FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Non-linear Mapping of Drosophila Populations based on Neuronal Structure

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Florian Ganglberger, BSc

Matrikelnummer 0828078

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Mitwirkung: Ass.Prof. Dipl.-Ing. Dr. Georg Langs

Wien, 5. September 2012 _____ _____

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Non-linear Mapping of Drosophila Populations based on Neuronal Structure

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Medical Informatics

by

Florian Ganglberger, BSc

Registration Number 0828078

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Assistance: Ass.Prof. Dipl.-Ing. Dr. Georg Langs

Vienna, 5. September 2012 _____ _____
(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Florian Ganglberger, BSc
Zeitling 31, 4320 Perg

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 every who was involved in this thesis. In particular, I want to thanks Georg Langs for his expert advices, his time and his patience. I thank Laszlo Tiran, who is referred as expert in this thesis annotated the data and helped me a lot with the biological background. The data was provided by the Barry Dickson Group of the Research Institute of Molecular Pathology.

In addition, i want to thank Katja Bühler (VRVIS) and Florian Schulze (VRVIS) for initializing this project and for their availability whenever I had questions. Also thanks to Professor Robert Sablatnig for supervising this thesis.

A heartfelt thanks to my family who supported me while my entire study.

Abstract

The behavior of drosophila can be linked to the activity of a set of neurons. These neurons express different genes whereby their manipulation and effects on neurons are subjects of current research. The automatic identification of similar neurons in different brains give biologists an opportunity to analyze a set of brain volumes without manual annotation. In this thesis, a similarity measure was developed to compare small, segmented neuronal structures. This measure can be used to build mappings of fly brains, based on the similarity in sub-regions. The mapping is performed by a non-linear dimension reduction method called Diffusionmaps. Fly brain images have varying quality regarding noise and location stability, so this method is well suited for use due to its robustness. The resulting mapping can be applied to provide biologists with an overview of the data and visualize differences and similarities between mutations. Additionally, a method for identifying similar regions of brains with different mutations is introduced. For this, a multi-modal genetic algorithm is applied to find brain areas which maximize the similarity measure between flies. In conclusion, the methods perform non-linear mapping of fly brain regions, based on similarity measures, which compare local appearance in confocal microscopy images. The maps reflect groups of flies that exhibit similar structure in local brain areas. On the other hand, also regions in fly brains can be detected that have structures regarding to specific genetic variations. In addition, the similarity measure can be used for image retrieval.

Kurzfassung

Das Verhalten von Fruchtfliegen (Drosophila) und dessen Abhängigkeit von Genen ist Gegenstand aktueller Forschung. Änderungen im Erbgut führen zu unterschiedlichen neuronalen Strukturen, die wiederum das Verhalten der Fliegen determinieren. Werden nun ähnliche Neuronen von verschiedenen Genalterationen gebildet, können Rückschlüsse zwischen Verhalten und Erbgut gezogen werden. Mithilfe automatischer Identifikation von ähnlichen Neuronen in unterschiedlichen Gehirnen können Biologen eine große Anzahl von Gehirnen analysieren, ohne diese einzeln betrachten zu müssen. Im Rahmen der Diplomarbeit wurde ein Ähnlichkeitsmaß entwickelt, welches es erlaubt segmentierte, neuronale Strukturen von 3D-Konfokalmikroskop Aufnahmen zu vergleichen. Damit ist es möglich, in Fliegenpopulationen nach ähnlichen Neuronen zu suchen, und somit Abfragen nach bestimmt Neuronen zu starten. Des Weiteren kann das Ähnlichkeitsmaß dazu verwendet werden, um mittels Diffusion Maps (eine Methode zur nicht-linearen Dimensionreduktion) Mappings von Fliegengehirnen zu erstellen, die die Relation der Fliegengehirne für Subregionen wiedergeben. Eine weitere Anwendung ist die Identifikation von ähnlichen Gehirnregionen von Fliegen mit unterschiedlichen Genalterationen. Dafür wird ein multi-modaler genetischer Algorithmus verwendet, der Regionen findet, die die Ähnlichkeit zwischen Fliegen maximieren. Als Resultat dieser Diplomarbeit wurden Methoden entwickelt, die nicht-lineare Mappings dazu verwenden, lokale Bereiche von Fliegengehirnen zu vergleichen. Die generierten Mappings stellen Gruppen von Fliegen dar, die die selbe Struktur in den gleichen Gehirnregion aufweisen. Andererseits ist es auch möglich Regionen von Fliegengehirnen zu bestimmen, die eine ähnliche Struktur für unterschiedliche genetische Variationen aufweisen. Schlussendlich ermöglichen die vorgestellten Methoden nach Genabhängigen Strukturen in Fliegenpopulationen zu suchen, welches eine schnelle Alternative zur manuellen Datenanalyse darstellt.

Contents

CHAPTER 1

Introduction

This chapter provides an overview of this thesis. In Section 1.1, the biological background is explained. Based on this, Section 1.2 describes the motivation of this thesis and the primary problem. The research aims are described in Section 1.3, and Section 1.4 lists a short outline of the thesis.

1.1 Biological Background

Aspects of the behavior of *Drosophila* can be linked to the activity of neurons [65]. For example, the male courtship behavior is based on 1500 neurons which express the fruitless gene [66] [37] [76]. The manipulation of this gene and the effect on the neurons is the subject of current research [19]. The search for relevant neurons and their organization gives biologists an opportunity to understand the neuronal basis of behavior [76]. To visualize neurons, *transgenic* systems (GAL4-UAS) [9] and genetically encoded fluorescent proteins (e.g. the membrane bound mCD8-GFP [48]) are used. They highlight neuronal structure in laser scanning confocal microscopes images.

For generation of the Vienna Tiles (VT) library [34] [47] the *Drosophila* genome was virtually cut into 2000 base pair long pieces that overlap by approximately 300 bases. Pieces (tiles) which were likely to be active in the nervous system were used to generate *transgenic* fly lines. Each of these *transgenic* fly lines contains one unique tile which can be considered as a genetic alteration. In the context of this thesis, such alterations are referred to as *mutation*. The usage of *VT-lines* allows to assign observed changes in the neuronal structure specifically to one particular mutation.

Laser scanning confocal microscope images are 3D volumes that capture the morphology of drosophila brains. An example of a fly brain is shown in Figure 1.1. The volumes show brain

tissue and specific traced neurons [9]. A neuron consists of a *cell body* (white round spots in Figure 1.1), *dendrites* (branching structures in Figure 1.1) and *axons* (tubular structures in Figure 1.1). *Dendrites*, also referred to as *arborisation* are afferent fibers, while the axon or *projection* is efferent relative to the cell body. The cerebral cortex is an area, which is surrounded by a connective tissue sheet (noisy area, brighter than the background in Figure 1.1). It includes the *cell bodies* of neurons, while the majority of *projections* and *arborisations* lies outside.

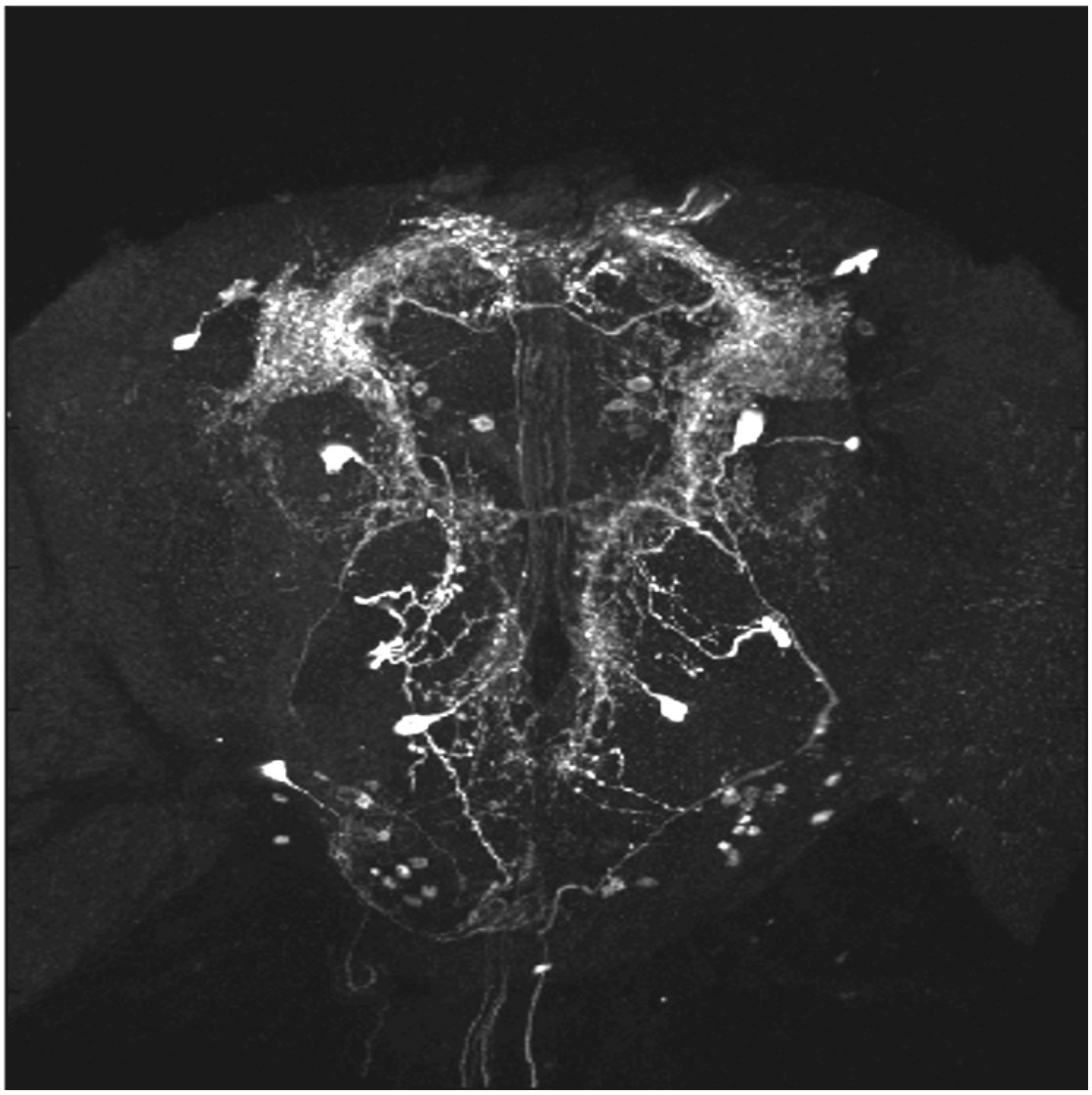


Figure 1.1: Example drosophila brain of **[D1]** dataset (Section 6.1) with a labeled set of neurons. The brain is visualized by a Maximum Intensity Projection (*MIP*) computed over the z-axis

1.2 Motivation

Automatic identification of corresponding neurons across a population with known genetic characteristics gives biologists an opportunity to analyze substantial numbers of cases in an exploratory fashion [36]. By computing a similarity measure of local parts such as *projections* or *arborisations*, computer-aided population analysis of neuronal structures regarding to their genes is possible [46]. Content based image retrieval enhances the systematically identification of sets of neurons that contribute to drosophila behavior compared to manual analysis.

Although the volumes are registered based on the *neuropils* (the major part of gray matter of the brain), which are a stable morphological reference [56], the neurons still exhibit inter-individual variability in location and shape. This is caused by genes (different genes can lead to different morphology [65]) and the limited accuracy of a non-rigid registration based on (compared to single neurons) large structures (neurophils) [56]. The resulting lack of overlap renders local similarity measurements challenging. Another factor contributing to similarity inaccuration is the noise level in some volumes which depends on the quality of experiments [76]. Noise can affect the precision of the search (proof in Section 6.2). Therefore, this influence should be minimized. Similarity measures have to take possible misalignment and variability into account, while at the same time capturing similarities of narrow structures such as projections.

The measure can be used to build embeddings of fly brains, based on the similarity in subregions. The created embeddings can also be used as a lookup table for a search in fly brains, with regard to a specified region. Because they are based on similarities, a search of neurons can be performed on the map instead of the data. This is a faster alternative compared to an adhoc computation of a similarity-based ranking. The results can be applied to provide biologists with an overview of the data, visualize differences and similarities between mutations, and analyze the variability of the brains.

1.3 Aim of this Thesis

The aim of this thesis is to develop a method to analyze and visualize the variability of fly brains and their relationship to different alterations of genes. This task can be split into four different goals which represent the main contribution of this thesis:

- A similarity measure to compare small, segmented neuronal structures like *axons* and *dendrites*. It shall take possible misalignments and variability into account.
- Image retrieval of neuronal structures. By using them as query patterns, brains which also contains these structures shall be top-ranked in the retrieval results retrieval results
- A mapping which represents the similarity of local brain areas in a set of flys with different mutations. Brains, which show the same neuronal structures will be mapped close together, while brains with different structures will exhibit a higher distance. The principle

idea behind this method is, that clusters of different mutations indicate that they express the same neuronal structures.

- A visualization of similar regions for brains with different mutations. Brains, which exhibit a low variability of structures for specific mutations, lead to a high similarity measure to each other. As a consequence, similar regions have a high mean inter-individual similarity (compared to dissimilar regions). Its visualization reveals corresponding neuronal structures of different mutations.

1.4 Synopsis

This thesis consists of 5 main parts: A state of the art analysis of the central methods used in this thesis, a description of the structure based similarity measure, the mapping of neuronal structure, the visualization of brain similarities and a validation of the results of this novel method.

Chapter 1 gives an introduction about the biological background, the motivation, as well as the aims of this thesis. It offers basic knowledge about neuronal structures, their dependency on genetic mutations and the need of their computer-aided analysis. In the last part of the chapter, the aims of this thesis are formulated.

In Chapter 2, state of the art methods related to the main contribution of this thesis are explained. Every section consists of description and the discussion of the field. Section 2.1 outlines approaches for comparing neurons in different ways, such as model- or branching-based methods as well as the entire field of computational drosophila brain analysis. Section 2.2 is related to non-linear mapping method which forms the core of the embedding method discussed in this thesis. Different methods for linear and non-linear methods are described. Section 2.3 describes the gradient vector flow which is used in Chapter 3. Measures for cluster stability to investigate the structure of the data are shown in Section 2.4. The last section of this chapter explains methods for multi-modal optimization for genetic algorithms which are used in Chapter 5 for similarity visualization.

Similarity of neuronal structures is introduced in Chapter 3. It explains the pre-processing of the volume data before it comes to the principle steps of the method. At first, local neuron orientation is estimated via structure tensors in Section 3.1 and expressed as a vector field. In Section 3.2, a query pattern is selected by automatic segmentation, which corresponds to the neuronal structure that will be compared. By dissolving the vector field into the surrounding area, location shifts and different shapes are taken into consideration. This process is described in Section 3.3. The last step is the calculation of the similarity by computing the Sum-of-Squares of Differences between the vector field of the query pattern and the other fly brains in Section 3.4. The process of sorting fly brains by their similarity to a query pattern represents the image retrieval.

Non-linear mapping of neuronal structure for population analysis is explained in Chapter 4. Non-linear dimension reduction on fly brain volume data is described in Section 4.1. This is

the basis of the similarity visualization in Chapter 5. Embedding-based clustering is used for investigation of the structure of the data. This is discussed in Section 4.2. Methods for image retrieval on embedded data are introduced in Section 4.3.

Similarity visualization is introduced in Chapter 5 which is a new method for identification and visualization of brain regions that are similar within subgroups of a study population. Section 5.1 and 5.2 describe a process multi-modal optimization via genetic algorithms to maximize a similarity criterion. The visualization of this criterion and its discussion can be found in Section 5.3.

The validation, experiments and results are reported in Chapter 6. The similarity measure is validated by using it as a classifier between similar fly brains and not-similar fly brains. The validation of the mapping is done by quantitative measures while the similarity visualization compares the results to expert-annotated brains. The results of the experiments and their discussion are also part of this chapter.

The Chapter 7 contains the conclusion and outlook. The Appendix consists of the parameter optimization (Appendix A) for the results (Chapter 6), additional results in Appendix B, and a glossary (Appendix C).

At the publication date of this thesis, it is planed to submit the method for computing the similarity measure (Chapter 3) to the Journal *Neuroinformatics*, by the name *Structure-based neuron retrieval across drosophila brains*. Further publications of the mapping is also in prospect.

CHAPTER 2

State of the Art

This chapter outlines the state of the art of methods either related to the thesis or used as part of the algorithm described in the next sections. Section 2.1 gives an overview of *Drosophila* brain related analysis and retrieval methods as well as similarity or distance metrics between neurons. Linear and non-linear dimensionality reduction methods are compared in Section 2.2. *Gradient Vector Flow*, which is used in Chapter 3, is explained in Section 2.3. Cluster Stability methods, such as those used for the validation of the non-linear mapping are outlined in Section 2.4. State-of-the-art evolutionary multi-modal optimization techniques, such as the ones used in Chapter 5 are subject of Section 2.5.

2.1 Related Work in Computational Drosophila Brain Analysis

This section is an overview of related work regarding to computational analysis of *Drosophila* brains. It includes approaches for detection of neuronal structures, their segmentation and registration, as well as their storage and retrieval.

According to Masse et al. [46], the mapping (labeling of neurons by tracers for visualization and analysis) of neuronal circuits in *Drosophila* brain is divided into two major parts: The labeling of many neural structures in a single brain (not topic of this thesis) by using electron microscopy [10] and sparse labeling of brains by light microscopy. The advantage of sparse labeling is the (relative to an entire brain) easier full automated segmentation. The labeled neurons can be investigated individually and piece-wise stitched together to a plan of neuronal circuits (like Yu et al.l [76]). The following methods are focused on sparse labeling and are closely related to this thesis, since they also focus on specific structures of interest.

Model based representation of neurons: The representation of a neuron can be model- or feature-based. Model-based approaches use image segmentation to build tree like skeletons of *axons* and *dendrites* [61]. Creating models of artificial neurons can be done by using parameters, such as the distribution of branching points, measured by "real" neurons and simulateing neurons by stochastic processes [58]. Neuronal growth can be simulated via diffusion limited aggregation [74] [44]. A method for modeling "real" neurons are compartment models which consist of snakes with cylinders that represent the tubular structure of neurons [61]. In such simulations branching points and their connectivity need to be selected by the user to generate an initial coarse representation of the neuron [61]. An automated method is the use of principal curves [31] as skeletons on tubular objects [4]. This approach traces the principle curve of the data through neuronal trees and is able to detect branches non supervised [4].

