $s^R_{t,a}$ which computes the similarity between images t and a in ROI R as follows:

$$s^R_{t,a} = \sum_{\mathbf{v} \in R} \frac{2\ \sigma(t, \mathbf{v})\ \sigma(a, \mathbf{v})}{\sigma(t, \mathbf{v}) + \sigma(a, \mathbf{v})} \tag{5.7}$$

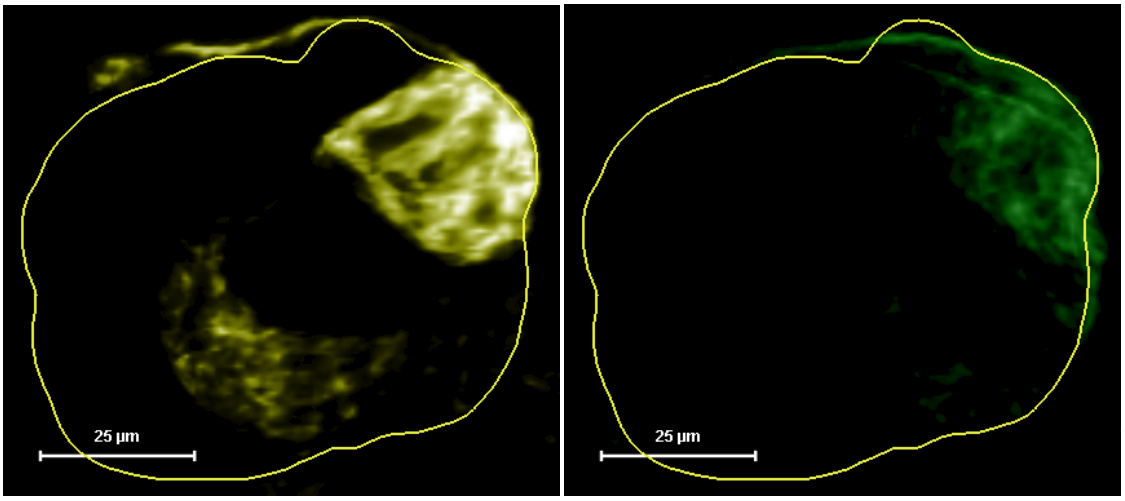


Figure 5.7: Staining Similarity query inside the left anntenal lobe (yellow outline). Left, base image. Right, best result image with Dice's coefficient ≈ 0.57.

Use Case 9: Find images with similar staining expression in ROI. The shape and position of a neuron defines its identity. Finding images with a similar staining expression to a template image in a ROI is a strong evidence that the images showing the same neuron. Figure 5.7 shows an example where the staining of an image inside the left anntenal lobe was used to query for images which are similar inside that region.

5.4 Implementation

The implementation of the theoretically defined index structure follows the requirements which are posed by the considered data and the defined use cases:

- **Data size**: The image and object database is expected to grow into the tenth of thousands of entries for both data types. In combination with the size of an individual dataset and its mapping in the index structure, it is obvious that no inmemory data structure can be used.
- **Fast access to coherent regions**: The previously defined High Level Queries operate on individually defined ROIs. The regions have in common that they cover generally spatially coherent regions. Querying is performed in an interactive fashion. Therefore it is required that the algorithm provides feedback in a short period of time (couple of seconds).
- **Dynamic Updatability**: The data collection is constantly growing. Therefore it is necessary to add support for incremental updates to the data structure.
- **Robustness & Reliability**: Data structures which operate from hard disks are exposed to the risk of disk failure of application crashes during write operations. Since corruption of

the data structure would implicate the expensive re-indexing of all data, it is necessary to take steps to provide an acceptable level of robustness and reliability.

- **Intuitive Interface**: The provided functionality must be accessible through a user interface which enables intuitive definition of the query parameters.

In the following we describe all aspects of the index implementation which are necessary to fulfill the presented requirements.

Data Preprocessing

A preprocessing step extracts the relevant information from the supported data types (see Section 5.2). Two parameters determine the amount of data which hase to be stored. First, the definition of relevance defines the ration of the voxels which are indexed. Second, the resolution of the index enables to balance between accuracy and data size.

Object Data: A distance map is computed from the object mask by applying a distance transform using the linear time algorithm proposed by Maurer et al. [79]. Positions are considered relevant if the computed distance is below a threshold of $40\ \mu m$ which is roughly a tenth of the edge length of V_T. For each relevant position an id and distance tuple is computed. Distance denotes in this case the minimal distance to the object. The data *item* is encoded in fife bytes, four bytes for the id and one byte for the distance. The object index is computed at a lower resolution than the original mask data of $384 \times 384 \times 83$. This leads to a lower accuracy, which is acceptable in our use case, but results in much lower memory consumption which compensates for a relatively high rate of relevant voxels of 5 - 10 $\%$.

Image Data: Voxels of the image data are considered as relevant if they contain a significant amount of staining. To extract the relevant voxels we perform image segmentation based on a combination of histogram based thresholding and binary morphology. This approach leads to a good separation of neuronal structures with effective noise suppression. A more detailed discussion about the used methods is beyond the scope of the paper. The segmented binary staining mask labels roughly 1% of the voxels. Because of the low number of relevant voxels we create the index at full image resolution.

Data Structure

The size of the index exceeds easily the amount of data which can be handled inside RAM. Hence the data must be stored out of core. Therefore we utilize a two level data structure which stores the actual indexed data in a large flat file on disk and uses a lookup table in RAM to locate the data of interest as presented by Bruckner et al. [15]. Crucial for optimal read performance is to keep the region to read as small and compact as possible. This requirement is addressed with two design decisions:

1. The payload for one position is stored in a consecutive space in the data file described by an *offset* and a *length*.

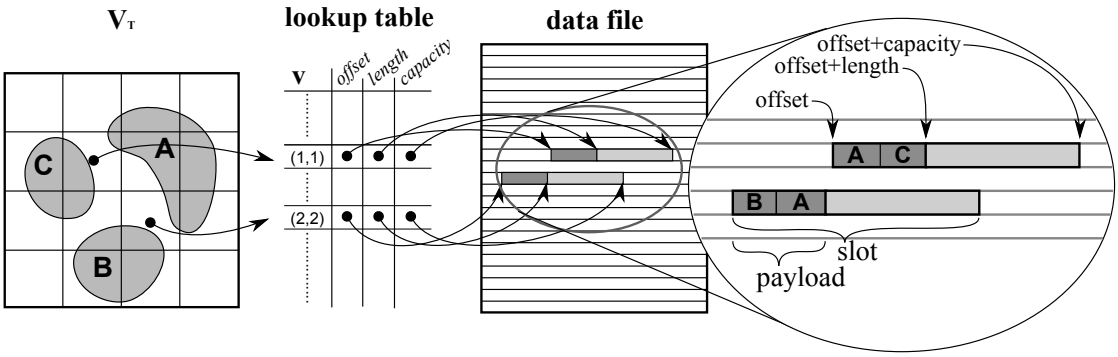


Figure 5.8: Data structure: Objects A, B and C contribute data to the index of V_T. For each discrete position in V_T the lookup table addresses the space where payload is stored in the data file.

2. Spatially coherent regions in V_T are mapped to the one dimensional data file in a coherency preserving way. More exactly we use a 3D Hilbert curve to map the 3D domain of V_T to the one dimensional data file.

The design is realized using two data structures (see Figure 5.8). First, the data storage which stores payloads on one large flat file on disk. Second, an inmemory lookup table which maps each position in V_T to an *offset* and a *length* in the data file. The mapping of spatial position to data file offset takes the Hilbert curve order into account.

Data Insertion

The original design lacked an efficient mechanism for inserting new datasets. However dynamic updatability is a crucial requirement since constantly new image data is acquired and additional objects are annotated. Therefore we propose an extension to the original data structure which retains the data access performance of the original implementation but provides the ability of an incremental data insertion.

To allow effective dynamic data insertion we implemented a *free space & resize* approach to overcome the actual limitations of standard file systems, that do not allow to insert data into an existing file. To realize this concept the basic data structures have been extended by adding a third parameter to the look-up table, the *capacity*.

The payload for a 3D position $\mathbf{v}$ is stored in a range of the data file starting by the *offset* address until the *offset+capacity* address. This data range will be called in the following *slot*.

New data is written to each slot starting from the *offset+length* address. Subsequently the *length* is incremented in order to include the written data into the payload.