For model based representations, similarity metrics like *TED* or DIADEM exist. The Tree Edit Distance (*TED*) is the amount of nodes which need to be added/removed from one tree to create a second one [78]. TED as well as standard morphometrics like diameter/length/surface have the disadvantage, that they are not spatial specific [26]. This can be avoided by the DIADEM metric [26]. It is able to compare morphological changes or errors by scoring corresponding nodes and branches by their connectivity to local regions.

Feature based representation of neurons: A feature based neuronal representation has the advantage, that a segmentation of the neuron and a bifurcation detection (for the branching) is not necessary because local image features such as structure tensors [59] are used. An example for this is multi-scale vessel detection [25] was developed for tubular structures. A similar approach is used in Masse et al. [46]. This paper is strongly related to this work.

As for this thesis, structure tensors [25] are used after a smoothing of the image data to detect tubular structures on *GAL4-UAS* [9] labeled neurons in *Drosophila* brains. The similarity between *axons* of different brains is calculated by first reducing the dimensionality from three dimensions (because of the 3D tangential vector) to one dimension by the dimensionality reduction algorithm developed by Chigirev and Bialek [12], and finally computing the mutual information between two brains. The purpose of this measure is to identify *neuroblast*[1] clones which are neurons that originate from neuronal stem cells with the same genome. Therefore a classifier is trained that needs a small set of manual annotated data. The work in this thesis is related to those approaches, but the focus is an retrieval of similar structures and the exploratory analysis of large population. Another difference is, that no vector field spreading (Section 3.3) is used to consider small differences in the shape and location shifts in Masse et al. [46].

In the context of structure tensors, tensor-based similarity measures are also relevant. A method for comparing neurons derived by Diffusion Tensor Imaging (DTI) was introduced in Durrleman [22]. It is based on vector fields generated by diffusion tensors. The vector fields of near neuronal bundles of two different individuals were smoothed by a Gaussian kernel to generate an overlap. The similarity metric is simplified an integral of inner products of the vector fields. In Verma

[1] cell which will develop into a neuron

et al [71], tensors are used as feature vectors and mapped by Isomaps [67] to determine low dimensional subspaces. This sets corresponding tensors in relation, which would also depend on the overlap and is therefore not for shifted neurons or neurons with different shapes.

Alignment of brains: For the development and application of those methods, aligned sets of *Drosophila* brains are needed. A tool for aligning *Drosophila* brains to atlases is BrainAligner [51]. It is designed to align *GAL4* labeled *Drosophila* brains to an atlas by identifying corresponding landmarks on a reference labeling. For this, a *nc82 antibody*[2] [73] was used to label the entire *neuropils*. Two brains are then registered by a non-regid local 3D alignment by matching corresponding 3D feature points via RANSAC [23]. A similar approach is used in BrainGazer [11]. The registration method is in principle the same as for Brain Aligner. Additionally it supports rendering of the brains and retrieval by visual queries (selection of the queries on the image data). This is closely related to this thesis, because all image data and segmentations are registered and generated by BrainGazer.

2.2 Dimensionality Reduction

The methods which are proposed in this thesis make intensive use of dimensionality reduction methods for the mapping of the fly brain data (Chapter 4 and 5). Due to the exploratory nature of this work, linear and non linear methods will be compared (Chapter 6). Basic linear methods are Principle Component Analysis (*PCA*) [1] and Multidimensional Scaling (*MDS*) [68]. While *PCA* preserves the variance of the data, *MDS* is based on the inter point distances. It can be proven, that these two methods (Classic metric *MDS* and *PCA*) minimize the same criterion and therefore lead to the same result [41]. Because the relations between fly brain areas are defined by distance matrices (Section 4.1), *MDS* is used.

In contrast to linear methods which are based on euclidean distances, non linear methods use different metrics like geodesic (Isomaps [67]) or kernel-based (Kernel *PCA* [60]) to reveal the underlying structure of the data. If this structure is non-linearly embedded, *PCA* and *MDS* would fail [70]. There are several methods of non linear dimensionality reduction methods which differ in the distance metric used and their parameters [70]. Fly brain images have varying quality regarding to noise and contrast, which depend on the quality of the experiments [11]. Therefore, Diffusion maps are well suited for use due to its robustness [13] compared to other methods [70].

Multidimensional Scaling

The name spans a variety of related methods. Classic metric *MDS* introduced by [68] is based on euclidean distances between data points, while non metric *MDS* [38] can use rank information of the data. A generalization of the classic metric, Sammon's mapping [57] allows a non-linear

[2] a large Y-shaped protein

mapping. Because *MDS* is used in this thesis as a linear comparison to Diffusion maps, classic metric *MDS* is described in this section. To increase readability, from now on classic metric *MDS* is referred to as *MDS*. In the following, the classic metric is explained.

The aim of *MDS* is to scale data $\mathbf{X} = \{x_1, x_2, ..., x_N\} \in \mathbb{R}^h$ with distances $d(x_i, x_j)$ from a high dimensional space $\mathbb{R}^h$ while a low dimensional space $\mathbb{R}^l$ (with $l \ll h$) by preserving the original distances. The coordinates of the points in both spaces are unknown. Actually, *MDS* does not preserve distances but pairwise scalar products. Therefore, the distance matrix needs to be converted to a matrix of pairwise scalar products, called the Gram matrix **S**, by the following equation [41]:

$$s(x_i, x_j) = -\frac{1}{2}(d^2(x_i, x_j) - \langle x_i \cdot x_i \rangle - \langle x_j \cdot x_i \rangle) \tag{2.1}$$

Because the data points are unknown, the Gram matrix can also obtained by double centering of the $N \times N$ distance matrix **D** [41]:

$$S = -\frac{1}{2}(\mathbf{D} - \frac{1}{N}\mathbf{D}1_N 1_N^T - \frac{1}{N}1_N 1_N^T\mathbf{D} + \frac{1}{N^2}1_N 1_N^T\mathbf{D}1_N 1_N^T) \tag{2.2}$$

The coordinates of the data in an l dimensional space can then calculated by an eigenvalue decomposition of **S** [41]:

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\boldsymbol{T}} \tag{2.3}$$

$$\mathbf{S} = (\mathbf{V}\mathbf{\Lambda}^{\mathbf{1/2}})(\mathbf{\Lambda}^{\mathbf{(1/2)}}\mathbf{V}^{\boldsymbol{T}}) \tag{2.4}$$

$$\mathbf{S} = (\mathbf{\Lambda}^{\mathbf{1/2}}\mathbf{V}^{\boldsymbol{T}})^{\boldsymbol{T}}(\mathbf{\Lambda}^{\mathbf{1/2}}\mathbf{V}^{\boldsymbol{T}}) \tag{2.5}$$

$$\hat{\mathbf{Y}} = \mathbf{I}_{\boldsymbol{l}\times\boldsymbol{N}}\mathbf{\Lambda}^{\mathbf{1/2}}\mathbf{V}^{\boldsymbol{T}} \tag{2.6}$$

$\mathbf{V} \in \mathbb{R}^{N\times N}$ holdes eigenvectors and $\Lambda \in \mathbb{R}^{N\times N}$ a diagonal matrix with the eigenvalues of **S**. The resulting coordinates $\mathbf{Y} = \{y_1, y_2, ..., y_N\} \in \mathbb{R}^l$ are a e low dimensional representation of **X**.

Diffusion Maps

Diffusion maps are a non linear embedding method which can find a lower dimensional representation of the data. The dimensionality reduction is based on local distances that are used in a diffusion process to describe the global structure of the data [13]. In principle, the diffusion process is a Markow random walk on a graph given by the distance matrix **D**. The random walk is processed for several time steps **t**, therefore the diffusion metric is time depended. As a result of the random walk, it is more likely to go to nearby data points while the probability to walk to a point far away tends to zero [13]. This connectivity is modeled by a Gaussian kernel function [13]

$$k(x_i, x_j) = e^{-\frac{(D_{x_i,x_j})^2}{\sigma}} \tag{2.7}$$

which is applied on the distance matrix **D**. The kernel scale σ controls the size of the neighborhood where the random walk is applied. The diffusion kernel needs to satisfy following constraints:

$$k(x_i, x_j) = k(x_j, x_i) \tag{2.8}$$

$$k(x_i, x_j) \geq 0 \tag{2.9}$$

For this reason, the distance matrix has to be symmetric (or an other kernel function must be used). The entries in the diffusion kernel can be interpreted as the probabilities to walk from one point to another point. This includes, that the sum of the probabilities to walk from one point to all other points is 1 [18]. This can be achieved by normalizing the rows of the $\mathbf{K}$ by its sums.

$$m_{i,j}^{(1)} = \frac{k(x_i, x_j)}{\sum_k k(x_i, x_k)} \tag{2.10}$$

This can be also formulated as

$$\mathbf{M}^{(1)} = \mathbf{Z}^{-1}\mathbf{K} \tag{2.11}$$

where $\mathbf{Z}$ is a diagonal matrix of the row sums. The result is the diffusion matrix $\mathbf{M}^{(1)}$ which defines the probability $p^{(1)}(x_i, x_j)$ of a walk from one point x_i to another point x_j in a single timestep. The probability of a walk after t timesteps is then

$$p^{(t)}(x_j, t|x_i = \mathbf{M}^t \tag{2.12}$$

which can also be written as $\mathbf{M}^{(t)}$. Similar to *MDS*, a low-dimensional mapping $\mathbf{Y}$ of the data can be calculated by using eigenvector decomposition of $\mathbf{M}^{(t)}$ [13].

$$\mathbf{M^{(t)}} = \mathbf{U\Lambda U^T} \tag{2.13}$$

which can be solved in similar to Equation 2.6

$$\hat{\mathbf{Y}} = \mathbf{I}_{l\times N}\mathbf{\Lambda}^{\mathbf{1/2}}\mathbf{V}^{\mathbf{T}} \tag{2.14}$$

$\hat{\mathbf{Y}} = \{y_1, y_2, ..., y_N\}$ represents the data points embedded in a l dimensional diffusion space.

2.3 Gradient Vector Flow

Originally, Gradient Vector Flow (*GVF*) is an extension for active contours [35]. Active contours are defined as curves on an image which can be moved by an internal force field induced by the curve itself and an external force field caused by the image data [35]. In this context, *GVF* can be seen as an external force field which pushes the curve onto the contours of objects even when the initialized position is bad [75]. Therefore *GVF* is used for automatic initialization of active contours for image segmentation [24] or motion tracking [55]. Another possilbe application is the generation of curve skeletons from 3D shapes [30].

GVF is a technique that diffuses the original contour [75]. Only a few iterations lead to a vector field that is close to the original with less spatial variance. This reduces noise while the initial vector field is mostly preserved [75]. A high amount of iterations spreads the contour information into the surrounding area. For this reason, *GVF* is used in this thesis to smooth

neuronal structures and expand them for a better comparison to other neurons (explained in Section 3.3).

GVF minimizes an energy function of a gradient field $\mathbf{V} \in \mathbb{R}^2$ of the image $\mathbf{x} \in \mathbb{R}^2$ by

$$\mathcal{E} = \int\int \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\mathbf{V}|^2 |\mathbf{V}_{GVF} - \mathbf{V}|^2 dxdy \tag{2.15}$$

to generate a vector flow field $\mathbf{V}_{GVF}(x, y) = [u(x, y), v(x, y)]$. In homogeneous areas (and therefore weak gradients), $|\mathbf{V}|$ is small. This leads to a domination of the sum of squares term. Therefore $\mathbf{V}_{GVF}$ will be a slowly varying field. In areas with strong gradients (areas with contours), $|\mathbf{V}_{GVF} - \mathbf{V}|^2$ will dominate, which results in a field that is nearly equal to $\mathbf{V}$. μ is a parameter that regulates the trade-off between those two terms.

Without proof [75], $\mathcal{E}$ can be minimized by an iterative numerical approximation

$$u_t(x, y, t) = \mu\nabla^2 u(x, y, t) - (u(x, y, t) - \mathbf{V}_x(x, y, z))(\mathbf{V}_x(x, y)^2 + \mathbf{V}_y(x, y)^2) \tag{2.16}$$

$$v_t(x, y, t) = \mu\nabla^2 v(x, y, t) - (v(x, y, t) - \mathbf{V}_x(x, y, z))(\mathbf{V}_x(x, y)^2 + \mathbf{V}_y(x, y)^2) \tag{2.17}$$

which converges if

$$\Delta t \leq \frac{\Delta x \Delta y}{4\mu} \tag{2.18}$$

whereas **t** is the iteration step. It converges faster on coarse images with large Δx and Δy, while large μ results in a slower convergence but smoother field [75].

To use the *GVF* on higher dimensional data (like on 3D vector fields which are used in this thesis), Equation 2.15 is defined for n dimensional data as

$$\mathcal{E} = \int_{\mathbb{R}^n} \mu|\nabla\nu|^2 + |\nabla f|^2 |\nu - \nabla f|^2 dx \tag{2.19}$$

where ∇ needs to be applied individual for every component of ν. Also the iterative Equations 2.16 and 2.17 can be expressed for higher dimensions by

$$\nu_t = \mu\nabla^2 \mathbf{V}_{GVF} - (\mathbf{V}_{GVF} - \mathbf{V})|\mathbf{V}|^2 \tag{2.20}$$

As a result, *GVF* can now applied on 3D vector fields which is shown in Section 3.3.

2.4 Cluster Stability

Cluster stability is a measure for validating the performance of clustering. There are three different types of clustering: Partitioning clustering assigns every object to a specific cluster. Hierarchical clustering divides the data stepwise into disjunct classes (divisive) or joins every object

successive into bigger groups (agglomerative). The third type assigns each object a membership probability. This is called fuzzy clustering. Different cluster stability measures focus on different clustering types [50]. This thesis is focused on disjoint groups of fly brains. Hence, this section outlines stability measures for partitioning clustering. Therefore it is necessary to identify the "best" amount of clusters k in terms of compactness or stability measures.

Compactness: An example for compactness is the DUNN index [21], which selects the amount of clusters that minimizes the ratio between the minimum distance within a cluster and the maximum distance between clusters. Another method is the David-Bouldin index [16] where the within cluster scatter in proportion to the cluster separation is measured.

Stability of k-means: Other methods investigates the data by their behavior regarding to the variability of the clustering [50]. They choose the amount of clusters which shows the lowest variation of subsets and therefore stable for input randomization. They reject cluster solutions were the data is randomly split for too large ks or randomly merged for too small ks. Ben-David [54] comes to the conclusion, that cluster stability measures depend on the function that is minimized, like the quadratic intra-cluster error for k-means, and is therefor influenced by the cluster algorithm. For this reason, and for simplification, the following measures are discussed in relation to the algorithm which is used in this thesis (k-means), although other clustering methods would be possible.

It has been shown by Rakhlin and Caponetto [54], that the stability of k-means is defined by the geometry and underlying distribution of every class. They proved, that the clustering is stable for unique minimization. In this context, a minimization is defined as the sum of the inter-class scatter of all cluster:

$$W(C) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - c_i\|^2 \tag{2.21}$$

where $\mathbf{C}$ are disjunct subsets of the data $\mathbf{X}$ with the centers $\bar{c}_i$. This infers, that the clustering is stable if the cluster centers c are too. For this reason, the stability depends on k with respect to the underlying distribution of the data if the amount of data points is large enough to cover the structure (if present) of the data for stable estimation of the centers.

Statistical tests for determining cluster stability: A method proposed by Bertrand and Mufti [7] uses statistical tests to determine cluster stability. Therefore the data set is sampled into subsets. For every subset, several parameters for the clustering are computed, like the cluster isolation [43] and the cluster validity [43]. The cluster stability is defined as the difference (in terms of a statistical test) of the distribution of the computed parameters and a random clustering [7].

Cluster Stability Index: Ben Hur et al. [6] proposed a method which defines the stability of a clustering by its ability to handle perturbed data. The data is sampled multiple times to generate in each case two sub samples with a sampling ration f ($f \geq 0.5$), then the paired-wise similarity between the cluster labels are computed. The distribution of the similarities gives information about the inherent structure of the data (depending on the cluster algorithm [5]).

The problem of the method proposed by Ben Hur et al. [6] is that the stability is not normalized by a random clustering and therefore not comparable between different sets. Although this is done by Bertrand and Mufti [7], it was not tested in the scope of this thesis. Instead, an approached by Lange et al. [40] was used due to its simplicity.

Lange et al. [40] introduced the *Cluster Stability Index* (*CSI*), which is used in this thesis. For this approach, k-means (or other clustering algorithms) is used as a classifier trained by one half of data, to predict the clustering on the other half of the data. In detail, the data $\mathbf{X}$ is divided randomly into two halves $\mathbf{X}^1$ and $\mathbf{X}^2$, then the clustering $kmeans()$ is applied on $\mathbf{X}^1$. The result of $kmeans(X)$ is then used to train the classifier $\phi()$ to predict the clustering on $\mathbf{X}^2$. $\phi(X^2)$ assigns the data points of $\mathbf{X}^2$ to the cluster centers of $kmeans(X)^1$. The Hamming distance [28] between the cluster labels computed on $\mathbf{X}^2$ and the labels $\phi(\mathbf{X}^1)$ defines the stability value ς_k. Because there is no unique representation of the cluster labels (cluster labels can be randomly permuted), the [39] is used to reassign the labels to maximize the agreement of $\mathbf{X}^2$ and $\phi(\mathbf{X}^1)$ and therefore minimize the Hamming distance to get the stability value $\varsigma_k(kmeans(X))$. This process is repeated multiple (**s**) times to compute the empirical average of the similarity measures $\hat{\varsigma}_k(kmeans(X))$. Because the stability value depends on k (the number of possible mismatches increases [40]), $\hat{\varsigma}_k(kmeans(X))$ is normalized by computing the empirical stability value for random labeling $\hat{\varsigma}_k(random(X))$

$$\bar{\varsigma}_k(kmeans(X)) = \frac{\hat{\varsigma}_k(kmeans(X))}{\hat{\varsigma}_k(random(X))} \tag{2.22}$$

which defines the *CSI* $\bar{\varsigma}_k$. The best amount of clusters can be selected by choosing the lowest stability index. The procedure explained is shown in Algorithm 2.1 [40].

Data: Data X, sampling amount **s**, and the amount of clusters k

Result: Cluster Stability Index $\bar{\varsigma}_k$

for s *times* **do**
    Split the data set into to equal halves X^1 and X^2
    Use $kmeans(X^1)$ to train the classifier ϕ
    Calculate the distance between $\phi(X^2)$ and $kmeans(X^2)$
end
Sample s random k-labelings to calculate their average stability value $\hat{\varsigma}_k(random(X))$
Normalize $\hat{\varsigma}_k(kmeans(X))$ by $\hat{\varsigma}_k(random(X))$ to get $\bar{\varsigma}_k()$

Algorithm 2.1: Algorithm for calculating the cluster stability

2.5 Multi-modal Optimization using Genetic Algorithms

Genetic algorithms, introduced by Holland [32] address the problem of optimization by simulating natural evolution. The basic idea is to express parameters of the function which shall be optimized as bit-strings similar (in fact, the genome is encoded in 4 different bases) to the chromosomes of living beings. One single bit-string represents one solution candidate, or in the context of evolution, individual. A fitness function is maximized by the individual that represents the parameter settings of the global optimum.

This can be found by generating a population P of the size $\#P$ and let the individuals "compete" against each other in terms of selecting the best f individuals. This principle is called " survival of the fittest" or *selection*. The next step of evolution is reproduction by *crossover* and mutation. *Crossover* selects pairs of "parents" whose bit-strings are recombined. The new children are inserted into the population until the population size $\#P$ is reached. By repeating this multiple times, individuals with high fitness can pair, while individuals with low fitness are removed from the population. In addition, *mutation* is used, to increase the diversity of the population. With certain probability, every bit of an individual is changed to the opposite. This keeps the individuals different, which can avoid local minima. The iterative process *selection* - *crossover* - *mutation* - *repeat* enables a convergence of the population to a a global optimum (but it can still fall in local optima [53]).

Multi-modal[3] optimization is a domain of techniques to optimize a function with multiple optimum solutions. In this case, the task is not only to find one global optimum, but many or local optima [64]. Although in the field of multi-modal optimization are more techniques like differential evolution [77], particle swarm optimization [3] or evolution strategy [33], the scope of this section is on genetic algorithms because the problem which is given in this thesis can expressed as bitstrings 5.2.

The first approach for finding multi-modal distinct solutions with genetic algorithms was introduced by Goldberg and Richardson [27]. A *niching*[4] method was proposed to reduce the genetic drift[5] of the selection. A slower drift allows more diversity in the population and therefore a parallel convergence of different solutions to an optimum. This can be achieved by *sharing*, where every individual in the population "shares" its resources with individuals that are similar (for example in terms of the Hamming distance of the bit-string). Therefore, large niches have reduced fitness which allows a separate development of characteristics without domination.

Another *niching* approach is crowding [17] used by Davidor [15] for multi-modal optimization. In the normal *selection* process, the $\#P - f$ individuals are replaced by new children. This can remove potential useful genotypes from the population [69]. Crowding alters this in a way, that a new child replaces the individual with the most similar bit-string (if the fitness is better). A parent has usually at least 50% of the genome of the child (depended on the kind of *crossover*) and is therefore most likely to be removed from the population.

Spatial Selection [14] adds a spatial component to the selection of parents. The population is placed on an n-dimensional grid to mimic isolation by distance. The population is separated on this map, and only individuals close together can produce offspring. So it is possible that niches develop in different regions of the map. This is done by choosing a random location on the grid. From this point, two r-step random walks are performed over the grid. The fittest individuals on this walk are chosen to be the parents. The produced child is then inserted on the original location if it is fitter than it. According to Turner [69], spatial selection can be used to enhance crowding and sharing. He came to the conclusion, that spatial sharing performs very badly compared to spatial crowding.

[3] Multiple aims

[4] Niche = similar individuals which lead to one optimum

[5] Convergence of the population to one specific solution, which would not be multi-modal

Li et al. [42] proposed a species conserving genetic algorithm which divides the population into species according to their similarity. Only individuals within a specie can be produce children. After every reproduction cycle, the species of the population a determined by their similarity to the species seed. The species seed is fittest individual of every specie from the previous cycle. Other approaches which are not investigated in this section (because they are simple not in the scope of this thesis) like clearing [52], restricted tournament selection [29] are compared by Singh et al. [64].

For this thesis, spatial selection [14] in combination with crowding [17] was used due to its simplicity and efficiency for higher population sizes [69] and because the spatial grid can be exploited for the initialization of the problem in Section 5.2. The exact application of these methods is described in Section 5.2.

2.6 Summary

This chapter outlined state of the art methods related to the main contribution of this thesis. At first, related work in computational *drosophila* brain analysis was described. Therefore, model based representation of neurons was compared to feature based representation. Feature based approaches provide the advantage, that segmentation or bifurcation detection is not necessary because local image features are used. Also the alignment of brains was explained because it is the pre-processing step for the brain images which are used in this thesis.

Dimensionality reduction is a main topic of this thesis, so linear and non-linear methods are outlined. The focus of this chapter was on Multidimensional scaling and Diffusion maps because they are used in the methods.

For vector field spreading, Gradient Vector Flow was described as state-of-the-art method. As an extension for active contours, a energy function is minimized to provide a smooth spread field in contrast to the initial vector field.

Another topic of this thesis is cluster stability. Measures for cluster validation were outlined, such as compactness and stability. Cluster stability is used in the methods for investigating the structure of the data, therefore the focus was on cluster stability measures. The two groups of measures were described, statistical tests for determining cluster stability and cluster stability index.

The last section of this chapter was about multi-modal optimization using genetic algorithms. The field of genetic algorithms was outlined with focus on *niching* for multi-modal optimization. Several methods such as spatial selection and crowding were explained and compared, whereby spatial selection and crowding was chosen for this thesis due to its simplicity and efficiency for higher population sizes.

CHAPTER 3

Structure-based Similarity Measure of Neurons

A set of 3D images $\mathbf{I}_1, \ldots, \mathbf{I}_N \in \mathbb{R}^{m \times n \times h}$, each showing a *drosophila* brain, is given. They are registered based on their neuropil structure [11]. In a query case $\mathbf{I}_Q$ a user marks a *query region* in the form of a binary mask $\mathbf{R} \in \{0,1\}^{m \times n \times h}$. Let $\mathbf{I}_Q^R = \langle {\mathbf{I}^R}_Q, \mathbf{R} \rangle$ denote the query, i.e., the query volume together with the query region definition. The objective is to define a distance function $d(\mathbf{I}_Q^R, \mathbf{I}_i^R)$ between the query and all volumes in the index. In the following it is described how to derive this distance function in a way that allows for spatial variability of the neural structures. The distance is basis for the ranking that constitutes the retrieval result (Section 3.4) and the distance matrix for the creation of the embeddings in Chapter 4 and 5.

To compare structures like *projections* and *arborizations*, a vector based approach is used as an alternative to voxel based methods (Section 6.2). Vectors along the neurons provide the advantage, that they model the structure and are simple to propagate to surrounding area. To compute the vectors, an eigenvalue decomposition of structure tensors is used [25] (Figure 3.1,(1)), while a Gradient Vector Flow (*GVF*) [75] spreads them into surrounding area (Figure 3.1,(3)). This can compensate variability of shape and location, because the spreading generates a bigger overlap even if the query-neuron is sligthly displaced . The Sum-Of-Squared-Differences of the resulting vector fields is used as a similarity measure (Figure 3.1,(4)) which allows to generate a similarity ranking concerning to the query pattern. This represents the image retrieval.

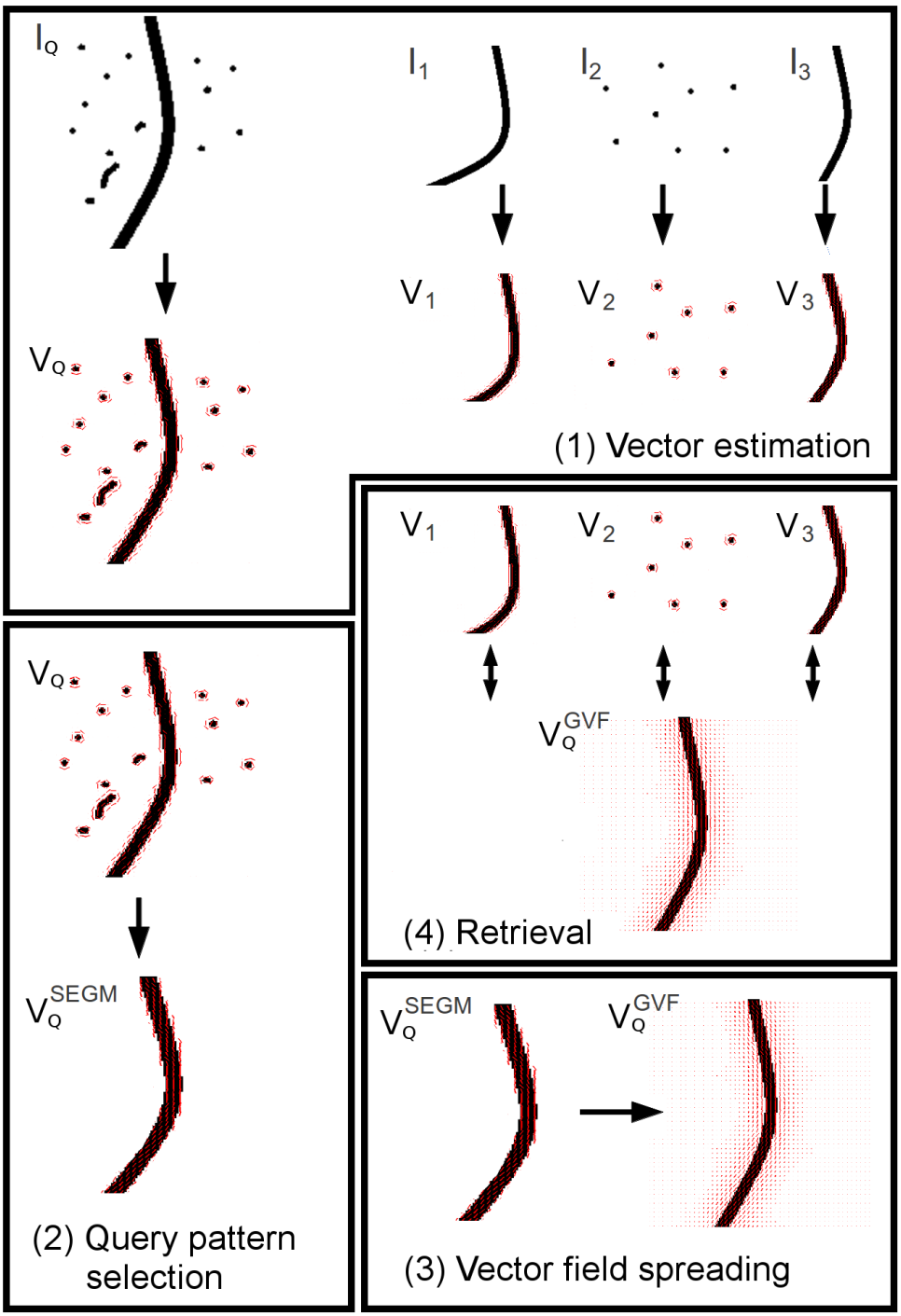
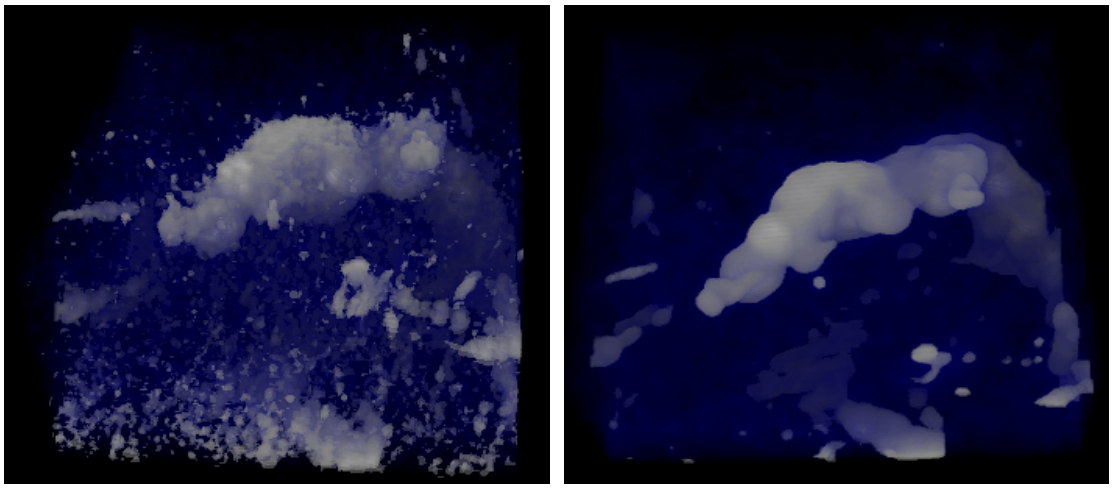


Figure 3.1: Principle steps of the retrieval process: (1) Estimate vectors along structures, (2) Choosing the query pattern (using of pre-segmented areas or an automated segmentation of the largest topology), (3) Propagation of the vector field into the surrounding area, (4) Compute similarity

3.1 Vector estimation

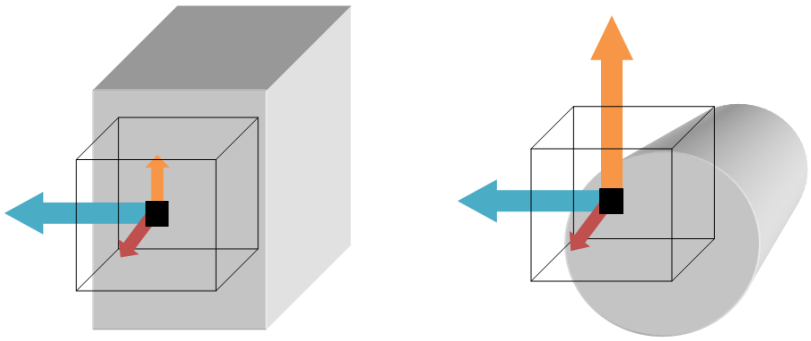
Due to noisy data and registration errors, *projections* and *arborizations* show slightly different shape on the surface of their tubular structure when comparing images of different *drosophila*. These different shapes can influence the vectors along the structure in a later step. To provide stable vectors, the data is smoothed by a 3D gaussian filter. Another effect is, that small individual differences between fibers disappear which enhances the comparability. In addition fine fibers of *arborizations* merge together to build a larger topology. As it will be shown in Chapter 6, this leads to a better comparison. The result of the smoothing are N filtered images $\mathbf{I}_i^F \in R^{m \times n \times h}$. An example for smoothing a neuron is shown in Figure 3.2. Noise and small structures are reduced in Figure 3.2 (b) compared to Figure 3.2 (b).



(a) (b)

Figure 3.2: A random neuron of the testset **[Testset1]** (introduced in Section 6.1 with (a) and without (b) smoothing. The neuron is visualized by the Matlab visualisation tool-kit [8]

Vectors that represeent the orientation of a tubular structure, are calculated by eigenvalue decomposition of structure tensors. A structure tensor is a second-moment matrix which describes points by gradients in their neighborhood. Compared to a gradient estimation method like the first order derivative, which computes vectors orthogonal to a structure (because of the highest change of intensity), eigenvalue decomposition of structure tensors generates also vectors along a structure. Sato [59] describes a structure tensor as a Hessian matrix based on local partial second order derivatives of the original volume.



(a) (b)

Figure 3.3: The eigenvectors of a structure tensor near a flat, surface-like neighbourhood (a)/on a round/tubular structure (b).

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix} \tag{3.1}$$

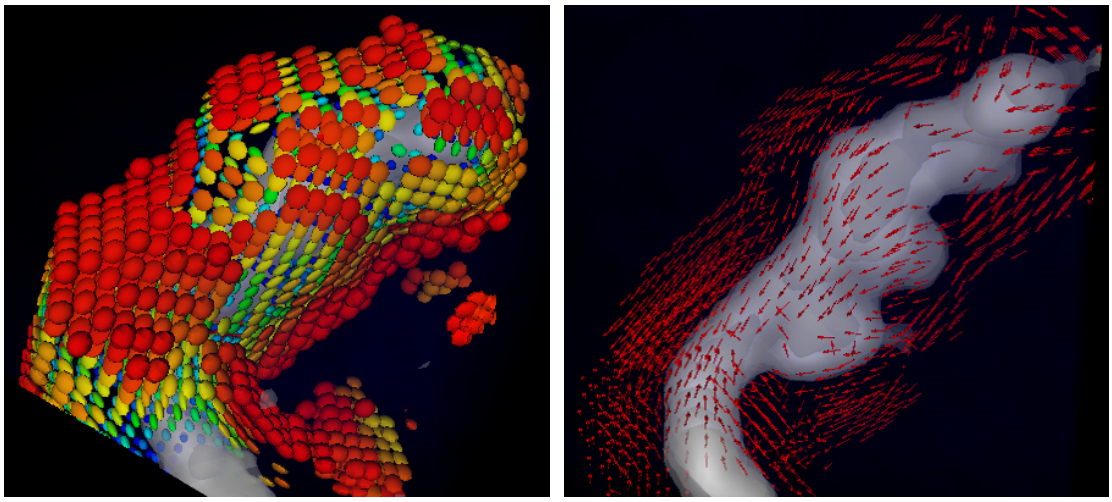
Additional information from the surrounding area can be used to enhance the detection of a tubular structure. In [25], the second order derivative from a Gaussian kernel measures the contrast between regions inside and outside the range $[-\sigma, \sigma]$. This can be achieved by calculating the partial derivatives $\mathbf{I}_x$, $\mathbf{I}_y$ and $\mathbf{I}_z$ for the neighbourhood of every voxel and convolute these volumes with a Gaussian kernel. So, the structure tensor $\mathbf{S}_{x,y,z}$ for every voxel in the filtered image $\mathbf{I}_i^F$ is

$$\mathbf{S}_{x,y,z} = \begin{pmatrix} \mathbf{I}_x^2 & \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_x\mathbf{I}_z \\ \mathbf{I}_x\mathbf{I}_y & \mathbf{I}_y^2 & \mathbf{I}_y\mathbf{I}_z \\ \mathbf{I}_x\mathbf{I}_z & \mathbf{I}_y\mathbf{I}_z & \mathbf{I}_z^2 \end{pmatrix} \tag{3.2}$$

The eigenvalues of the structure tensor, λ_1, λ_2 and λ_3, indicate topological structures in the 3D image $\mathbf{I}_i^F$. If

$$\lambda_1 >> \lambda_2 \approx \lambda_3 \tag{3.3}$$

is true for a voxel $\mathbf{I}_{x,y,z}$, a flat, surface-like neighborhood is found. The largest eigenvector $\mathbf{v}_1$ stands normal on the surface, while $\mathbf{v}_2$ and $\mathbf{v}_3$ point along the surface. Figure 3.3 (a) shows the direction of the vectors on a flat surface.