If *length* becomes equal to *capacity* for at least on of the slots, no more data can be inserted without risking the overflow of this *slot*. Therefore all *slots* are resized by rewriting the whole data file. The resize procedure copies the payload for each position in original order into the new data file while increasing the amount of free space in each *slot*. At the same time the lookup

table is updated as well. The *offset* values are adjusted to the new addresses and the *capacity* value is set to the new slot size.

Data Access

Data access is the bottleneck of the algorithm because the data is located on a hard disk which results in high latency and relatively low bandwidth (compared with data access in RAM). Basically the index function $\phi(\mathbf{x})$ is implemented as a read operation on the data file at a specific range (*offset* to *offset+length*) which is defined by the lookup table for position $\mathbf{x}$. The locality preserving Hilbert curve mapping from 3D space into the data file causes that the data file needs only to be accessed on a relatively small and consecutive area which effectively reduces I/O overhead. The whole data access consists of three steps:
First, if a ROI is selected it is translated into a set of discrete positions inside the V_T bounds.
Second, the access order is optimized by sorting the access ranges by their positions in the file. This is important since the ROI is sampled using scanline conversion while the payloads in the data file are ordered using a Hilbert curve. Sorting the access order enables the algorithm to read the data from front to back in one consecutive pass.
Third, the data is read from front to back. Experiments showed that using the capabilities of the operating system to map slices of the file into memory is the most efficient way of doing this. Each read payload is handed over to a so called *evaluator-function* e which is responsible to aggregate the necessary information depending on the current kind of query.

Input: R : region of interest
Input: e : evaluator function $e : (\mathbf{x}, \{p_1, ..., p_n\}, S) \rightarrow S$
Output: S : List of results, type depends on the evaluator function
`// toDiscretePositions returns a list of discrete positions`
1 X := toDiscretePositions(R);
2 X_o := optimizeAccessOrder(P);
3 $S := \emptyset$;
4 **for** $\mathbf{x}$ *in* X_o **do**
5 $\quad S := e(\mathbf{x}, \phi(\mathbf{x}), S)$;
6 **end**

Algorithm 5.1: Data Access Algorithm

The access procedure is shown in algorithm 5.1. The *evaluator function* e is repeatedly called and updates in each iteration the result list S. Note that there is no possibility to control the order of processing by the evaluation procedure. Therefore the function must be invariant concerning the process order.

Evaluator Function

The *evaluator function* is responsible for aggregating the final result data. The function is called with the payload of all positions selected by the ROI in an unknown order. With each function call an intermediate result is updated. After all positions have been visited the final result is

computed by combining the intermediate results. In the following we explain the implementation of the evaluator functions for the high level queries *Region Object Query* and *High Staining Query* (Section 5.3) in detail, the queries for *Overlap in Region* and *Staining Similarity* are implemented in a similar way.

Input: $\mathbf{v}$: position
Input: P: payload at position $\mathbf{v}$
Input: S: previously aggregated result list containing tuples of object ids id_o and result values either distances or overlaps (id_o, δ)
Output: S: updated result list

```
1  for all (id_o, δ) ∈ P do
2      if (id_o, ∗) ∉ S then
3          if δ > 0 then
4              add (id_o, δ) to S;
5          else
6              add (id_o, −1) to S;
7          end
8      else if (id_o, δ_min) ∈ S then
9          compare the related stored distance δ_min with δ ;
10         if δ_min > 0 and δ_min > δ then
11             δ_min := δ ;
12         else if δ_min ≤ 0 and δ = 0 then
13             δ_min := δ_min − 1 ;
14         else if δ_min > 0 and δ = 0 then
15             δ_min := −1 ;
16         end
17     end
18 end
```

Algorithm 5.2: Region-Object Query *evaluator-function*

Region Object Query The purpose of this algorithm is to compute the minimal distance or overlap to all objects in the database. δ describes either the minimal distance to the object with id id_o or the number of overlapping voxels. To distinguish between both, overlaps are stored as negative values. At each function call (which happens for each position $\mathbf{v}$ in the ROI exactly once) result list S is updated with the current payload. See algorithm 5.2 for exact implementation details.

High-Staining Query This algorithm computes for all images in the database how much staining they have in a specific ROI. Therefore the evaluator function aggregates a set of tuples containing image ids and number of stained voxels. Each call of the function iterates over the list of image ids in the payload for the specific position. For each id it is checked if the result set contains already a tuple with the specific id. If this is the case the staining counter is incremented. If there is no such tuple, a new tuple with the counter set to one is inserted. See algorithm 5.3 for exact implementation details.

Input: $\mathbf{v}$: position
Input: P: payload at position $\mathbf{v}$
Input: S: previously result list containing tuples of image ids and numbers of stained voxels (id_i, ν)
Output: S: updated result list

```
1 for all id_i ∈ P do
2     if S has a tuple (id_i, ν) then
3         ν := ν + 1 ;
4     else
5         insert (id_i, 1) in S ;
6     end
7 end
```

Algorithm 5.3: High Staining Query *evaluator-function*

Robustness and Reliability

To ensure data integrity we implemented three methods which increase the robustness and reliability of the indexes.

Reliable Write Order: Adding new data to the index can cause a major risk to the data structure since it can not be avoided that the write process fails in the middle of the operation. However data is only written in the unused space in each data slot and the new data can only be accessed if the count field n_i in the lookup table is updated. To ensure that only data is accessed which was successfully and completely written to the data file, the update of the lookup table is delayed until the data has been completely written to the index file.

Journaling and Snapshot Restore: In order to restore data which might be lost because of a write error, we keep a journal file which logs the write history of the index. The journal file is altered in an *append only* mode which is expected to be the safest method to write to a file.

The journal consists of a header which holds the name of the current index and lookup table files as well as the previous (before last resize) set of files. After the header four byte words are added in append only mode. This words can be the id of the dataset which was inserted into the data structure or one of two special tokens. The special tokens are:

- *Saved*, which is written every time when the lookup table is saved on disc.
- *Resized*, which is written if the data structure has been resized.

If the data structure is initialized, first the journal file is read to discover which files hold the index and the lookup table. In a second step the journal is scanned for the last *saved* or *resized* token. If one of these tokens is at the very end of the journal, the data structure can be used as is. Otherwise all datasets denoted by the ids after the last *saved* or *resized* have to be inserted again.

In case of a major problem which has corrupted the current index, the journal is used to identify the files of the index before the last resize. The old version can then be opened and all datasets whose IDs are after the last *resized* must be inserted again.

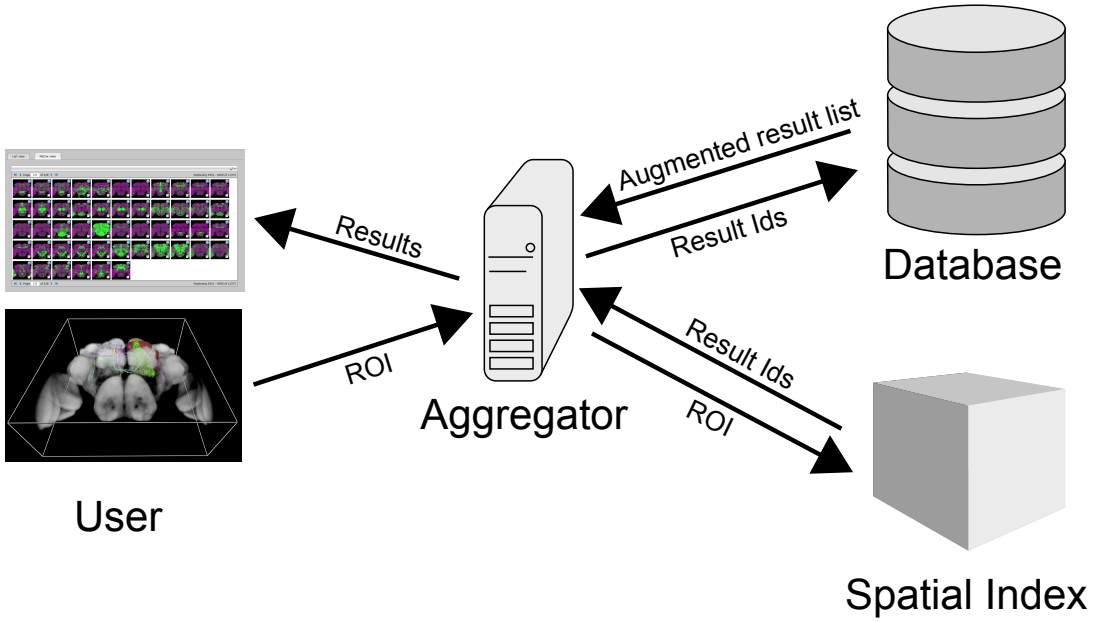


Figure 5.9: System overview: The user issues a query by the definition of a ROI using a visual interface. The ROI is forwarded to the spatial index which answers with the list of item ids and associated quantifications. The result items are augmented by a meta data lookup in a relational database. Subsequently the queried data is displayed to the user.