(a) (b)

Figure 3.4: Normalised structure tensors of a neuron represented by elispoids (a) and the largest eigenvector (red arrows) after normalisation (b).

A tubular or round structure, is found when

$$\lambda_1 \approx \lambda_2 >> \lambda_3 \tag{3.4}$$

This means that two orthogonal vectors $\mathbf{v}_2$ and $\mathbf{v}_3$ are normal to the surface and $\mathbf{v}_1$ shows along the structure. The orthogonal vectors on a tubular structure are shown in Figure 3.3 (b). The eigenvector $\mathbf{v}_3$ with the smallest eigenvalue ($\mathbf{v}_3$) is the vector with the smallest gradient which is orthogonal to $\mathbf{v}_1$ and $\mathbf{v}_2$. To use the structure tensors as description of a tubular structure, a normalization, described in [20], is applied.

$$\lambda_i' = e^{(-\gamma|\lambda|)} + \epsilon \quad \text{for i=1,2,3} \tag{3.5}$$

$$\lambda_1' \leq \lambda_2' \leq \lambda_3', \quad \lambda_i* = \frac{\lambda_i'}{(\lambda_2'\lambda_3')^2} \quad \text{for i=1,2,3} \tag{3.6}$$

In principle, the tensors are transformed from a gradient orientation to a structure orientation. The structure tensors of a neuron and their orientation (largest eigenvector) are visualized in Figure 3.4 (for the visualization of the volume data, the Matlab visualization tool-kit was used [8]). The ratio $\frac{\lambda_3}{\lambda_2}$ indicates how tubular and delimited to the surrounding area a structure is. A topology value t, defined as $1 - \frac{\lambda_3}{\lambda_2}$ for every voxel, can be used to differentiate *axon* and *arborizations* from noise and background.

For further computation (Section 3.2), only v_3 and the topology value is needed to describe topologies. Therefore, v_3 is stored in a vector matrix $\mathbf{V}_i \in \mathbb{R}^{m \times n \times h \times 3}$ and $\mathbf{t}$ in the topology matrix $\mathbf{T}_i \in \mathbb{R}^{m \times n \times h}$. The vector estimation is performed for every image $\mathbf{I}_i$.

3.2 Selection of the query pattern

Structures like other *projections/arborizations* near the query pattern, can have negative effects on the vector spreading, so they have to be removed from the vector matrix $\mathbf{V}$. This can be done by using either manual selection of the query pattern or an automatic segmentation within a bounding box.

In a previous step, the topology matrix $\mathbf{T}$ was introduced, which contains a topology value between 0 and 1 for every voxel. *projections* are usually large connected structures in the topology matrix $\mathbf{T}$, also *arborizations* are merged to one as a result of the smoothing in Section 3.1. The window segmentation can be performed on the binary thresholded topology matrix $\mathbf{T}_r$ which can be achieved by thresholding the topology values in $\mathbf{T}$. To find the optimal threshold, Otsu's method [49] is used because it is a fast standard method. It minimises the weighted sum of intra-class variances

$$\sigma_w^2(r) = \omega_1(r)\sigma_1^2(r) + \omega_2(r)\sigma_2^2(r) \tag{3.7}$$

where ω_i are the probabilities of the separated classes by a threshold r, and σ their variances.

Within a bound box, the largest topology can be determined by using a standard connected component labelling algorithm [62] for binary images. The algorithm is simple straight-forward implementation: Search for the next unlabelled pixel and use a flood-fill algorithm to label all connected pixels. This is done as long as there are unlabelled pixels. Figure 3.5 shows, that the segmentation of an axon reduces the amount of structures which would be compared because only the big axon in the middle is selected.

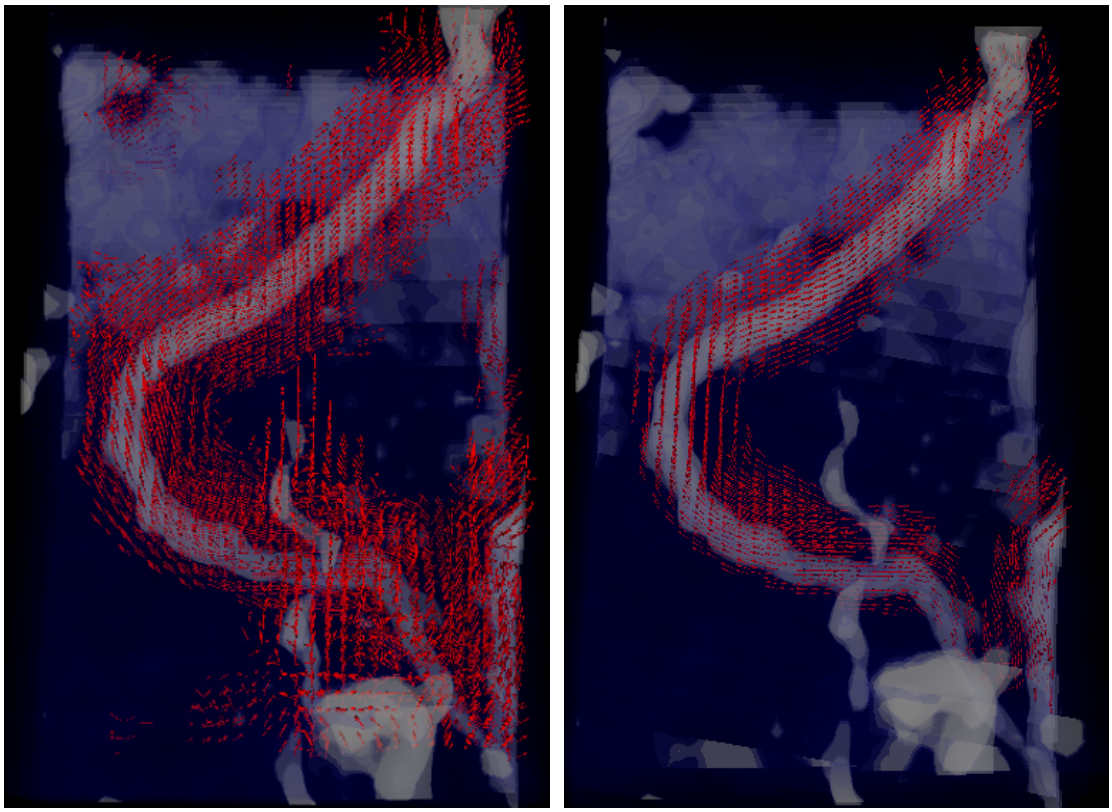
For either manual selection or automatic segmentation, the query region is denoted as $\mathbf{R} \in \mathbb{R}^{m \times n \times h}$ which is a binary mask that is 1 for voxels that are part of the query region and 0 if they are not. $\mathbf{R} = \mathbf{T}_r$ for automatic segmentation.

3.3 Vector field spreading

To spread the vector into the surrounding area, Gradient Vector Flow ([75] and [30]) is used. By applying this on the query pattern $\mathbf{V}_Q$, the structures are expanded and diffused, so it is more likely to match them in related volumes. Another property of *GVF* is the smoothness of the produced gradient vector flow field $V_{GVF} = [u(x,y,z), v(x,y,z), w(x,y,z)]$. This field is defined by the minimization of the energy function

$$\mathcal{E} = \int\int\int \mu(u_x^2 + u_y^2 + u_z^2 + v_x^2 + v_y^2 + v_z^2 + w_x^2 + w_y^2 + w_z^2) + |\mathbf{V}_Q|^2 |\mathbf{V}_{QGVF} - \mathbf{V}_Q|^2 dxdydz \tag{3.8}$$

If $|\mathbf{V}_Q|$ is small (no/less structure), the first term dominates, which leads to a slowly varying field. If $|\mathbf{V}_Q|$ is large, the term $|\mathbf{V}_Q^{\mathbf{GVF}} - \mathbf{V}_Q|$ is minimized, resulting in vectors near $\mathbf{V}_Q$, but



(a) (b)

Figure 3.5: Axon without segmentation (a) and with largest topology segmentation (b). The red arrows represents the vectors along the structure.

forced to be smoothed in homogeneous regions. The trade-off between these two terms can be controlled by the parameter μ.

In [30], it can be shown that Equation 3.8 can be numerical approximated in an iterative way by

$$u_t(x,y,z,t) = \mu\nabla^2 u(x,y,z,t) - (u(x,y,z) - V_{Qx}(x,y,z))|V_Q(x,y,z)|^2 \tag{3.9}$$

$$v_t(x,y,z,t) = \mu\nabla^2 v(x,y,z,t) - (v(x,y,z) - V_{Qy}(x,y,z))|V_Q(x,y,z)|^2 \tag{3.10}$$

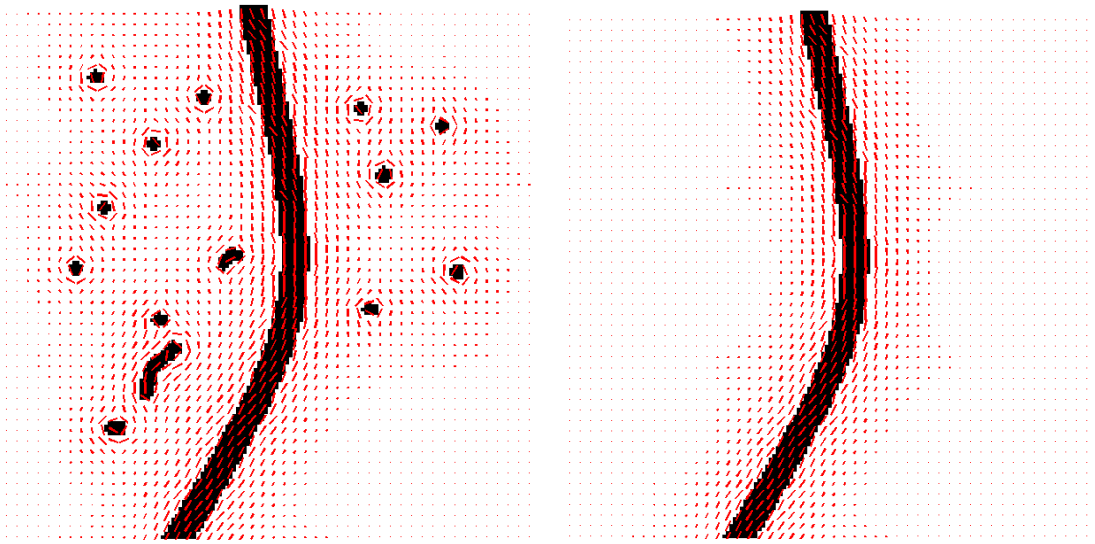
$$w_t(x,y,z,t) = \mu\nabla^2 w(x,y,z,t) - (w(x,y,z) - V_{Qz}(x,y,z))|V_Q(x,y,z)|^2 \tag{3.11}$$

which converges if

$$\Delta t \leq \frac{\Delta x \Delta y \Delta z}{6\mu} \tag{3.12}$$

whereas t is the iteration step. This iterative solution converges faster on large vectors, while large μ results in a slower convergence but smoother field.

The gradient vector flow field depends on the quality of the data. Noisy data and additional structures in the query area can deform the field in a negative way. Therefore Figure 3.6 shows synthetic 2D data. The red lines represent the vector along the structures, spread by *GVF*. As one can see, the vector field of the largest topology (b) is smoother than the vector field from the whole image (a).



(a) (b)

Figure 3.6: GVF (a) of synthetic, 2D data, compared to GVF of the largest topology (b). Red lines represents the vectors along structures, spread by GVF

3.4 Similarity measure and retrieval

The measure of similarity is based on the vector fields in the query area **R**. Due to the computational intensity the *GVF* is only used on the query pattern $\mathbf{V}_Q^R$. Therefore, the *GVF* is applied on the unit vectors of the structures $\mathbf{V}_Q^R$, so the structure itself has a vector norm of 1, while the norm is decreasing with the distance. This presents the probability of finding the structure at a location. The resulting vector field $\mathbf{V}_Q^{R^{GVF}}$ can be compared within the query area **R** in other volumes $\mathbf{V}_{1..N}$ by the Sum-of-Squares of Differences between two vector fields [22].

$$distance(\mathbf{V}_Q^{R^{GVF}}, \mathbf{V}_i^R)^2 = \sum_{x \in \mathbf{R}} ||\mathbf{V}_{Q_x}^{R^{GVF}} \mathbf{V}_{Q_x}^{R^{GVF}}|| + \sum_{x \in \mathbf{R}} ||\mathbf{V}_{i_x}^R \mathbf{V}_{i_x}^R|| - 2 \sum_{x \in \mathbf{R}} ||\mathbf{V}_{Q_x}^{R^{GVF}} \mathbf{V}_{i_x}^R|| \tag{3.13}$$

Then, for two images $\mathbf{I}_i$ and $\mathbf{I}_j$, the distance function $d(\mathbf{I}_i, \mathbf{I}_j)$ can be defined as

$$d(\mathbf{I}_i^R, \mathbf{I}_j^R) = \begin{cases} distance(\mathbf{V}_i^{R^{GVF}}, \mathbf{V}_j^R) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3.14}$$

where $\mathbf{I}_i^R$ is the query image with the corresponding (spreaded) vector field $\mathbf{V}_Q^{R^{GVF}}$ and $\mathbf{V}_j^R$ is the vector field of $\mathbf{I}_j^R$. The distance is than calculated for the region **R**.

The result is a distance function $d(\mathbf{I}_i^R, \mathbf{I}_j^R)$ that can be used for image retrieval within a query region. This is done by ording the the vector fields $\mathbf{V}_i (i = 1..N)$ of the images $\mathbf{I}_i (i = 1..N)$ ascending by their distance to the vector field $\mathbf{V}_Q^{GVF}$ of the query case $\mathbf{I}_Q^R$.

3.5 Summary

This section introduced a similarity measure for *drosophila* brain images **I** in the query area **R**. Spatial variability and variability in the shape of the neural structures are taken into consideration by the usage of *GVF*. The method defines a distance function $d(\mathbf{I}_i^R, \mathbf{I}_j^R)$ which can be used for image retrieval. In addition, the distances can be used for the creation of distance matrices for the generation of embeddings. This is described in Chapter 4.

CHAPTER 4

Non-linear Mapping of Neuronal Structure

This chapter will describe a method for non-linear mapping of *drosophila* populations for specific brain regions. A binary matrix $\mathbf{R} \in \mathbb{R}^{m \times n \times h}$ defines a brain region of interest. The similarity measure $d(\mathbf{I}_i^R, \mathbf{I}_j^R)$ of Chapter 3 is applied for the region R to define the relations between *drosophilas*. Then, dimensionality reduction is used to generate a low dimensional mapping of the population to reveal the inherent structure of the data. The concept of this process is shown in Figure 4.1. A *drosophila* population consists of two different groups (red and green) for a specific region (rectangle) which is not known. The plot shows the structure of the data by mapping similar brains close together.

Diffusion maps are used to generate embeddings of the data. They are based on a diffusion matrix which is created by using the similarity measure of Chapter 3. The diffusion process and the mapping of the data is described in Section 4.1. A method for analyzing the mapping for structure is shown in Section 4.2. Therefore the clustering behavior on the data is investigated by several measures that are also explained in this section. Retrieval based on the embedded data is described in Section 4.3. Section 4.4 summarizes the chapter.

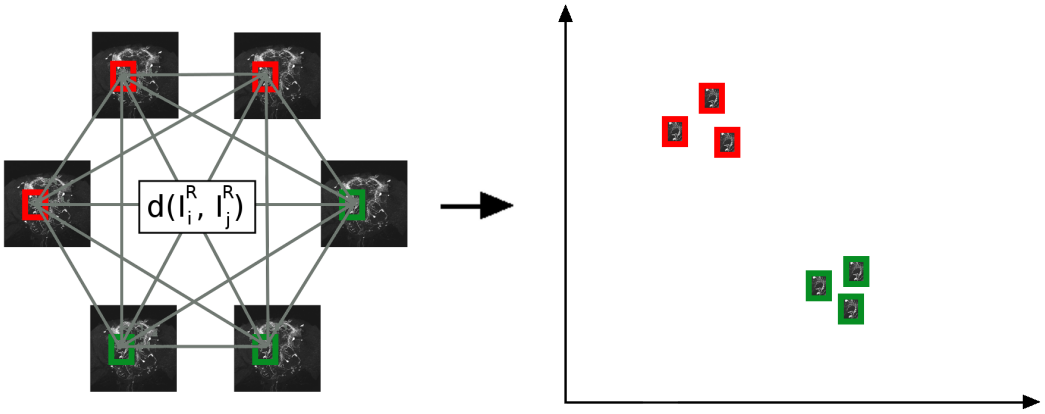


Figure 4.1: The distance function $d(\mathbf{I}_i^R, \mathbf{I}_j^R)$ is used to define the relations between a *drosophila population* to generate a low dimensional representation of the data which reveals the underlying structure of the data (two different groups, red and green).

4.1 Mapping

For non-linear mapping techniques like Diffusion maps, a distance metric for data points is needed [13]. For local brain areas, a similarity measure was introduced in Section 3.

For every brain volume $\mathbf{I}_i (i = 1..N)$, the distance to every other brain volume $\mathbf{I}_j (i = 1..N)$ can be computed within a sub area (defined by the superscript ^{R}) which corresponds to the query pattern used in the last chapter. If $d(\mathbf{I}_i^R, \mathbf{I}_i^R)$ is this distance from Equation 3.14, the distance matrix can be defined as

$$D_{i,j}^R = d(I_i^R, I_j^R) \tag{4.1}$$

$\mathbf{D}^R \in \mathbb{R}^{N \times N}$ captures the structure of the population, focusing on the region $\mathbf{R}$. Hence, it can be used to map the brains to a l dimensional diffusion space by adapting the basic Diffusion Map Algorithm [18]. Algorithm 4.1 uses the distance matrix $\mathbf{D}^R$ to generate a l-dimensional embedding of the data points (brain volumes). The principles of Diffusion maps are explained in Section 2.2. The distance matrix defines a graph of volumes $\mathbf{I}_i$, and the diffusion map represents transition probabilities of a random walk in this graph.

As one can see in Equation 3.14 (Section 3.4), the distance function is not symmetric (because the *GVF* is only computed on the query pattern). Because a condition for the Diffusion map algorithm is that the kernel matrix k is symmetric [18] ($k(x, y) = k(y, x)$), the distance matrix $\mathbf{D}$ needs to be symmetrized before the kernel function is applied.