Secure Resize Resizing the payload table consists of a large sequence of error prone read and write operations. To ensure that the index contains the same data before and after resize a checksum is computed over all occupied data fields in the file that must remain the same after resize.

5.5 System Integration and Query Interface

The presented spatial index method is integrated into a larger client server setup. The user operates a client software which is able to visualize 3D object and image data. The querying tools are executed from within this visual interface. If a query is executed, the associated ROI is send over the network to a query server. This server forwards the query to the spatial index server which answers with a list of image or object ids combined with the query-depending quantification (e.g., minimal distance to or overlap with the object described by the id, or number of stained voxels which the image denoted by the id shows in the specified region). Subsequently the server retrieves additional data, such as name and preview image, for each result item by querying a relation database. Finally the sorted result list is send back to the user. Figure 5.9 depicts a schematic overview over the system.

The user interface is adapted from the original implementation by Bruckner et al. [15]. The

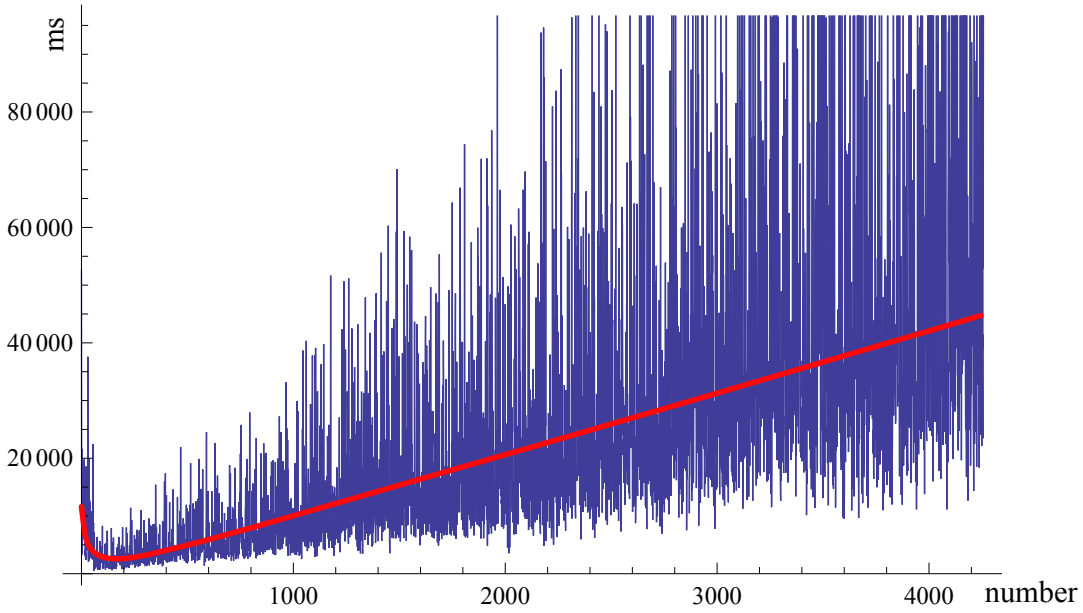


Figure 5.10: Timing: distance map insertion into the object index. X axis denotes number of inserted data sets, Y axis denotes time in milliseconds.

system allows to define query regions using visual query metaphors. It is possible to paint a path on any visible structure to define a *Path Query* or to select objects which thereafter define a region of interest for *Region Object*, *High Staining* and *Staining Similarity* Queries. To define arbitrary regions we added a so-called *Brush Query* which works similar to the *Path Query* but marks a region with an adjustable diameter.

5.6 Time Performance

Time performance is critical for the presented method. The system must be able to execute queries in a short period of time (within seconds) and it must be able to insert new datasets within a reasonable time enabling to keep up with the amount of new data. Performance of data access and insertion is decreasing while the index grows. Therefore it is important to measure performance over index size in order to estimate the scaling behavior of the system.

The insertion performance has been measured for both of the indexes. The results are plotted in Figure 5.10 I^{obj} and 5.11 for I^{img}. In both cases a linear growth of insertion time over the number of inserted data can be observed. Insertion of a distance map in index I^{obj} (Figure 5.10) takes a mean insertion time of ≈ 35 seconds if the index already contains $3000 - 4000$ objects. Insertion of images into index I^{img} (Figure 5.11) consumes about 100 seconds for an index which contains already 3000 images. The discrepancy to I^{obj} insertion timing is due to the higher resolution of the image data and the more global distribution of data over the data volume.

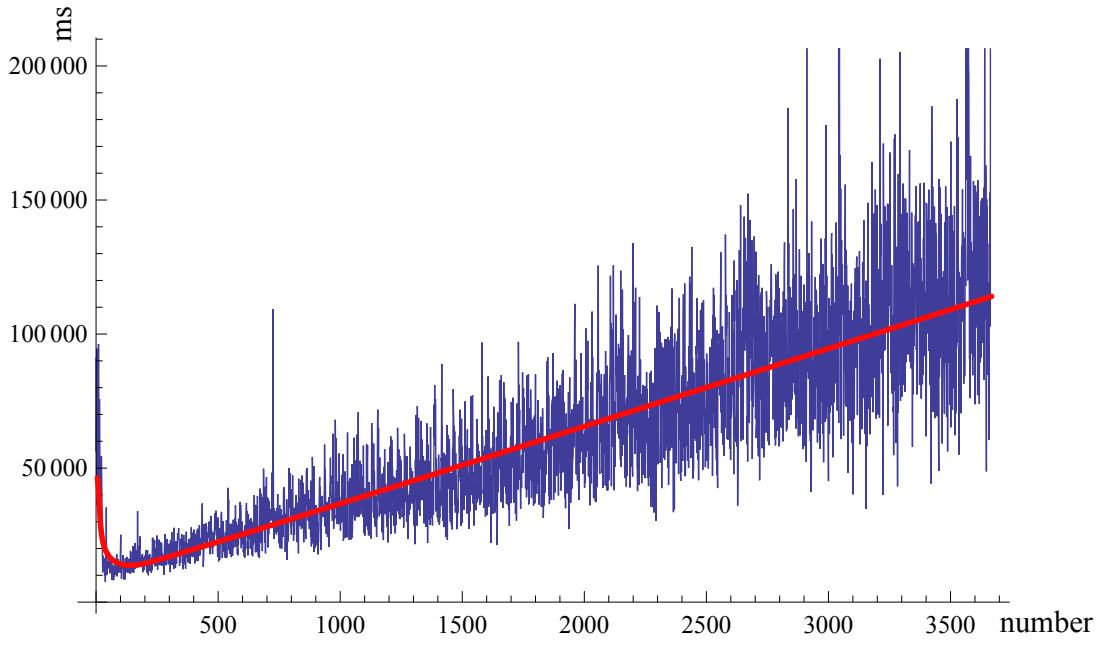


Figure 5.11: Timing: Staining masks insertion into the image index. X axis denotes number of inserted data sets, Y axis denotes time in milliseconds.

Insert performance is important because the system must be able to keep up with the incoming new data. However querying performance has a much more direct impact on every day work of the system users. Beside the core queries, which access only a single position in the index, we evaluate two different access patterns namely sparse sampled path queries and dense sampled region queries.

Sparse sampled access to the data structure is used for path queries on the I^{obj} index. We measured path query performance in terms of the number of accessed voxels versus the area which is covered by the path. The local access optimizing property of the data structure is reflected in the measurements depicted in Figure 5.12 computed over a data collection of 5000 entries. As long as the path is covering an area defined by a box with an edge length lower the 90 voxels, the access time is stable around 30 ms no matter how many voxels are accessed. However if the covered area becomes larger we observe escalation of access times where also the number of points becomes a linear influence factor.

Dense access to the data structure is used for region queries in both indexes. First we measure the general scaling behavior in relation to the size of the index. Figure 5.13 shows timings for region queries of a 32^3 area on the I^{obj} index. A linear scaling over the index size can be observed with an access time of approximately one second on an index containing 5000 entries.