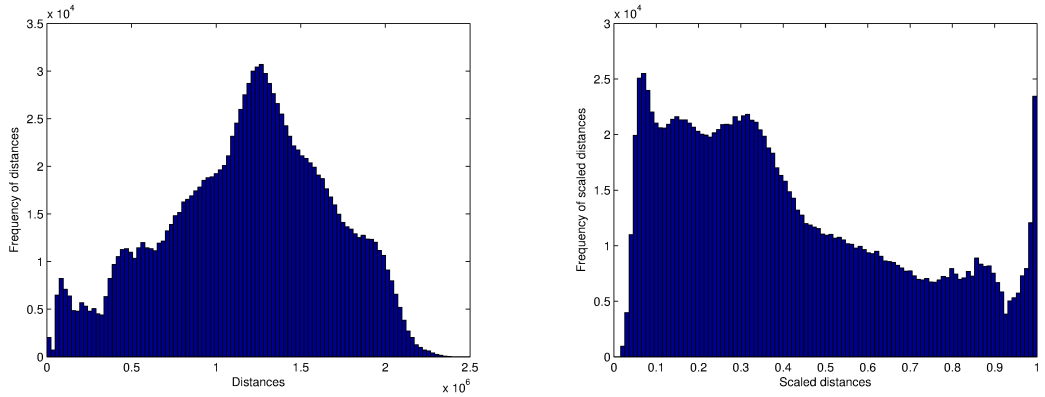
$$D_{i,j}^R = \frac{D_{i,j}^R + D_{j,i}^R}{2} \tag{4.2}$$

For the purpose of simplification, the symmetrized distance matrix is denoted as “distance matrix“ in this thesis.

The probability for walking between to data points $\mathbf{I}_i^R$ and $\mathbf{I}_j^R$ is calculated by using the kernel function

$$k(I_i^R, I_i^R) = e^{-\frac{(D_{i,j}^R)^2}{\sigma}} \tag{4.3}$$

which transforms values in the distance matrix by an exponential function to values between 0 and 1. $k(I_i^R, I_i^R)$ represents the local similarity measure within a neighborhood. The size of the neighborhood is defined by the kernel scale σ (distance of a walk on the graph) which is based on the structure and density of the data [18]. In the context of the distance measure which is used in this thesis, using the maximal distance between two points as σ transforms the data to the total space between 0 and 1. This can be seen in Figure 4.2 where (a) is the distribution of the distances of a set of 110 fly brains[1] accumulated over the entire brain (R = entire brain) and (b) is the distribution transformed by $\mathbf{k}(I_i^R, I_i^R)$.



(a) Distribution of the distances

(b) Distribution of the scaled distances

Figure 4.2: Distributions of the distances (a) and the transformed distribution (b) of a set of 110 fly brains

The resulting Kernel matrix $\mathbf{K}$ is then normalized by their row sums given by the diagonal matrix $\mathbf{Z}$ of to generate the probability matrix $\mathbf{M}$

$$\mathbf{M} = \mathbf{Z}^{-1}\mathbf{K} \tag{4.4}$$

The diffusion process on the graph is represented by calculating the diffusion matrix for diffusion time $\mathbf{t}$

$$\mathbf{M}^{(t)} = \mathbf{M}^t. \tag{4.5}$$

It represents the probability of walks along the graph edges geometry. An increasing value of t maps brains with low similarity closer together. The mapping itself is then computed by an eigenvector decomposition of $\mathbf{M}^{(t)}$. The first l-eigenvectors represents a l-dimensional diffusion space.

[1] **[D1]** set given in Section 6.1

Data: Distance matrix $\mathbf{D}^R$ for brain area $\mathbf{R}$, Kernel scale σ, timesteps t in the diffusion process, dimension of diffusion space l

Result: Lower dimensional mapping $\mathbf{Y}^R$

1 Calculate the probability for walking between two data points I_i^R and I_j^R by using the kernel function $k(I_i^R, I_i^R)$. The result is a kernel matrix $\mathbf{K}$

2 Compute the probability matrix $\mathbf{M}$ by normalizing by the row sums D of the Kernel matrix $\mathbf{K}$, so $\mathbf{M} = \mathbf{D}^{-1}\mathbf{K}$

3 The diffusion matrix at the timestep t is defined as $\mathbf{M}^{(t)} = \mathbf{M}^t$

4 Calculate the eigenvectors of the diffusion matrix

5 Use the first l-eigenvectors normalized by their eigenvalues to get the l-dimensional diffusion space

Algorithm 4.1: Adapted Basic Diffusion Mapping Algorithm [18]

The result of Algorithm 4.1 is a mapped *drosophila* population $\mathbf{Y}^R$ in an l-dimensional space.

An example for the mapping of a population for an area R can be seen in Figure 4.4. For this, a set of 110 fly brains[2] is used, where 35 contain the *pIP10* neuron [76]. The segmented neuron is shown in Figure 4.3.

The purple segmentation defines the pattern R which was used to map the population mapped to a 2D space by Diffusion maps in Figure 4.4 (b). The circles represent brains with *pIP10*, the crosses brains without *pIP10*. Figure 4.4 (a) shows a mapping by the linear dimensionality reduction method Multidimensional Scaling (MDS) [1]. While *MDS* can not separate the dataset, the Diffusion maps algorithm divides the dataset into two distinct groups. This maps the true relation (two groups, which are not similar to each other) better than the *MDS*.

4.2 Embedding-based clustering

Applying clustering algorithms on the embedded data is a way of investigating the structure of a *drosophila* population for a specific brain region **R**. Clusters in the data represents different groups of flies with similar neuronal structures (proof in Section 6.6). The purpose of this section is to introduce a method for finding a clustering of the data that maximizes several quality criteria.

As clustering algorithm, k-means clustering (Section 2.4) is used due to its simplicity. Additionally, it has been shown by Rakhlin and Caponetto [54], that the stability of k-means is defined

[2] **[D1]** set given in Section 6.1

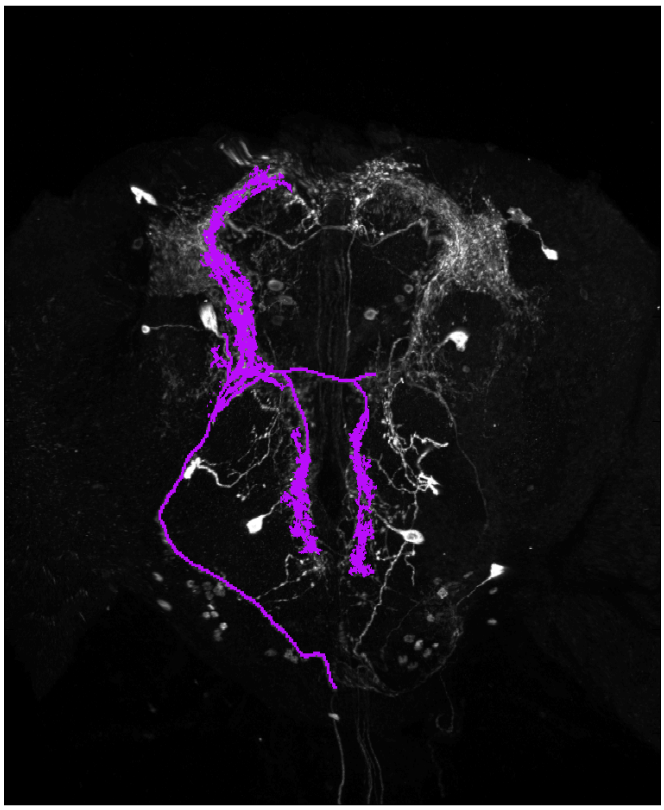
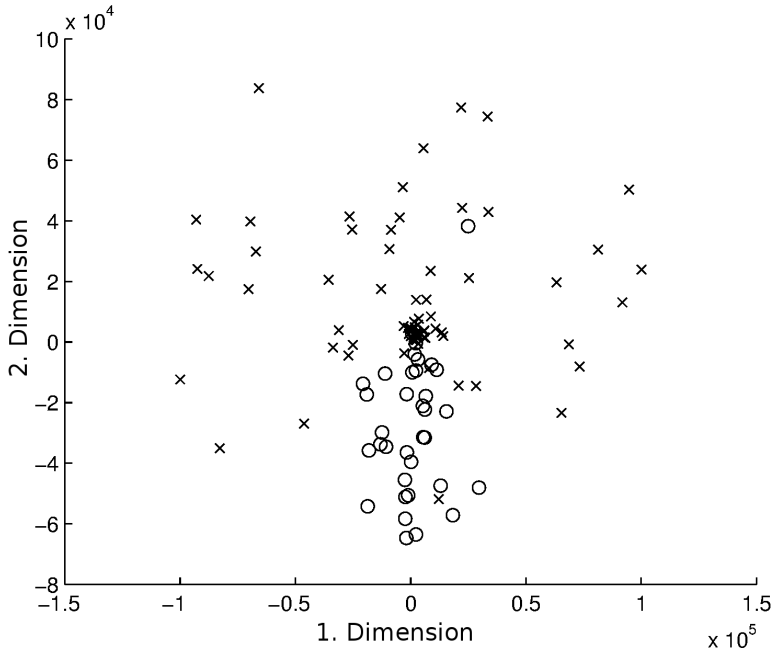


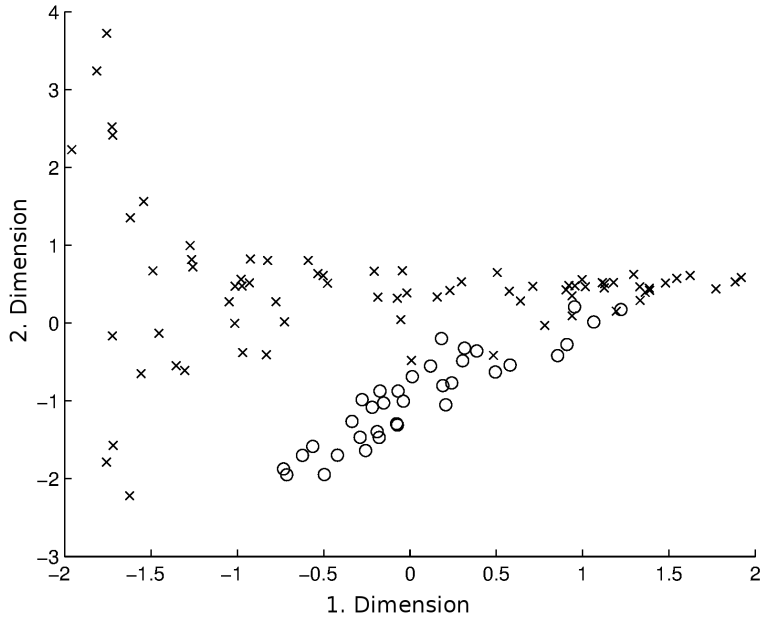
Figure 4.3: Fly brain with segmented *pIP10* neuron (purple)

by the geometry and underlying distribution of every class. Therefore it is appropriate to explore the data for structure. To perform k-means clustering, the amount of clusters k, the dimensionality of the embedding l and the diffusion time t needs to be estimated. For this, three quantitative evaluation functions are used.

Nearest Neighbor Mutation: To estimate the diffusion time t and the dimensionality of the embedding l, the mutation labels (*VT-Lines*) of brains are used to act as a ground truth. Fly brains with the same genetic mutation are samples of the same *genotype* (like clones), so fly brains of the same mutation are more similar to each other than to brains of an other mutation. If brains of the same mutation are not mapped close together (but close to a different mutation), information about the similarity is lost. A *Nearest Neighbor Mutations* (*NNM*) is defined as a mutation where all of its brains mapped closest together (brains of the same *VT-Line* are their nearest neighbors). The dimensionality of the embedding and the diffusion time can be determined, by maximizing the amount of *NNMs*.



(a) Mapping of the segmented area to a 2D space by MDS, The circles represent brains with *pIP10*, the crosses brains without *pIP10*



(b) Mapping of the segmented area to a 2D space by Diffusionmaps

Figure 4.4: Comparison between Diffusionmaps and *MDS* mapping methods. Both figures show the first two dimensions of the mapping The circles represent brains with *pIP10*, the crosses brains without *pIP10*.

Intra vs inter VT-line distance: In addition to the number of *NNM*, the distances within the brains of a *VT-line* (intra) and the distances between *VT-lines* (inter) can be compared. The mapping is more confident, if the brains of a *VT-line* are closer together in relation to the overall distance between the lines for the reason that brains of the same *VT-line* are samples of the same *genotype* (analog to *NNM*). By comparing the distribution of the intra and inter *VT-line* distance, and maximizing their difference, the dimensionality of the embedding and the diffusion time can be estimated.

Cluster Stability Index: The *Cluster Stability Index* (*CSI*) [40] which is described in Section 2.4 is used to investigate the structure of the data. The stability is calculated for k-means clustering for different k. A low (close to 0) *CSI* means high stability of the clustering, therefore structure in the data can be expected. The *CSI* is normalized to random labels, therefore a high (close to 1) *CSI* indicates a random clustering and in conclusion no structure.

The three measures can be used to evaluate clusterings on embeddings with different parameters. Clusterings that maximize the amount of *NNM* and the *Intra vs inter VT-line distance* while minimizing the *CSI* are considered to represent meaningful structures that can further investigated by a biologist. The application of those methods are shown in Section 6.3 and 6.6.

4.3 Embedding-based Retrieval

This section describes two methods of image retrieval on the mapped data for the purpose of a potential improvement. This is evaluated in Section 6.5. The retrieval can be either performed based on the distance on the mapping or by clustering. For both methods, the mapping $\mathbf{Y}^R$ needs to calculated first.

Distance on the mapping: The euclidean distance for every data point $\mathbf{Y}_i^R$ of the mapped data to the query pattern $\mathbf{Y}_Q^R$ is calculated.

$$d(\mathbf{Y}_Q^R, \mathbf{Y}_i^R) = ||\mathbf{Y}_Q^R - \mathbf{Y}_i^R||_2 \tag{4.6}$$

The order of the data points ascending by their distance to the query image represents the retrieval result.

Hierarchical distance: Another method is the use of the *hierarchical distance* on the mapping. The clustering behavior of the data is exploited for retrieval. Points that are more likely to be in the same cluster are specified to be more similar than points that are unlikely to cluster together. Therefore, a *hierarchical distance* is defined as the maximum tree depth in a complete linkage clustering dendrogram [63] that connects two points $\mathbf{Y}_Q^R$ and $\mathbf{Y}_j^R$ of the mapped data

$$d(\mathbf{Y}_Q^R, \mathbf{Y}_i^R) = \min depth(C_Q^R \cap C_i^R) \tag{4.7}$$

where C_Q^R is the set of clusters that contains brain Q and C_i the set of clusters that contains i. The basic idea of this metric is shown in Figure 4.5. The minimum tree depth of the path that connects (1) and (2) is 2, while the minimum tree depth of the path between (2) and (3) is 1. Therefore, (1) and (2) are closer together.

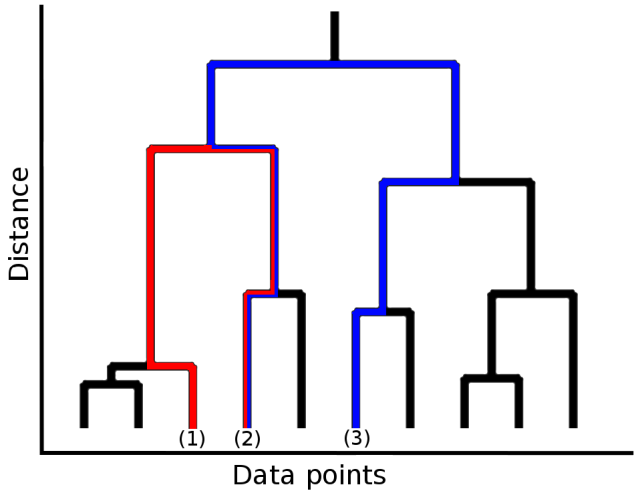


Figure 4.5: Example dendrogram to visualize the idea of a hierarchical cluster metric. Red and blue shows the path in the tree that connects the points (1) and (2), and (2) and (3)

Retrieval algorithm: The process of the two retrieval methods is described in Algorithm 4.2. The retrieval result is represented by ordering the brains by their hierarchical distance/euclidean distance in the embedding to the query brain.

Data: Binary mask R which locate the structure

Result: Classification error

1 Generate the distance matrix D^R

2 Compute the mapping Y^R of D^R

3 (Perform a hierarchical clustering on the mapping)

4 Order the brains by their hierarchical/euclidean distance to the query brain

Algorithm 4.2: Algorithm for embedding enhanced retrieval

4.4 Summary

Based on the distance function of Section 3.4 a method for non-linear mapping of *drosophila* populations for specific brain areas is introduced. The mapped data $\mathbf{Y}^R$ can be used for visualizing the relations between brains, as well as for investigation of the data by clustering algorithms. In addition, embedding-based retrieval methods are described.

CHAPTER 5

Similarity Visualization

In addition to finding structure in the data, the embedding can be used to detect regions in the *drosophila* brain that exhibit similar neuronal structures for a subset of the population. This means that non-linear mapping is used to identify similar regions within a set of brains S. Therefore, these regions need to be distinguished from regions with a high inter-individual variability. This is done by using the similarity criterion for brain areas which is described in Section 5.1. Because the computation of every possible region and their criterions is to expensive, the selection of areas is modeled as a multi-modal optimization problem. The method, which uses genetic algorithms for solving, is explained in Section 5.2. The visualization of the outcome is then defined in Section 5.3.

5.1 Similarity Criterion

A simple criterion can be created by applying the similarity measure of Chapter 3 between each *drosophila* brain $\mathbf{I}_i (i \in S)$ in the subset S of the population $\mathbf{I}_j (j = 1..N)$.

$$S \subset P \tag{5.1}$$

Regions with similar structures for all brains in the set have a lower mean distance compared to regions with a high variability (proof in Section 6). It will be shown in Chapter 6, that distances on the embedding represents the true geometric structure of the data better than the similarity measure only.