The actively used image and object database contains currently more than 22600 images and 1200 segmented objects for the brain template. To measure the performance at this state we evaluated logging data acquired during the productive use of the system. On average region queries to the large staining index accessed 18716 voxels and executed in 3.8 seconds. Fig-

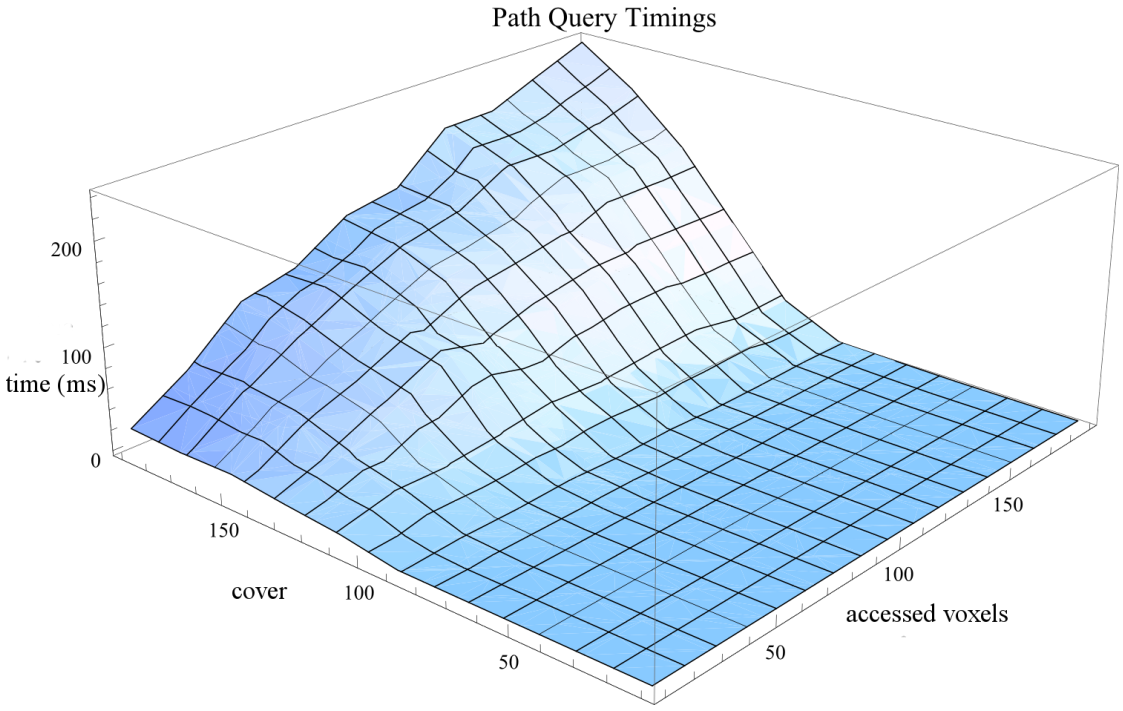


Figure 5.12: Access time for Path queries over the number of accessed voxels and the area which is *covered* inside the volume (*cover* denotes the edge length of an even sized box).

ure 5.14 shows the separate queries. Since the algorithm is I/O bound we don not measure a difference between high staining and similarity queries. The object index is currently an order of magnitude smaller which explains the lower access times we measure. Path queries usually access 5-25 voxels and are executed in an average of 200 milliseconds. Object to object queries need on average 1100 milliseconds. Insertions to the image index are executed in an average of 290 seconds.

The large variance which can be observed in all measurements has mainly three reasons: First, we measure mainly I/O performance (access time) which naturally shows a certain level of variance. Second, data which is inserted might show different spatial distribution which affects locality on the disk and therefore write performance, and explains why some datasets can be inserted faster than others. Third, data access is usually limited by a compact region of interest. Therefore the occupancy of the index at this region influences the amount of data which must be read and therefore the access time.

5.7 Discussion and Conclusion

In this paper we presented a method for spatial indexing and its application for large 3D object and image collections. The method is used for retrieval of images and objects based on local image expression and object neighborhood. The same data structure is used for the computa-

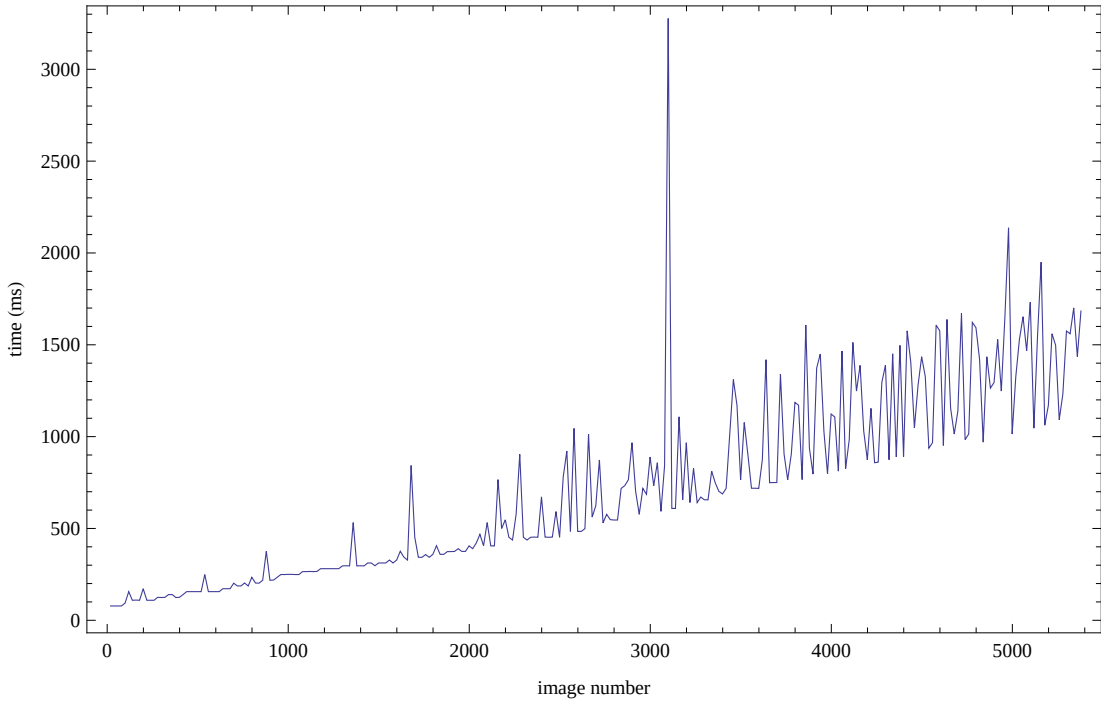


Figure 5.13: Access time using a 32^3 box as ROI. Access time scales linearly with the number of inserted images.

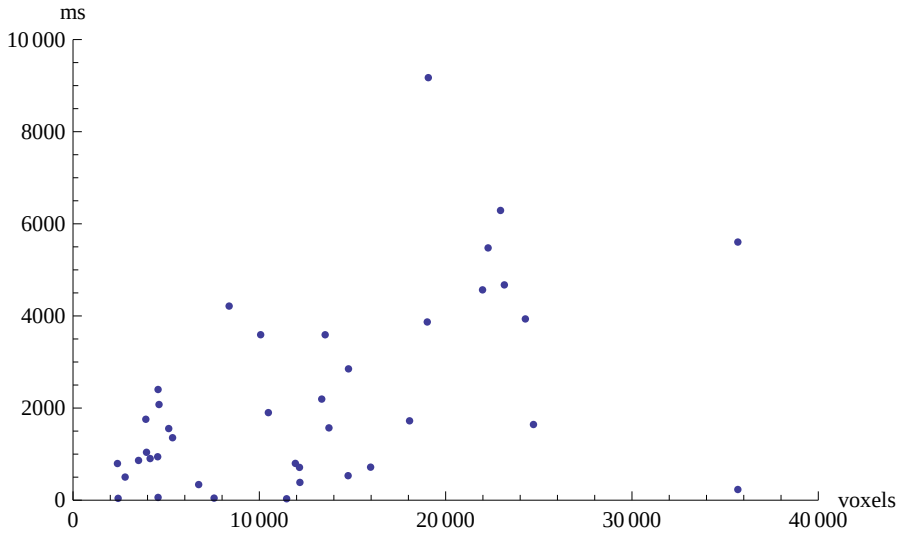


Figure 5.14: Region queries to the image index filled with more than 22600 images logged through a usual exploration session on our production installation. The plot shows access time in milliseconds against size of the region in voxels.

tion of additional information like overlap between objects and similarity between local image regions. Furthermore, the systems provides a robust method for incremental insertions of new dataset which allows to cope the reality of data acquisition in circuit neuroscience research. The evaluation shows that the algorithm computes results in a short period of time which enables the application of this method for interactive exploration of the image and object database.