Let $\mathbf{Y}^R_{1..N}$ be the embedded data of a drosophila population for a region $\mathbf{R}$, and $\mathbf{D}^{Y^R}_{1..N}$ the distance matrix of $\mathbf{Y}^R_{1..N}$ based on the euclidean distances of the mapping. Therefore, on embedded data

$\mathbf{Y}^R_{1..N}$, the mean distance within a subset S of the population is defined as

$$\bar{\mathbf{D}}_S^{Y^R} = \frac{\sum_{i \in S} \sum_{j \in S} D_{i,j}^{Y^R}}{\#S^2} \tag{5.2}$$

where $\#S$ is the amount of *drosophila* brains in the subset. A calculation of the distance matrix for every possible area is too expensive, therefore, the *drosophila* brains can be divided into a set of non-overlapping cubic windows[1] $\mathbf{W}$ of equal size by a grid (*window grid*). For every window, a distance matrix $D^{W_i}_{1..N}$ of the $\mathbf{I}_j (j = 1..N)$ is created. Due to the additivity of Sum-of-Squared differences, the distance matrix $\mathbf{D}^{W_i+W_j}$ of a region which consists of the windows $\mathbf{W}_i$ and $\mathbf{W}_j$ is equal to the sum of their particular distance matrices $\mathbf{D}^{W_i}$ and $\mathbf{D}^{W_j}$. The effect of *GVF* and the differences in the structure tensors on the edges of the windows are ignored for simplification.

$$\mathbf{D}^{W_i+W_j} = \mathbf{D}^{W_i} + \mathbf{D}^{W_j} \tag{5.3}$$

The mapping $\mathbf{Y}^{W_i+W_j}_{1..N}$ can than generated by applying Algorithm 4.1 on $\mathbf{D}^{W_i+W_j}$. Hence, the criterion can be computed for every subset (every combination of windows) of $\mathbf{W}$. Subsets which minimize the criterion correspond to the similar regions. These subsets can be identified by the usage multi-modal optimization techniques.

5.2 Multimodal Optimization

The criterion can be computed for every single window. The disadvantage of this method is, that it captures only local similarity. Structures which cover more than one window are split and not treated as one structure. Therefore, windows can be combined (Equation 5.3) and recombined. Finding combinations that minimize the criterion $\bar{\mathbf{D}}_S^{Y^R}$ (Equation 5.2), where $\mathbf{R}$ represents a region of combined windows, can be seen as optimization problem.

The principle idea is to find all (or at least a representative subset of) possible solutions. A solution is defined as an individual[2] P_i (of a population P) which represents a connected set of windows (a region) that minimizes the criterion $\bar{\mathbf{D}}_S^{Y^R}$. For the usage of genetic algorithms, the region $\mathbf{R}$ of an individual can be expressed as a binary string $\mathbf{R}_B$ to act as the genome.

$$\mathbf{R}_B = \{0, 1\}^{\#W}. \tag{5.4}$$

$\mathbf{R}_B$ is a bit-string of length $\#W$, where $\#W$ is the amount of non-overlapping windows that split the brain. If the ith element of $\mathbf{R}_B$ is 1, $\mathbf{W}_i$ is part of the region that is represented by the individual. The criterion (Section 5.1) is defined as fitness. In this case, the genetic algorithm needs to minimize the fitness function. Therefore, a "better" fitness is denoted as smaller.

$$fitness(P_i) = \bar{\mathbf{D}}_S^{Y^R_B} \tag{5.5}$$

[1] 3D bounding box

[2] connected set of windows

The process of the calculation of the fitness is also described in detail in Algorithm 5.1.

Data: Distance matrix $\mathbf{D}_{1..N}^{W_i}$ for every window $\mathbf{W}_i (i = 1..N)$, genome $\mathbf{R}_B$ of the individual P_i, subset of interest S

Result: fitness of P_i

1 Calculate $\mathbf{D}_{1..N}^{R_B}$ according to Equation 5.3 (summarize the distance matrices of all windows that are in the genome $\mathbf{R}_B$.

2 Generate the embedded data $\mathbf{Y}_{1..N}^{R_B}$ by applying Algorithm 4.1 on $\mathbf{D}_{1..N}^{R_B}$

3 Calculate the euclidean distance matrix of the subset $\mathbf{D}_{S}^{Y_B^R}$ on the embedded data $\mathbf{Y}_{1..N}^{R_B}$

4 fitness(P_i) $=_{S}^{Y_B^R}$

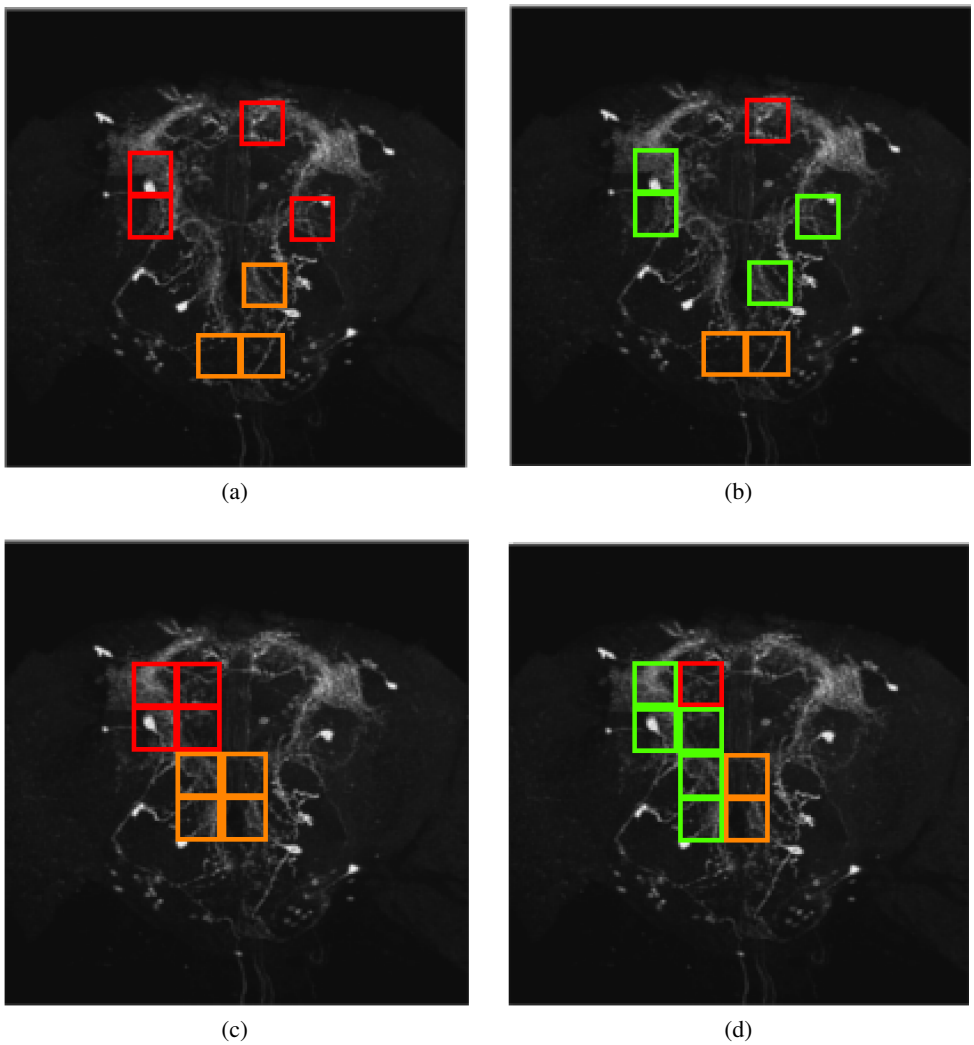
Algorithm 5.1: Calculation of the fitness of an individual

The fitness of regions found indicates the similarity of the covered area which can be used to visualize the variability by a heatmap (Section 5.3). Because basic genetic algorithms are only capable of finding one possible solution [45], spatial selection [14] and crowding [17] is adapted for multi-modal optimization. The principle parts of the optimization are:

- **Spatial Selection** [14] places individuals of a population randomly on a two dimensional map to establish spatial isolation between them. Only parents which are close together (i.e. they can reach each other by a r steps) can perform a crossover and make children. This *simulated environment* allows a separate development of subpopulations and therefore a convergence in distinct solutions [14]. The selection process is performed by choosing a random individual P_{random} an perform two r-step random walks to get two parents. The crossover of the parents P_{new} will replace the individual P_{random} if

$$fitness(p_{new}) > fitness(P_{random}) \tag{5.6}$$

- **Crossover** merges two individuals of the population P to a new child and computes its fitness. For this, the child consists of a subset of the parent windows. An example for this can be seen in Figure 5.1 (a) and (b). Because a region is a connected area (otherwise it would not be a region but more than one) also a connectivity constraint is used which means, that only connected components are valid solutions. This includes that only parents that lie next to each other can produce these components. Figure 5.1 (c) and (d) visualize this concept. The constraint also reduces the amount of possible solution.



(a) (b)

(c) (d)

Figure 5.1: (a) Two parents (red and orange windows) without connectivity constraint, (b) Produced child (green windows) by a uniform crossover of (a), (c) Two parents (red and orange windows) with connectivity constraint, (d) Produced child (green windows) by a connectivity constrained crossover of (c)

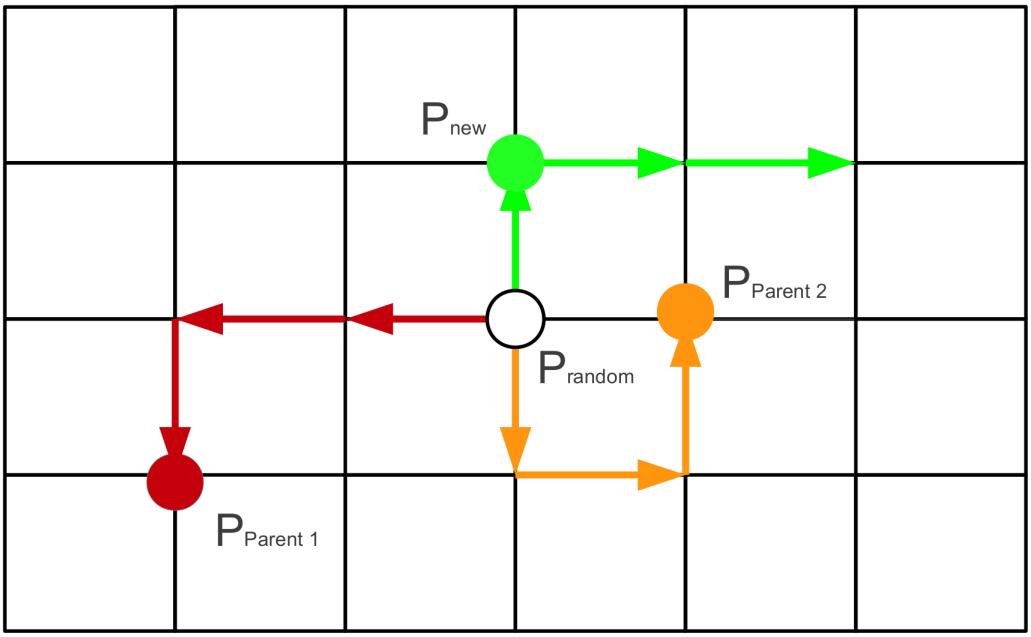


Figure 5.2: Concept of Crowding and Spatial Selection. The arrows form random walks on the *spatial grid*, P_{random} is a random initial point, $P_{Parent1}$ and $P_{Parent2}$ are the parents and P_{new} is the child.

- **Crowding** [17] is an extension of *Spatial Selection* which effects the insertion into the population. *Spatial Selection* would always replace individuals with lower fitness than the new individual P_{new}, and removes thereby potential useful information from the population [69]. The use of crowding avoids this by only replacing individuals with a similar genome (=windows) to P_{new}. This is performed by computing a r-step random walk from a random individual P_{random}, and replacing the individual which has the highest window-overlap to P_{new} on the walk. An illustration of this process can be seen in Figure 5.2.

- **Mutation** To increase diversity of the population, a random window is added to or removed from the new individual P_{new} to generate $P_{mutated}$. If it is still a connected component, use $P_{mutated}$ as new individual, otherwise use P_{new}.

The combination of these techniques are used in Algorithm 5.2. The parameters of the algorithm are the population size $\#P$ and the random walk steps r. A high population size increases the diversity, but is computationally expensive [69]. A low amount of random steps leads to a higher amount of distinct solutions (compared to a high amount of random steps [69]), but a slower convergence [69]. A constraint of the search space is the connectivity of a region. A region consists of connected windows, otherwise it would not be a single region but more

than one.Another point is that connected regions allow a more directed evolution compared to uniform crossover since uniform crossover selects a random subset, which can also be a disconnected solution, while the result of the crossover method in Algorithm 5.2 is already valid (as connected region). This concept is shown in Figure 5.1. Figure 5.1 (a) shows two parents (red and orange windows) without connected regions which make a child by a uniform crossover (Figure 5.1 (b)). The connectivity-constrained crossover selects a connected random subset of two parents (red and orange windows in Figure 5.1 (c)) which leads to a child that is also a connected region (green window in Figure 5.1 (d)).

The initialization of the algorithm can exploit the spatial selection in combination with the connectivity constraint. Giving the spatial selection map the same aspect ratios as the *window grid* of the windows, the individuals can be placed on the map according to their window locations. This means that individuals that are able to perform a crossover due to the connectivity constraint are close together and different brain areas can develop a optima separately from each other.

The output of Algorithm 5.2 is a set of solutions (the final population) which minimize the mean distance of all brains to each other in distinct regions. The algorithm is finished when the maximum amount of epochs e is reached. The visualization of the solutions by a heat map is described in Section 5.3.

Data: Distance matrix $\mathbf{D}_{1..N}^{W_i}$ for every window W_i, population size $\#P$, amount of random steps r, amount of epochs e

Result: A set of distinct solutions (final population) P consisting of their areas fitness

1 Generate a random population P with $\#P$ individuals (=connected area)

2 (Spatial Selection) Place every individual on a $\sqrt{\#P} \times \sqrt{\#P}$ 2D *spatial grid* according to their location of the windows

3 **for** $1 - e$ **do**

4 (Spatial Selection) For every point on the *spatial grid* (in a random order), perform two r-step random walks to get the parents

5 (Crossover) If the parents form a connected area, choose a random sub-area to form the child P_{new}

6 (Crowding) Perform a r-step random walk. Take the individual $P_{i,j}$ on the way, which has the highest overlap to the child P_{new}

7 (Mutation) Choose a random window. Add it or remove it from P_{new} to generate $P_{mutated}$. If it is still a connected area, $P_{new} = p_{mutated}$

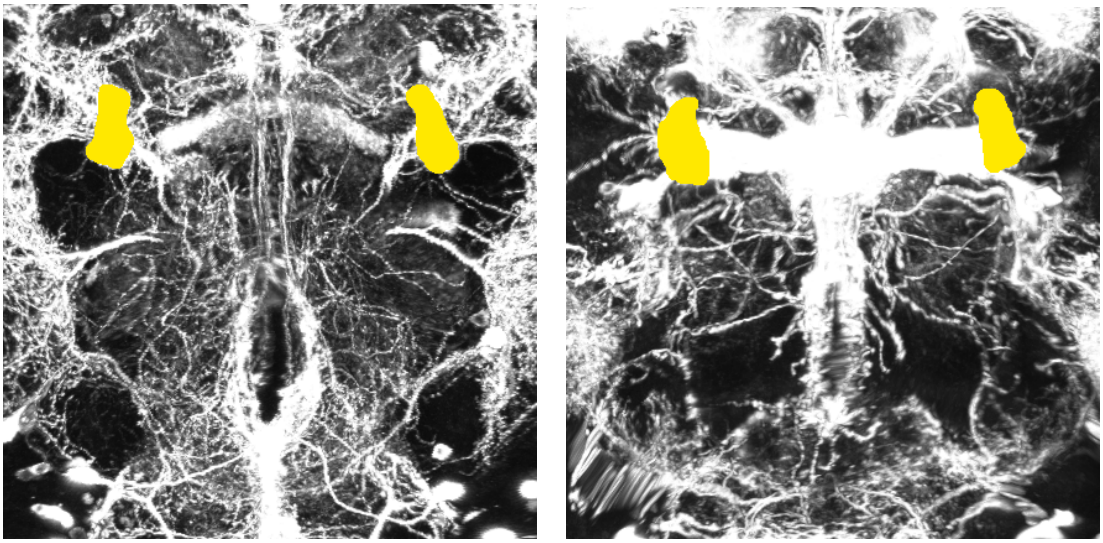
8 If the fitness of the child P_{new} is better than the fitness of individual $P_{i,j}$, replace $P_{i,j}$ with the P_{new}

9 **end**

Algorithm 5.2: Multimodal optimization of the search for similar regions

5.3 Visualization

The visualization of similar regions is based on the output population of Algorithm 5.2. As example for the visualization, respectively as input data, serves **[D2]**, the similarity is visualized for **[SVGT2]** subset (part of **[D2-EXT-VIS]**) of Section 6.1. The maximum intensity projections (*MIP*) of two representative brains are shown in Figure 5.3 with labeled areas (yellow, part of the *mushroom body*) which are annotated to be similar. The *MIPs* include only slices which contain the annotated regions to allow visual determination.



(a) (b)

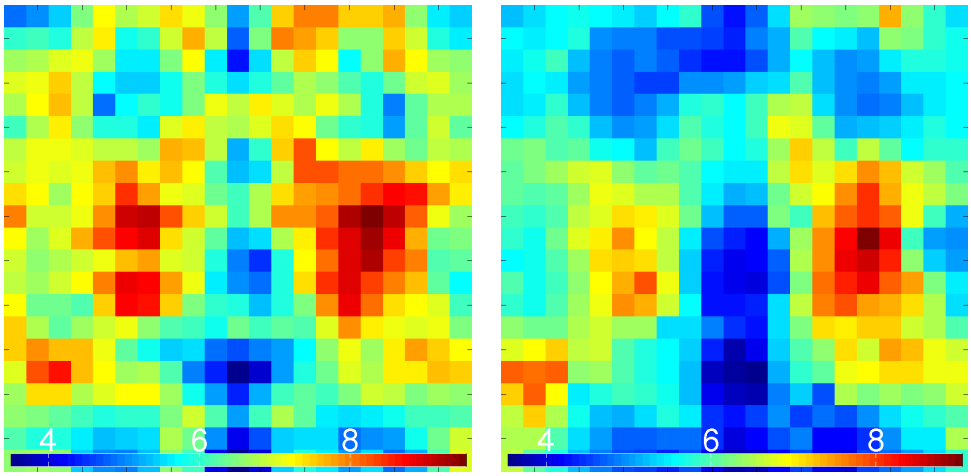
Figure 5.3: Maximum intensity projections of two example brains of two different mutations with labeled ground truth (area which is annotated to be similar)

By computing the fitness only for single windows (an individual P_{W_i} consists of only one window W_i), and hence without optimization, the fitness/ similarity-heatmap captures only local similarity. Structures which cover more than one window are not treated as one structure. This means, that the different windows show different similarities although they cover the same structure. This can be seen in Figure 5.4 (a) especially for the vertical middle structure in Figure 5.3. Dark blue represents high similarity, while dark red and brown indicates low similarity. For 2D visualization, only the layers (on the z-axis) which contains the *mushroom body* are used, and the fitness of the windows is summed over the z-axis.

After using Algorithm 5.2, the result looks different (see Figure 5.3 (b)). The visualization is done by plotting for each window W_i the mean fitness of all solutions that contains W_i. The image looks more smooth compared to Figure 5.4 and homogenious in the regions that contain similar regions. Also the overlap with the annotated similar regions is increased compared to the single-window visualization. As one can see, also a long vertical area is blue-colored. This is not in contradiction with the ground truth, because the annotations are only covering the *mushroom-body* (more details in Section 6.1).

Another method for visualization is not only to use the mean of the mapped distances $\bar{\mathbf{D}}_S^{Y^{R_B}}$, but comparing the mapped distances of the subset of interest $\mathbf{D}_S^{Y^{R_B}}$ with the mapped distances of the remaining dataset $\bar{\mathbf{D}}_{N\setminus S}^{Y^{R_B}}$ by an one-sided U-Test ($utest(x_1, x_2)$). The resulting p-values are then used as fitness.

$$fitness_U(P_i) = utest(\bar{\mathbf{D}}_S^{Y^{R_B}}, \bar{\mathbf{D}}_{N\setminus S}^{Y^{R_B}}) \tag{5.7}$$



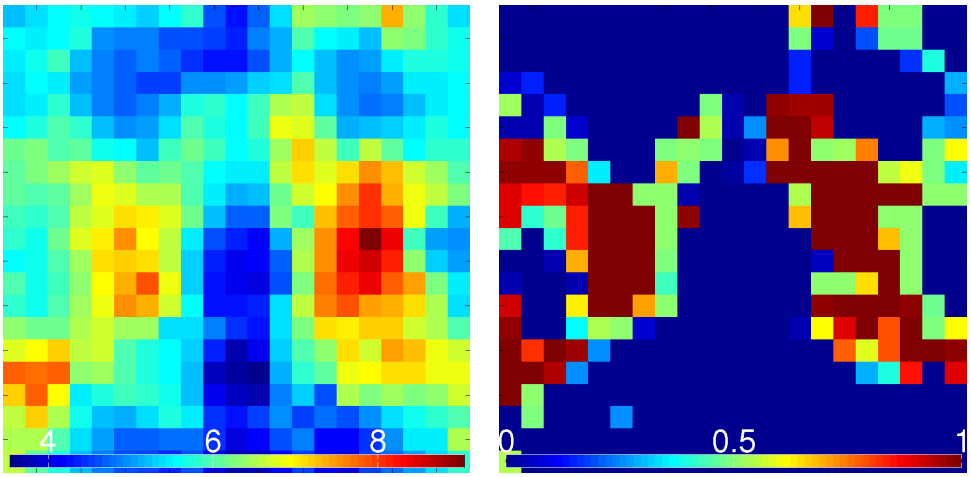
(a) (b)

Figure 5.4: Fitness of every single window ($fitness(P_{W_i})$) (a) and the mean fitness ($fitness(P_i)$) (b) of every window of the optimized solutions. Both figures show only windows within the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity).

The result of this approach is visualized in Figure 5.5 (b). The results similar to the mean fitness 5.5 (a), but less smooth and stronger distinction between similar and not-similar.

5.4 Summary

This chapter introduced a method for finding similar neuronal structures for a subset S of *drosophila* brains. A criterion $\bar{\mathbf{D}}_S^{Y_B^R}$ was defined for a region $\mathbf{R}_B$ that consists of a set of cubic windows. By using multi-modal optimization techniques, combinations of windows can be found that minimize the criterion. The resulting solutions can be visualized by a heatmap to show neuronal structures that are similar for the subset S.



(a) (b)

Figure 5.5: Mean fitness ($fitness(P_i)$) (a) and mean p-value ($fitness_U(P_i)$) (b) of every window of the optimized solutions in the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity).

CHAPTER 6

Validation and Results

This chapter is about the validation of the methods introduced. Section 6.1 gives details about the experiment datasets which are used for the the images and experiments of this thesis. To facilitate reading, the section is subdivided into experiments, each evaluating a particular aspect of the methodology. Each experiment-sections consist of the formulation of the aim, the corresponding evaluation function, the used dataset as well as the results and the discussion. Section 6.2 describes the evaluation of the similarity measure by image retrieval. Section 6.3 describes the validation of the non-linear mapping and how the structure of a population can be summarized. The similarity visualization on brains is the subject of Section 6.4. The embedding can be also used for enhancing the image retrieval of Section 6.2. This is described in Section 6.5. In addition, a completely exploratory analysis of data is performed and discussed in Section 6.6. All results of the performed experiments were computed by using optimized parameters which are estimated and explained in Appendix A.

6.1 Experiment Data

The set-up is based on expert labeled data which consists of 1151 3D drosophila brain volumes acquired by a Zeiss LSM 510 confocal microscope, that can visualize traced neurons. The volume data is stored as 100MB Amira files [2]. The last 97.320.960 bytes of a file are a $768 \times 768 \times 165$ matrix, which express the brain-volume by intensity values between 0 and 255 (Figure 1.1). The data was exported by BrainGazer [11] and provided by the Barry Dickson Lab [19].

The testset can be divided into four parts:

- Data set 1 **[D1]** consists of 110 3D volumes, which are related to the *pIP10* neuron [72]. 35 brains show the *pIP10* neuron, so they are similar to each other, while the remaining 75

brains are negative for this neuron. Figure 6.1 shows two fly brains with 2 segmented *projections* and 3 segmented *arborizations* of this neuron. *projection* L_pIP10, arborization A,B and C belongs to the same neuron (L_pIP10), while R_pIP10 is the corresponding neuron in the other hemisphere of the brain. These five structures are the test cases for the retrieval evaluation which is described in Section 6.2 and the enhanced retrieval in Section 6.5.

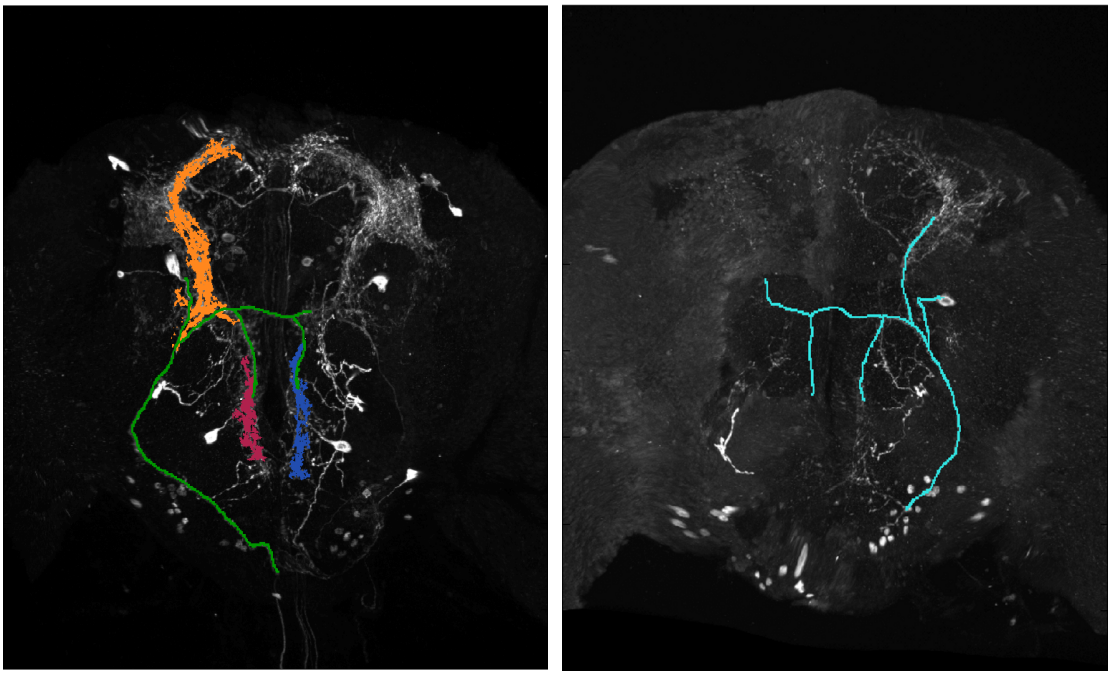
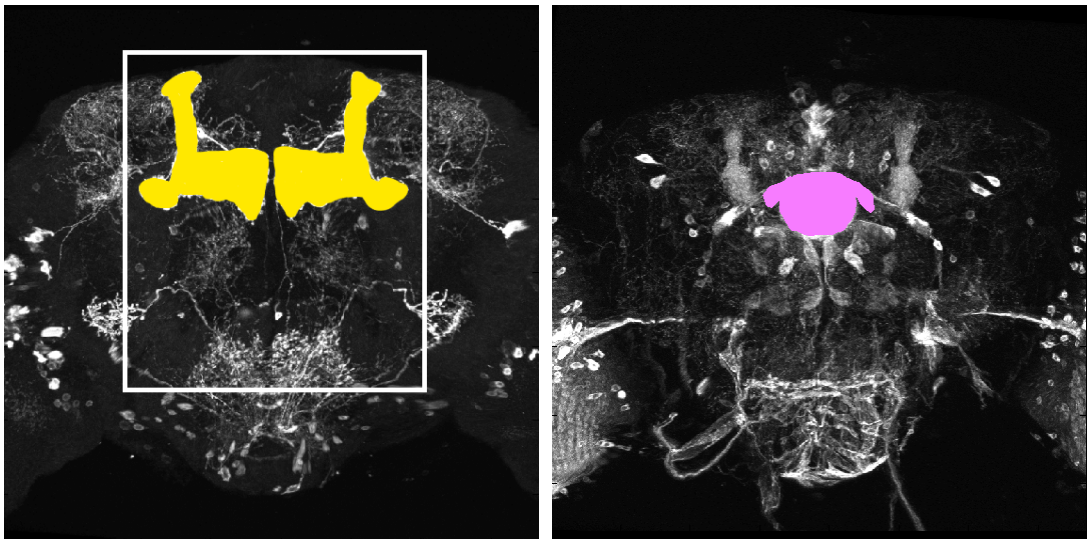


Figure 6.1: Segmented structures: projection L_pIP10 (green), projection R_pIP10 (cyan), arborization A (orange), arborization B (red), arborization C (blue). The colours correspond to Figure 6.6 and 4.3 (the purple structure in Figure 6.6 and 4.3 is the combination of L_pIP10, arborization A,B and C)

- Data set 2 **[D2]** consists of 1041 various drosophila brains without a priori known predefined appearance characteristics. For each brain the corresponding *VT-line* is known. The 1041 brains contain overall 350 *VT-lines*, with each *VT-line* being represented by 2-6 (on average 3) example brains. This set is used for the evaluation of the embedding and summarizing the structure of a population 6.3.

- Data set 3 **[D2-EXT]** is a sub set of **[D2]** with 62 brain. It is named *EXT* ("external neuron") after the neuron which can be found in the *Mushroom-body* and the *Central complex* region. The regions are annotated and shown in Figure 6.2.

- Data set 4 **[D2-DPM]** is also sub set of **[D2]** and consists of 24 brains with expressed *DPM* ("dorsal paired medial") neuron in the *Mushroom-body* region (Figure 6.2).



(a) Mushroom Body (b) Central complex

Figure 6.2: Two example brains of the testset where *Mushroom-body* is yellow colored and the *Central complex* is pink. The white border marks the area which is shown in Figure 6.3

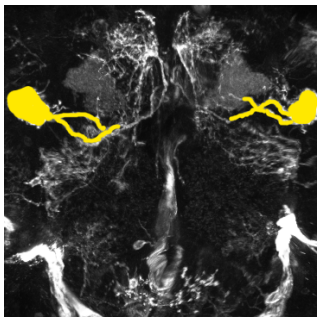
- Data set 5 **[D2-EXT-VIS]** is identical to **[D2-EXT]**. However, for a sub set of **[D2-EXT]**, labels of 9 different structures of 6-12 images each (each containing 2 *VT lines*) for which the morphological structure in the *Mushroom body* is particularly similar. Figure 6.3 shows one example brain of every subset with labeled annotations.

6.2 Evaluation of retrieval based on distance function

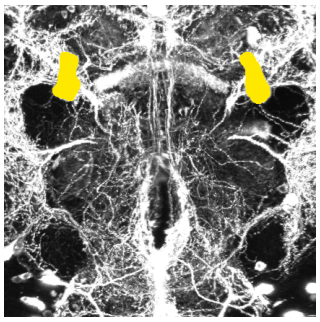
This section describes the verification of the distance function of Chapter 3. No rankings or other measures are available as ground truth. Instead manual annotations of the volumes by an expert who classified each volume if it contains the *pIP10* neuron (*projection* + corresponding *arborization*) or not are available. The annotated volumes represents 110 images of the experiment data **[D1]** which were mentioned in Section 6.1.

Research Question

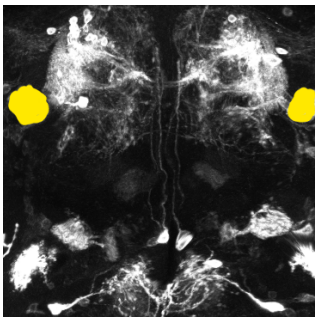
Can the distance function (Section 3.4) be used for content based image retrieval, to retrieve cases that contain a specific query structure? Given 6 query structures (Figure 6.1), can cases that contain the same structures be retrieved?



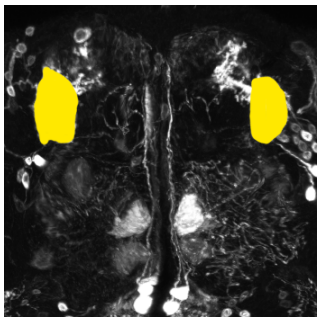
(a) SVGT1: Crossing the middle of horizontal lobes



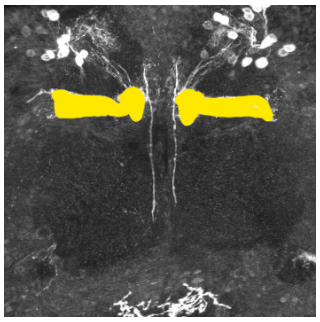
(b) SVGT2: Heel



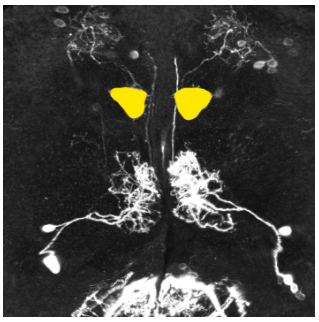
(c) SVGT3: Heeltip



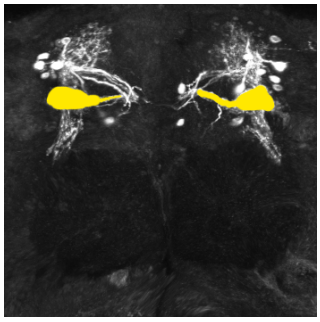
(d) SVGT4: Heeltip + vertical lobe



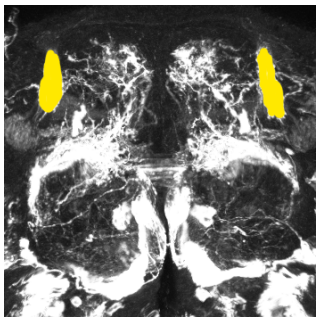
(e) SVGT5: Lower horizontal neuron



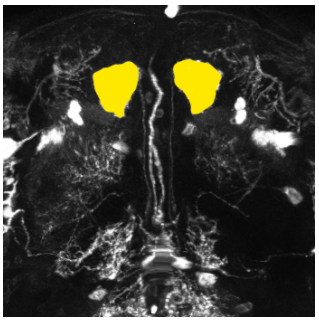
(f) SVGT6: Medial heeltip



(g) SVGT7: Medium horizontal lobes



(h) SVGT8: Vertical lobes



(i) SVGT9: Medial Uppertip

Figure 6.3: Example brain of every Similarity Visualization Ground Truth (SVGT) with labeled annotations (yellow) **[D2-EXT-VIS]**

Validation method

The retrieval is performed on datset **[D1]**. By using the measure as a discriminator for *pIP10* and *not pIP10* labels, the classification error indicates the performance and quality of the method. The theoretical decision boundary is the amount of equal volumes. *pIP10* volumes with a sim-

ilarity to a query pattern below the boundary, and *pIP10* volumes above the boundary are false classified. An example of this concept can be seen in Figure 6.4. It shows the density of the distribution of *not pIP10* and *pIP10* volumes over the rank and the theoretical decision boundary.