The most similar method for the analysis image databases is the BRAID system presented by Herkovits and Chen [44]. BRAID can be used to answer similar questions to the data as our methods. However Herkovits and Chen give no reason to believe that their system can deal with similar large image collections with a comparable performance because they rely on strictly sequential processing of individual images.

Lin et al. [72] presented the "Neuron Navigator" which serves a similar purpose of exploring the object collections as our method. However their approach has two drawbacks compared to ours. First, it relies on connectivity information, which is not naturally given. In fact, the investigation of the connectivity is one of the research goals where our system is used for. Second, the spatial search is implemented through a lookup grid with a relatively low resolution which yields to a low accuracy and and prohibits the accurate computation of overlaps.

Quality and Accuracy

The presented method enables to evaluate and quantify local properties in spatially coherent regions over larger data collections in a very efficient way. Since we apply this method to 3D image and object data, it is primarily used to retrieve datasets of interest from the collection. Image and object retrieval is a purposeful process, the user expects the system to deliver exactly the datasets which match the defined search criteria. Therefore it must be discussed which parameters have influence on quality and accuracy.

The presented method is primarily responsible for fast retrieval of the requested information. The quality of the results depends therefore exclusively on the data itself and the applied preprocessing. The image index contains binary information defining whether an image has staining at a certain position. Hence, the per voxel classification is responsible for the result. It is therefore possible that the algorithm misses to retrieve datasets because the corresponding signal was not recognized by the preprocessing. Therefore special care must be taken by selection and tailoring of the preprocessing method.

The resolution of the index is another factor which can influence the accuracy of the result. The object index uses a lower resolution than the segmentation data. Therefore distances retrieved from the index have a lower accuracy. This influence also the quantification of overlap between objects.

A third factor influencing the result is the input data itself. After acquisition the image data undergoes a registration step, which establishes the fundamental precondition for spatial indexes, i.e., spatial correspondence over all datasets. However both the registration method and inter-subject variability introduce a level of spatial uncertainty into the data. The different sub structures of the neuron have different levels of variability. Cell bodies are much variable in position and shape than arborizations and arborizations are more variable than projections. With our data processing pipeline we observe an average error of $5\ \mu m$. This spatial error can have a major impact on certain quantifications. If structures are compared (e.g., object overlap

or staining similarity) whose diameter is in the range of the spatial error it is possible that the algorithm yields no usable result.

Utility of the Method for Circuit Neuroscience

The system is in active use for over a year. The feedback we get from the users underlines the utility of the provided tools for solving frequent queries to the data. For example common use cases involve the analysis of certain neurons which requires to find other datasets, both images and annotated objects, which depict the same structure. The different query types offer a variety of possibilities to extract a reasonable number of candidates from the database. The users state that this kind of research is impossible to perform without an automatic method which restricts the candidate list because of the size of the data collection.

Since the presented method focuses on achieving fast feedback times we were interested if the interactive employment of the tools leads to a gain of utility or insight into the data. The users report that the interactive execution of the querying tools influences the way they approach the problem. For example it is simply possible to probe the indexes at different regions and adjust while doing this the query parameters based on the insight gained from the intermediate results. This procedure enables interactive exploration of the image and object space in an intuitive way.

Limitations and Future Work

In the field of big data processing it is important to keep the scaling behavior of the algorithm and data structures in mind. In this area the presented method convinces with linear scaling behavior in all domains. However the data collections are constantly growing therefore it will be necessary, at some point in the future, to develop methods which allow to scale beyond the point when access and insertion becomes slow or the data storage too large for the current monolithic implementation. Possible solutions will most likely be found in methods which divide the index into smaller parts executable in parallel. The application of the *mapreduce* [24] approach provides a possible solution for this problem.

CHAPTER 6

Conclusion

Throughout this thesis I presented four research projects which approach the problem of interactivity in analysis and exploration of volumetric image data from different perspectives.

A highly interactive system with strict limitations concerning interaction is presented in chapter 2. The method allows the surgeon to explore the anatomy of the patient during the intervention by providing a virtual endoscopic view. The virtual view is synchronized to the real endoscope by an intra-operative navigation system. This allows to provide additional anatomical informations directly related to the current intervention state. The visualization system is able to provide synchronized real time feedback to the movement of the tracked surgical instruments and endoscope. Problematic in this scenario is the interaction of the user with the system, which is naturally limited in this environment. This limits the possibility of adjusting the visualization to the specific needs of the current stage of the intervention. In response to this, visualization methods have been developed which are robust and provide useful insight at all situations of the intervention. Furthermore in situations where interaction is not avoidable, the user interface is implemented in a way that it can be controlled by the already tracked instruments and a footswitch.

A method for feature detection using a machine learning approach is proposed in chapter 3. As part of a larger system for semi-automatic spine annotation, the feature detection module must meet strict runtime limitations. Furthermore the allowed memory usage is restricted as well. The proposed method solves this problems using an extended and optimized probabilistic boosting tree classifier algorithm. A multi-scale approach in combination with cascading and classifier sorting improves the runtime efficiency. Memory management using a cache hierarchy for derivative image data improves memory usage.

Chapter 4 presents a system for fast similarity based retrieval of neuronal substructures. The definition of similarity between these structures is difficult and due to varying expectations not universally solvable. To achieve fast retrieval feedback the system uses an indexing step performed offline. The method follows a bag of features approach for similarity calculation. The chapter explains how the characteristic properties of the different structures are expressed by sets of feature vectors.

A different approach for image and object retrieval was presented in chapter 5. Exploration of large image and object collections by local means such as local neighborhood, overlap or image expression is an extremely useful tool. However for exploration tasks the system has to perform within strict time limits. The presented method solves the problem through a large, extremely efficient and updatable index structure for spatial data. This indexes are applied to object data and image data. This allows to explore the object space by local neighborhood or overlap means as well as the image space by local expression or expression similarity.

6.1 Interactivity as Key Requirement

As the presented projects demonstrate, the creation of interactivity poses problems which have to be approached from many different sides. The most prominent problem to solve is to provide solutions with sufficient time performance enabling interactive interaction. During the work on these projects we learned that users are willing to sacrifice some of the quality of the result, if they gain fast feedback that allows to adjust the settings and simply re-do the processing. Therefore a big challenge is to balance the trade off between speed and quality. For example the classification algorithm used for feature detection in chapter 3 is severely restricted in its ability to produce a good classification rate by the proposed enhancements (classifier sorting and cascading), but the gain on execution speed compensates for this.

Another widely applicable method which helps to balance speed and quality is to use approximations. This is used in chapter 4 where shape and appearance of neuronal structures are approximated by a small set of feature vectors, and in chapter 5 where the staining expression is approximated by a binary mask. In both cases due to approximation the loss of accuracy has only a minor effect (if any) on the quality of the result but leads to a huge improvement of runtime performance.

The computation of this approximations as well as the building of index or look up structures where the data is stored in is typically performed in a preprocessing step ahead of time. Generally it is a promising approach to perform a part of the workload before the interactive analysis starts. However the costs which come with this approach should not be underestimated. In fact the software is no longer independent from some infrastructure. Preprocessing must be done as soon as new data is available in order to allow the user to start the analysis task as soon as possible. This requires for example server installations of the computation and also central storage for the preprocessing results.

Time performance is for sure the most important factor for interactivity. However it should not be ignored that other components are also important for interactive interaction. Chapter 2 covers a system where time performance never was a problem. However the special environment required to rethink the approach of user interaction as well as visualization. In this case the problem could be solved by a minimal elaborated interaction approach in combination with a visualization system that makes adjustments after initialization almost unnecessary.

To generalize this, computational methods that enable interactivity must consider the full interaction loop and need to find appropriate solutions for user interaction, computation and result presentation. User interaction must be effective and adequate for the specific task. The

computation must be fast enough to provide results with an acceptable quality. Results must be presented in a reasonable and coherent way.

6.2 Future Challenges

Volumetric images are now around for over three decades. Therefore a multitude of algorithms have been published to process and visualize this data. In the beginning volume data was almost only used for medical imaging but along with an increase of volumetric imaging methods also the fields of application widened.

Hanchuan Peng introduced in 2008 [101] *Bioimage informatics* as the new challenging field for engineering in biology with a variety of open tasks as segmentation, tracking, annotation, mining, atlas building and image data management. At the first look the tasks does not seem to be new, most of them have been studied for medical images for more then two decades. But the difference is that the imaging methods developed and refined for biological purposes are able to produce tenth of thousands of images for a single study in a relatively short time. Peng speaks of a "deluge of complicated biological and biomedical images" which "pose significant challenges for the image computing community" [101]. The projects presented in chapter 4 and 5 deal with exactly this kind of data and present methods which allow to interactively explore this collections.

A challenge in the future will be to find methods which allow to cope with this vast amount of data, to find ways which let the user interactively explore the image collections, and to find ways which allow analysis tasks on amounts of data which can not be handled in a serial way.

Not only in bioimaging the large data problem has to be approached. In medical imaging large amounts of data have been accumulated in the PACS archives of the hospitals. It is tempting to aggregate this data in order to derive new knowledge. However this image data has not been created to be spatially comparable. This poses problems which must be solved in order to make these collections accessible and explorable. Therefore problems such as image classification, object recognition, object localization and object association have to be solved for large collections of images with a large variability in image quality (e.g., the Khresmoi EU project [87] approaches this problem area).

In order to retain interactivity in exploration of these large data collections it will be inevitable to use indexing and pre-computation techniques to a large extent. Therefore unsupervised image data mining as preparation for interactive exploration and analysis will most likely be a interesting field for future research.

Bibliography

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[2] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, and J.M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

[3] S. Allaire, J.J. Kim, S.L. Breen, D.A. Jaffray, and V. Pekar. Full orientation invariance and improved feature selectivity of 3d sift with application to medical image analysis. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8. IEEE, 2008.

[4] J.D. Armstrong, K. Kaiser, A. Müller, K-F. Fischbach, N. Merchant, and N.J. Strausfeld. Flybrain, an on-line atlas and database for the drosophila nervous system. *Neuron*, 15(1):17–20, November 1995.

[5] A.H. Barr. Ray tracing deformed surfaces. *Computer Graphics*, 20(4):287–296, 1986.

[6] D. Bartz. Virtual endoscopy in research and clinical practice. *Computer Graphics Forum*, 24(1):111–126, 2005.

[7] S. Belongie and J. Malik. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[8] J. Beyer, M. Hadwiger, S. Wolfsberger, and K. Buhler. High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1696–1703, 2007.

[9] M. Blank and W.A. Kalender. Medical volume exploration: gaining insights virtually. *European Journal of Radiology*, 33(3):161–169, 2000.

[10] J.E. Blankenship and B. Houck. Nervous system (invertebrate), 2012.

[11] T. Boskamp, A. Homeyer, H. Laue, M. Schwier, F. Link, and H. Peitgen. Medical image analysis. *IEEE Pulse*, 2:60–70, 2011.

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

[13] A.H. Brand and E.L. Dormand. The gal4 system as a tool for unravelling the mysteries of the drosophila nervous system. *Current Opinion in Neurobiology*, 5(5):572–578, 1995.

[14] A.M. Bronstein, M.M. Bronstein, and L.J. Guibas. Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval. *ACM Transactions on Graphics*, 30:1:1–1:20, 2011.

[15] S. Bruckner, V. Soltészová, M.E. Gröller, J. Hladuvka, K. Bühler, J.Y. Yu, and B.J. Dickson. Braingazer–visual queries for neurobiology research. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1497–504, 2009.

[16] S. Cachero, A.D. Ostrovsky, J.Y. Yu, B.J. Dickson, and G.S.X.E. Jefferis. Sexual dimorphism in the fly brain. *Current Biology*, 20(18):1589–1601, September 2010.

[17] S.R. y Cajal. The structure and connexions of neurons. In *Nobel Lectures, Physiology or Medicine 1901-1921*. Elsevier Publishing Company, 1967.

[18] A. Cardona, S. Saalfeld, I. Arganda, W. Pereanu, J. Schindelin, and V. Hartenstein. Identifying neuronal lineages of drosophila by sequence analysis of axon tracts. *The Journal of Neuroscience*, 30(22):7538–7553, June 2010.

[19] A. Chiang, C. Lin, C. Chuang, H. Chang, C. Hsieh, C. Yeh, C. Shih, J. Wu, G. Wang, Y. Chen, C. Wu, G. Chen, Y. Ching, P. Lee, C. Lin, H. Lin, C. Wu, H. Hsu, Y. Huang, J. Chen, H. Chiang, C. Lu, R. Ni, C. Yeh, and J. Hwang. Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution. *Current Biology*, 21(1):1–11, December 2010.

[20] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[21] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[22] A.M. Cormack. Early two-dimensional reconstruction and recent topics stemming from it. In *Nobel Lectures, Physiology or Medicine 1971-1980*, volume 7. World Scientific Publishing Co., Singapore, 1992.

[23] E.R. Davies. A modified Hough scheme for general circle location. *Pattern Recognition Letters*, 7(1):37–43, 1988.

[24] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[25] C Demiralp. Similarity coloring of DTI fiber tracts. In *Proceedings of DMFC Workshop at MICCAI*, 2009.

[26] D. Dey, P.J. Slomka, D.G. Gobbi, and T.M. Peters. Mixed reality merging of endoscopic images and 3-D surfaces. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 1935 of *Lecture Notes in Computer Science*, pages 796–803, 2000.

[27] E. de Divitiis and P. Cappabianca, editors. *Endoscopic Endonasal Transsphenoidal Surgery*. Springer, 2003.

[28] H.W. Dong. *The Allen reference atlas: A digital color brain atlas of the C57Bl/6J male mouse.* John Wiley & Sons Inc, 2008.

[29] R. Donner, B. Micusik, G. Langs, and H. Bischof. Sparse MRF appearance models for fast anatomical structure localisation. In *British Machine Vision Conference*, volume 2, pages 1180–1189, 2007.

[30] R.R. Edelman, K.U. Wentz, H. Mattle, B. Zhao, C. Liu, D. Kim, and G. Laub. Projection arteriography and venography: initial clinical results with MR. *Radiology*, 172(2):351–357, 1989.

[31] R. Fagin. Combining fuzzy information from multiple systems (extended abstract). In *Principles of database Systems, Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on*, pages 216–226. ACM Press, 1996.

[32] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *IEEE Visualization*, pages 319–326. IEEE Computer Society Press, 1998.

[33] J. Fehr, A. Streicher, and H. Burkhardt. A bag of features approach for 3D shape retrieval. *Advances in Visual Computing*, 1(4):34–43, 2009.

[34] I. Fujishiro, T. Azuma, and Y. Takeshima. Automating transfer function design for comprehensible volume rendering based on 3d field topology analysis. In *IEEE Visualization*, pages 467–563. IEEE, 1999.

[35] C. Golgi. The neuron doctrine - theory and facts. In *Nobel Lectures, Physiology or Medicine 1901-1921*. Elsevier Publishing Company, 1967.

[36] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, volume 2, pages 1458–1465, 2005.

[37] S. Grimm, S. Bruckner, A. Kanitsar, and E. Gröller. Flexible direct multi-volume rendering in dynamic scenes. In *Proceedings of Vision, Modeling, and Visualization*, pages 505–512, 2004.

[38] D. Grosser, J. Diatta, and N. Conruyt. Improving dissimilarity functions with domain knowledge, applications with ikbs system. *Principles of Data Mining and Knowledge Discovery*, 1910(l):163–186, 2000.

[39] U. Güntzer, W.T. Balke, and Werner Kiessing. Optimizing multi-feature queries for image databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 419–428. Morgan Kaufmann Publishers Inc., 2000.

[40] S. Guthe, M. Wand, J. Gonser, and W. Straßer. Interactive rendering of large volume data sets. In *IEEE Visualization*, pages 53–60. IEEE, 2002.

[41] R.H. Güting. An introduction to spatial database systems. *The International Journal in Very Large Data Bases (VLDB)*, 3(4):357–399, 1994.

[42] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross. Real-time ray-casting and advanced shading of discrete isosurfaces. *Computer Graphics Forum*, 24(3):303–312, 2005.

[43] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15. Manchester, UK, 1988.

[44] E.H. Herskovits and R. Chen. Integrating data-mining support into a brain-image database using open-source components. *Advances in medical sciences*, 53(2):172–181, 2008.

[45] W. Hibbard and D. Santek. Interactivity is the key. In *Proceedings of the Chapel Hill workshop on Volume visualization*, pages 39–43. ACM, 1989.

[46] J. Hladuvka, A. König, and E. Gröller. Curvature-based transfer functions for direct volume rendering. In *Spring Conference on Computer Graphics*, volume 16, pages 58–65, 2000.

[47] G.N. Hounsfield. Radiography, 1978. US Patent 4,115,698.

[48] G.N. Hounsfield. Computed medical imaging. In *Nobel Lectures, Physiology or Medicine 1971-1980*, volume 7. World Scientific Publishing Co., Singapore, 1980.

[49] K. Ibukuro, C. Charnsangavej, M.H. Chasen, A.B. Cinqualbre, D.H. Herron, T.J. Robinson, and S. Wallace. Helical CT angiography with multiplanar reformation: techniques and clinical applications. *Radiographics*, 15(3):671–682, 1995.

[50] Ihab F. Ilyas, George Beskales, and Mohamed a. Soliman. A survey of top- k query processing techniques in relational database systems. *ACM Computing Surveys*, 40(4):1–58, October 2008.

[51] X. Jiang, H. Bunke, K. Abegglen, and A. Kandel. Curve morphing by weighted mean of strings. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 4, pages 192–195. IEEE, 2002.

[52] S.H. Joshi, J.D. Van Horn, and A.W. Toga. Interactive exploration of neuroanatomical meta-spaces. *Frontiers in Neuroinformatics*, 3(38), 2009.

[53] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, and M.E. Gröller. Cpr-curved planar reformation. In *IEEE Visualization*, pages 37–44, 2002.

[54] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Computer Vision, International Journal of*, 1(4):321–331, 1988.

[55] T. Kawamata, H. Iseki, T. Shibasaki, and T. Hori. Endoscopic augmented reality navigation system for endonasal transsphenoidal surgery to treat pituitary tumors: Technical note. *Neurosurgery*, 50(6):1393–1397, June 2002.

[56] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506–513. IEEE, 2004.

[57] A.C. Keene and S. Waddell. Drosophila olfactory memory: single genes to complex neural circuits. *Nature Reviews Neuroscience*, 8(5):341–354, 2007.

[58] J. Kettenbach, W. Birkfellner, and P. Rogalla. Virtual endoscopy of the paranasal sinuses. In E. Neri, D. Caramella, A. Aschoff, and C. Bartolozzi, editors, *Image Processing in Radiology: Current Applications*, pages 151–171. Springer, 2007.

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

[60] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *IEEE Visualization*, pages 255–262. IEEE Computer Society, 2001.

[61] W. Konen, S. Tombrock, and M. Scholz. Robust registration procedures for endoscopic imaging. *Medical Image Analysis*, 11(6):526 – 539, 2007.

[62] Stephen Koslow. Databasing the brain. *Nature*, 406(August):822–825, 2000.

[63] A. Krüger, Chr. Kubisch, B. Preim, and G. Strauss. Sinus endoscopy - Application of advanced GPU volume rendering for virtual endoscopy. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1491–1498, 2008.

[64] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *Proceedings of the annual conference on Computer graphics and interactive techniques*, pages 451–458. ACM, 1994.

[65] E. LaMar, B. Hamann, and K.I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *IEEE Visualization*, pages 355–361. IEEE Computer Society Press, 1999.

[66] M. Langer and K.-D. Kuhnert. A new hierarchical approach in robust real-time image feature detection and matching. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[67] R. Lapeer, M.S. Chen, G. Gonzalez, A. Linney, and G. Alusi. Image-enhanced surgical navigation for endoscopic sinus surgery: evaluating calibration, registration and tracking. *International Journal of Medical Robotics and Computer Assisted Surgery*, 4(1):32–45, March 2008.

[68] P. C. Lauterbur. All science is interdisciplinary - from magnetic moments to molecules to men. In *Les Prix Nobel. The Nobel Prizes 2003*. Nobel Foundation, 2004.

[69] Ping-chang Lee, Yu-tai Ching, H M Chang, and A. Chiang. A semi-automatic method for neuron centerline extraction in confocal microscopic image stack. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 959–962, 2008.

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

[71] X. Li and A. Godil. Exploring the bag-of-words method for 3d shape retrieval. In *IEEE Image Processing*, pages 437–440, 2009.

[72] C.Y. Y Lin, K.L. L Tsai, S.C. C Wang, C.H. H Hsieh, H.M. M Chang, and A.S. S Chiang. The neuron navigator: Exploring the information pathway through the neural maze. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 35–42. IEEE, IEEE, 2011.

[73] H. Ling. Diffusion distance for histogram comparison. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 246–253, 2006.

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

[75] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1150–1157, 1999.

[76] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[77] Christophe M., Marcin D.and Nicolas U., and M. Amini. High-level feature detection with forests of fuzzy decision trees combined with the rankboost algorithm. Technical report, Université Pierre et Marie Curie-Paris, 2007.

[78] P. Mansfield. Snap-shot MRI. In *Les Prix Nobel. The Nobel Prizes 2003*. Nobel Foundation, 2004.

[79] C.R. Maurer, R. Qi, and V. Raghavan. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003.

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

[81] M. Mayberg, E. Lapresto, and E. Cunningham. Image-guided endoscopy: description of technique and potential applications. *Neurosurg Focus*, 19(1:E10), 2005.

[82] A. Militzer and F. Vega-Higuera. Probabilistic boosting trees for automatic bone removal from ct angiography images. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7259, pages 725946–725953, 2009.

[83] M. Minsky. Microscopy apparatus, December 19 1961. US Patent 3,013,467.

[84] J.R. Mitchell, S.J. Karlik, D.H. Lee, and A. Fenster. Computer-assisted identification and quantification of multiple sclerosis lesions in mr imaging volumes in the brain. *Journal of Magnetic Resonance Imaging*, 4(2):197–208, 2005.

[85] B. Moberts, A. Vilanova, and J.J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *IEEE Visualization*, pages 65–72, 2005.

[86] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[87] Henning Müller, Allan Hanbury, and Alexandre Cotting. Khreshmoi. Internet, December 2012. `http://www.khresmoi.eu` [Online; accessed 17.12.2012].

[88] D.R. Nadeau. Volume scene graphs. In *Proceedings of the IEEE Symposium on Volume Visualization*, pages 49–56. ACM, 2000.

[89] NCHC and BRC/NTHU. Fly circuit. Internet, 2012. `http://www.flycircuit.tw/` [Online; accessed 15.11.2012].

[90] NEMA: Medical Imaging & Technology Alliance. The dicom standard. Internet. `http://medical.nema.org/standard.html` [Online; accessed 15.12.2012].

[91] A. Neubauer. *Virtual Endoscopy for Preoperative Planning and Training of Endonasal Transsphenoidal Pituitary Surgery*. PhD thesis, VRVis and Vienna University of Technology, 2005.

[92] A. Neubauer, S. Wolfsberger, M.-T. Forster, L. Mroz, R. Wegenkittl, and K. Bühler. Advanced virtual endoscopic pituitary surgery. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):497–507, 2005.

[93] M. Niemeijer, M.K. Garvin, K. Lee, B. van Ginneken, M.D. Abràmoff, and M. Sonka. Registration of 3d spectral OCT volumes using 3D SIFT feature point matching. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7259, pages 51–59, 2009.

[94] P. Nomikos, R. Fahlbusch, and M. Buchfelder. Recent developments in transsphenoidal surgery of pituitary tumors. *Hormones*, 3(2):85–91, 2004.

[95] C. North, B. Shneiderman, and C. Plaisant. User controlled overviews of an image library: A case study of the visible human. In *Proceedings of the first ACM international conference on Digital libraries*, pages 74–82. ACM, 1996.

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

[97] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Science*. Springer, 2002.

[98] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[99] D. Osumi-Sutherland, M. Longair, and J.D. Armstrong. Virtual fly brain: An ontology-linked schema of the drosophila brain. Poster at International Conference on Biomedical Ontology, November 2009.

[100] J. Pawley. *Handbook of biological confocal microscopy*. Springer, 2006.

[101] H. Peng. Bioimage informatics: a new area of engineering biology. *Bioinformatics (Oxford, England)*, 24(17):1827–36, September 2008.

[102] H. Peng, Z.ngcai Ruan, F. Long, J.H. Simpson, and E.W. Myers. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 28:348–353, 2010.

[103] K.P. Philip, EL Dove, DD McPherson, NL Gotteiner, MJ Vonesh, W. Stanford, JE Reed, JA Rumberger, and KB Chandran. Automatic detection of myocardial contours in cine-computed tomographic images. *IEEE Transactions on Medical Imaging*, 13(2):241–253, 1994.

[104] H. Ray, H. Pfister, D. Silver, and T.A. Cook. Ray casting architectures for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):210–223, 1999.

[105] C. Reinhart, C. Poliwoda, T. Günther, W. Roemer, S. Maass, and C. Gosch. Modern voxel based data and geometry analysis software tools for industrial CT. In *Proceedings of World Conference on Nondestructive Testing (WCNDT)*, 2004.

[106] J.C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71. IEEE, 2007.

[107] P. Rogalla. Virtual endoscopy of the nose and paranasal sinuses. In P. Rogalla, J. Terwisscha van Scheltinga, and B. Hamm, editors, *Virtual Endoscopy and Related 3D Techniques*, pages 17–37. Springer, 2001.

[108] P. Rogalla, A. Nischwitz, S. Gottschalk, A. Huitema, O. Kaschke, and B. Hamm. Virtual endoscopy of the nose and paranasal sinuses. *European Radiology*, 8(6):946–950, July 1998.

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

[110] W. C. Röntgen. The nobel prize in physics 1901. Nobelprize.org, November 2012. `http://www.nobelprize.org/nobel_prizes/physics/laureates/1901/` [Online accessed 17.12.2012].

[111] Z. Salah, D. Bartz, F. Dammann, E. Schwaderer, M. M. Maassen, and W. Strasser. A fast and accurate approach for the segmentation of the paranasal sinus. In *Bildverarbeitung für die Medizin*, Informatik Aktuell, pages 93–97. Springer, 2005.

[112] C.A. Schneider, W.S. Rasband, and K.W. Eliceiri. NIH image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7):671–675, 2012.

[113] M. Scholz, B. Fricke, St. Tombrock, M. Hardenack, K. Schmieder, M. von Düring, W. Konen, and A. Harders. Virtual image navigation: A new method to control intraoperative bleeding in neuroendoscopic surgery. *Journal of Neurosurgery*, 93:342–350, 2000.

[114] F. Schulze, K. Bühler, A. Neubauer, L. Holton, and S. Wolfsberger. Intraoperative Navigation virtueller Endoskopie bei endoskopischen transsphenoidalen Eingriffen an der Hypophyse. In *Proceedings of the Jahrestagung der Deutschen Gesellschaft für Computer- und Roboterassistierte Chirurgie (CURAC)*, pages 215–218, 2008.

[115] F. Schulze, K. Bühler, A. Neubauer, A. Kanitsar, L. Holton, and S. Wolfsberger. Intraoperative virtual endoscopy for image guided endonasal transsphenoidal pituitary surgery. *International Journal of Computer Assisted Radiology and Surgery*, 5(2):143–154, 2010.

[116] F. Schulze, M. Hansen, K. Bühler, and T. Liu B. Dickson. Dynamic spatial indexes for large registerred image and object collections. Technical Report TR-VRVis-2012-008, VRVis Research Center, Donau-City-Strasse 1, 1220 Vienna, Austria, December 2012.

[117] F. Schulze, D. Major, and K. Bühler. Fast and memory efficient feature detection using multiresolution probabilistic boosting trees. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, volume 19, pages 33–40, 2011.

[118] F. Schulze, M. Trapp, K. Bühler, T. Liu, and B. Dickson. Similarity based object retrieval of composite neuronal structures. In *Proceedings of Eurographics Workshop on 3D Object Retrieval (EG3DOR)*, pages 1–8, 2012.

[119] R. Scorcioni, S. Polavaram, and G. Ascoli. L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature protocols*, 3(5):866–76, January 2008.

[120] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the International Conference on Multimedia*, pages 357–360. ACM, 2007.

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

[122] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003.

[123] W. Skalli, F. Lavaste, and J.L. Descrimes. Quantification of three-dimensional vertebral rotations in scoliosis: what are the true values? *Spine*, 20(5):546–553, 1995.

[124] D. Speray and S. Kennon. Volume probes: Interactive data exploration on arbitrary grids. *Computer Graphics*, 24(5):5–12, 1990.

[125] P.K. Spiegel. The first clinical x-ray made in america–100 years. *American Journal of Roentgenology*, 164(1):241–243, 1995.

[126] D. Stalling, M. Westerhoff, and H.C. Hege. *Amira: A highly interactive system for visual data analysis*, volume 38, chapter 38, pages 749–767. Elsevier, 2005.

[127] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, 39(3):441–471, December 2007.

[128] K. Tingelhoff, A. Moral, M. Kunkel, M. Rilk, I. Wagner, K. Eichhorn, F. Wahl, and F. Bootz. Comparison between manual and semi-automatic segmentation of nasal cavity and paranasal sinuses from CT images. In *Engineering in Medicine and Biology Society (EMBS)*, pages 4683–4689, 2007.

[129] A.W. Toga. Neuroimage databases: the good, the bad and the ugly. *Nature Reviews Neuroscience*, 3(4):302–309, 2002.

[130] Z. Tu. Probabilistic boosting-tree: learning discriminative models for classification, recognition, and clustering. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1589 –1596, 2005.

[131] S. Tweedie, M. Ashburner, K. Falls, P. Leyland, P. McQuilton, S. Marygold, G. Millburn, D. Osumi-Sutherland, A. Schroeder, R. Seal, and H. Zhang. FlyBase: enhancing drosophila gene ontology annotations. *Nucleic acids research*, 37(Database issue):555–559, January 2009.

[132] D.C. C Van Essen. Windows on the brain: the emerging role of atlases and databases in neuroscience. *Current Opinion in Neurobiology*, 12(5):574–579, 2002.

[133] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[134] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–511 – I–518, 2001.

[135] M.Y. Wang, C.R. Maurer Jr, J.M. Fitzpatrick, and R.J. Maciunas. An automatic technique for finding and localizing externally attached markers in CT and MR volume images of the head. *IEEE Transactions on Biomedical Engineering*, 43(6):627–637, 1996.

[136] M.Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM, 2000.

[137] F.W. Wehrli, B.R. Gomberg, P.K. Saha, H.K. Song, S.N. Hwang, and P.J. Snyder. Digital topological analysis of in vivo magnetic resonance microimages of trabecular bone reveals structural implications of osteoporosis. *Journal of Bone and Mineral Research*, 16(8):1520–1531, 2001.

[138] L. A. Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. PhD thesis, University of North Carolina at Chapel Hill, 1991.

[139] S. Wolfsberger, A. Neubauer, K. Bühler, R. Wegenkittl, Th. Czech, S. Gentzsch, H. Böcher-Schwarz, and E. Knosp. Advanced virtual endoscopy for endoscopic transsphenoidal pituitary surgery. *Neurosurgery*, 59(5):1001–1010, 2006.

[140] R. Yagel, D. Stredney, G. J. Wiet, P. Schmalbrock, L. Rosenberg, D. J. Sessanna, and Y. Kurzion. Building a virtual environment for endoscopic sinus surgery simulation. *Computers and Graphics*, 20(6):813–823, December 1996.

[141] J. Yamashita, Y. Yamauchi, M. Mochimaru, Y. Fukui, and K. Yokoyama. Real-time 3-D model-based navigation system for endoscopic paranasal sinus surgery. *Biomedical Engineering, IEEE Transactions on*, 46(1):107–116, 1999.

[142] J.Y. Yu, M.I. Kanai, E. Demir, G.S.X.E. Jefferis, and B.J. Dickson. Cellular organization of the neural circuit that drives drosophila courtship behavior. *Current Biology*, 20:1–13, September 2010.

[143] S. Zambal, A. Schöllhuber, K. Bühler, and J. Hladuvka. Fast and robust localization of the heart in cardiac MRI series. In *Proceedings of International Conference on Computer Vision Theory and Applications*, volume 2, 2008.

[144] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Fast automatic heart chamber segmentation from 3d CT data using marginal space learning and steerable features. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8. IEEE, 2007.