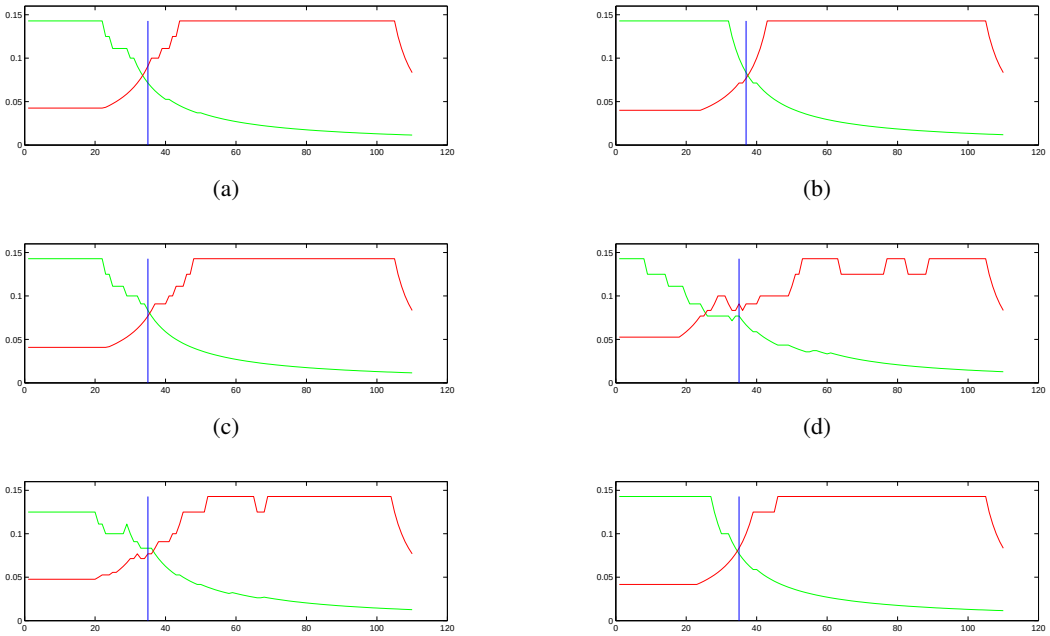


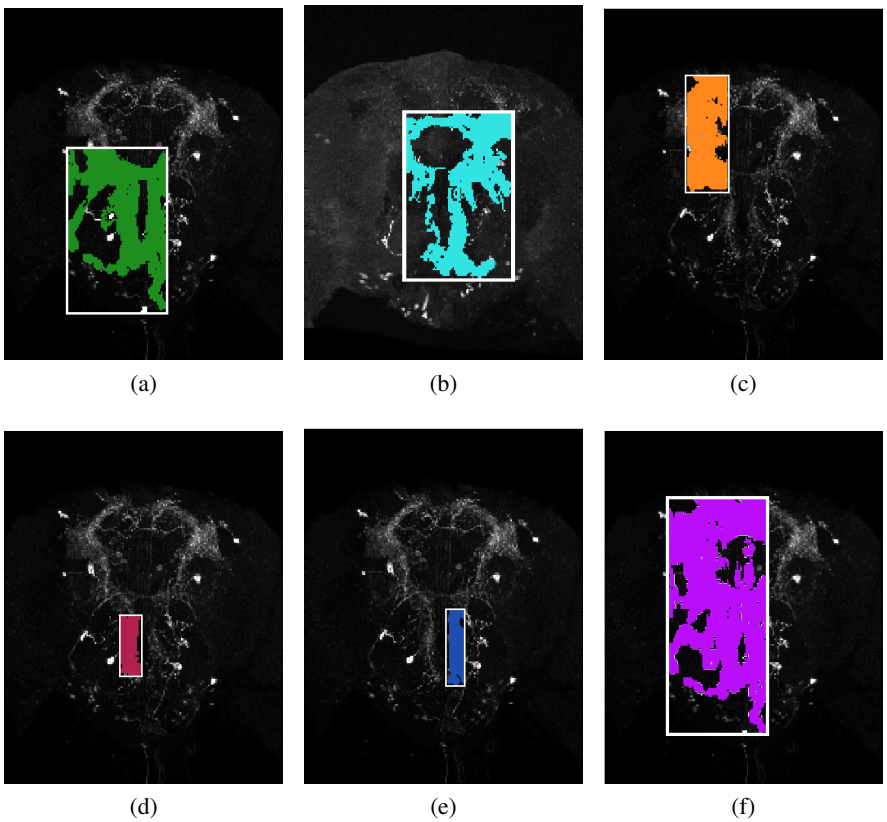
Figure 6.4: The Figure shows the distribution (y-axis) of *pIP10* (green)/*not pIP10* (red) volumes over their search rank (x-axis) (the blue line is theoretical decision boundary) for all 6 testruns (row-wise order: L_pIP10 proj, R_pIP10 proj, arbA, arbB, arbC and L_pIP10

To validate the search algorithm, three different query region definitions for each search-structure (L_pIP10, R_pIP10, arborization A,B and C) were applied and the classification error calculated. For the experiments, the optimized parameters (Appendix A) were used.

1. **Manual query region segmentation**: The pre-segmented structures of Figure 6.1 were used as query pattern. The segmentations are given as binary masks. The *query area* (area which is compared between the images) is defined by the (morphological) dilation of the binary masks.

2. **Automatic segmentation within a bounding box**: To evaluate query region selection via bounding box, we artificially created a bounding box around the query region. The query region was then automatically determined as described in Section 3.2. The selected structures are shown in Figure 6.5. As one can see, it is not possible to select only *projections* due to their direct connection to *arborizations*. This could be solved by split the *projection* into smaller sub-regions. In this case it is not necessary because the *arborizations*

and *projections* of the same neuron are usually present simultaneously (so they have the same ground truth). The *query area* can be derived by a (morphological) dilation of the segmentations.



(a) (b) (c)

(d) (e) (f)

Figure 6.5: Structures segmented by largest topology within a window (white): projection L_pIP10(projection,green), projection R_pIP10 (*projection*,cyan), arborization A (orange), arborization B (red), arborization C (blue), L_pIP10 (projections + arborization A,B and C, purple). The colors are corresponding to Figure 6.6 and 6.1

3. **No segmentation within a bounding box**: To see the effect of the automatic segmentation of the query pattern, also a retrieval within a window without segmentation is computed. The *query area* corresponds to the windows of the automatic segmentation in Figure 6.5.

To validate the methods, the results shall be also compared to a voxel-based method. The method is straight forward: The query pattern and the volumes are thresholded via Otsu's method [49] within the *query area* (defined by the windows which are also used for the automatic/no segmentation methods) shown in Figure 6.1). Then, the resulting binary images are compared by the DICE coefficient.

Results and Discussion

The results of the three test cases of this section are shown and discussed in the following.

Table 6.1 shows that the manual query region segmentation approach performs with a mean classification error of 8% better than the other methods on average. This is caused by the accurate selection of the structure compared to the automatic segmentation. In relation to the manual selection, the segmentation of the largest topology (automatic segmented), which is described in Section 3.2, increases the classification error for neurons to 9.3% because the form of the neuron and the rectangular window leads to a selection of more structures than necessary. If these structures do not correlate with the ground truth, the classification error increases. The classification error is higher or at least equal to the automatic segmentation method (on average 13.2%) . This allows to conclude, that the segmentation improves the results of the retrieval. The voxel-based method is faster than the other methods because it does not need to preprocess structure tensors. However, Table 6.1 shows, that the mean classification error is with 26.3% worse compared to the other two methods.

The algorithms were also tested on the whole L_pIP10 neuron (L_pIP10 + arbA + arbB + arbC). The additional information improved the result for the pre-segmented method but also increased the error for the automatic segmentation.

Figure 6.6 gives an example for the outcome of a ranking based on the similarity to a query pattern. Pre-segmented test-cases (first image of every row, the colors are corresponding to Figure 6.6 and 6.1) are (are used to compute a similarity to the other brains of the testset. The top rankings can be seen on the left side of the black line, while the least ranked are on the right side.

	pre-segmented	automatic segmented	no-segmentation	voxel
proj. L_pIP10	0.09	0.07	0.14	0.29
proj. R_pIP10	0.03	0.05	0.14	0.09
arbA	0.07	0.09	0.10	0.25
arbB	0.16	0.12	0.18	0.30
arbC	0.10	0.14	0.14	0.36
L_pIP10	0.03	0.09	0.09	0.29

Table 6.1: Classification error of the pre-segmented method and the method with largest topology segmentation compared to a voxel based method. The top rankings for the pre-segmented method can be seen in Figure 6.6

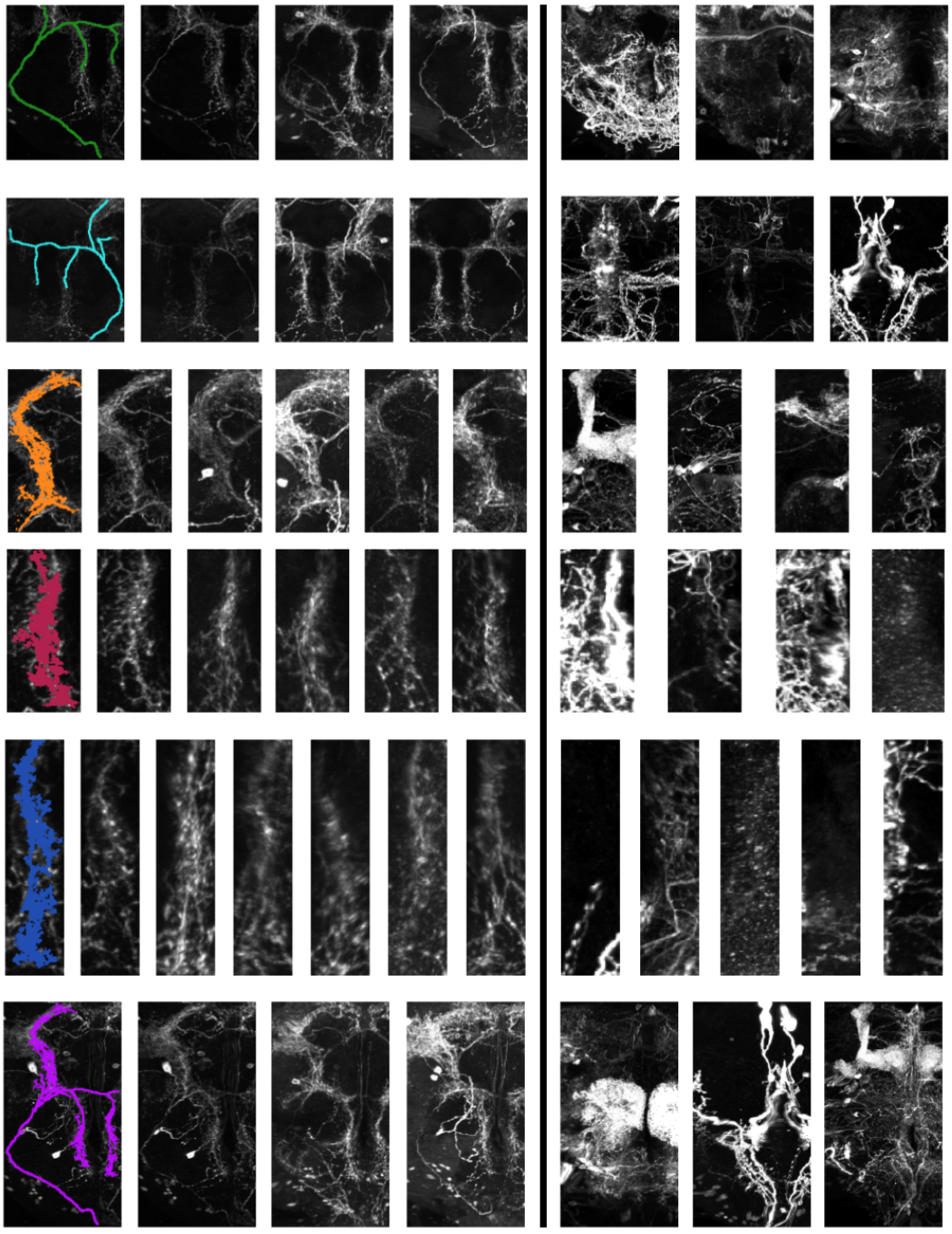


Figure 6.6: Top retrieval ranking of 6 testruns on the left side, least ranked on the right side of the black line (row wise, first image is the query pattern)

6.3 Evaluation of embedding and summarizing the structure of a population

Non-linear mapping is a central part of this thesis, therefore this section describes the validation of embeddings of large datasets like **[D2]**. In this experiment, the mapped data of **[D2]** is investigated *Mushroom body* area (Figure 6.2) by quantitative methods that allow to draw conclusions regarding the structure of the population.

Research Question

Is it possible to find relevant structure in the population by clustering in an embedding space that is based on local appearance similarity?

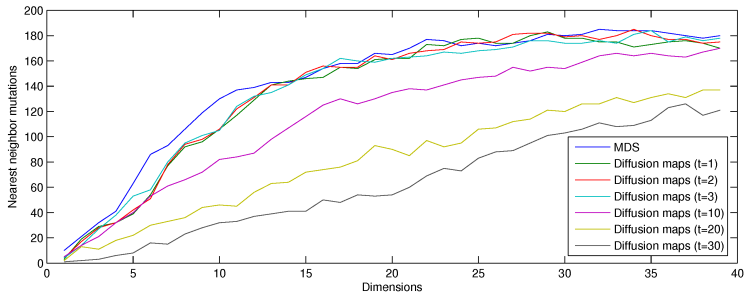
Validation method

The principle question of this experiment is how the mapped data tells something about the characteristics of a population. Multidimensional scaling (as example for a linear method) and Diffusion maps (as example for a non-linear method) shall be compared on the **[D2]** dataset for the *Mushroom body* area (Figure 6.2) regarding to their performance and clustering behavior. For the validation of the mapping techniques quantitative evaluation functions are used: Amount of *Nearest Neighbor Mutations*, Intra vs. inter VT-line distance and the *Cluster Stability Index*. The methods are explained in Section 4.2.

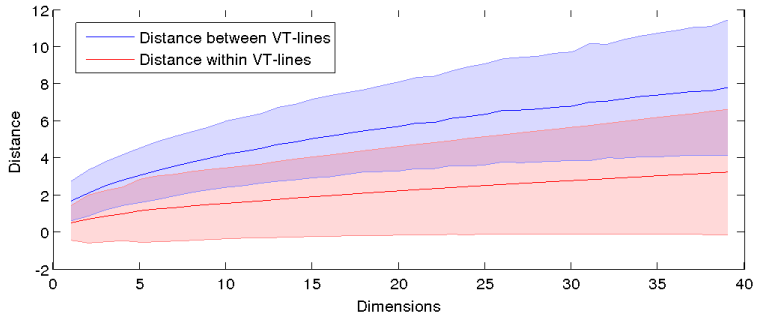
The parameters of Diffusion maps (Diffusion time, σ and dimensionality) are varied and compared to *MDS* regarding to their quantitative performance (*NNM* and *Cluster Stability*) on the **[D2]** data.

Results and Discussion

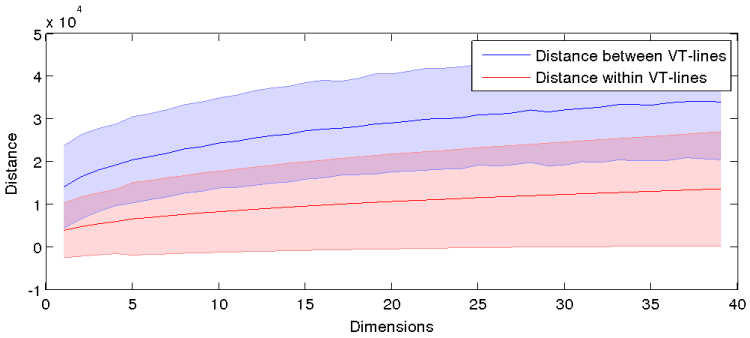
Figure 6.7 (a) shows the amount of *NNM* in relation to the dimensions of the generated embeddings for *MDS* and Diffusion maps at different diffusion times. The performance of both methods, Diffusion maps and *MDS*, are equal for dimensions above 20, and better for *MDS* below. There is no significant difference for low diffusion times ($t \leq 3$), so $t = 1, 2, 3$) are possible. This suggests that intra-individual variability (relation between brains of the same mutation) can also mapped in a linear way. The relation between the distance between *VT-lines* and the distance within *VT-lines* is shown in Figure 6.7 (b) for Diffusion maps, and in (c) for MDS. The bold lines correspond to the mean distances within *VT-lines* (blue) and between *VT-lines* (red). The blue and red areas mark the standard deviation, therefore an overlap conforms to a not significant difference. A closer mapping of brains of the same *VT-line* compared to the overall distance between the lines corresponds to an approximation of the ground truth. Both



(a) Nearest Neighbor Mutations vs dimension of the embedding



(b) Diffusion maps: Distribution (given by the mean and the standard deviation) of the *Intra VT-line distance* (red) and the *Inter VT-line distance*. The dimensions of the embedding is indicated on the x-axis

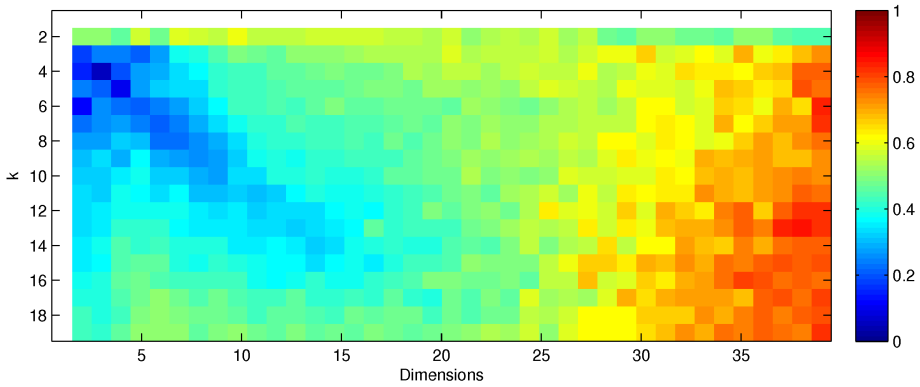


(c) MDS: Distribution (given by the mean and the standard deviation) of the *Intra VT-line distance* (red) and the *Inter VT-line distance*

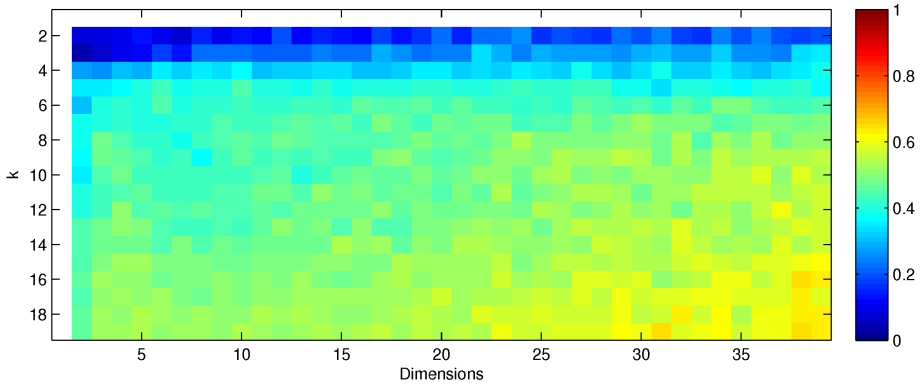
Figure 6.7: Performance of the mapping depending on the dimension of the embedding

methods exhibit the best mapping below 20, whereby *MDS* performs better for higher dimensions. This complies to the outcome of the *NNMs*.

Figure 6.8 (a) and (b) describe the relation between the dimension of the embedding, the number of clusters k and the cluster stability in two heatmaps. The *CSI* was computed for different k (2-20). The colors of the heatmaps correspond to the stability. Dark blue means a high stability



(a) Diffusion maps: Cluster Stability Index vs dimension of the embedding



(b) MDS: Cluster Stability Index vs dimension of the embedding

Figure 6.8: Performance of k-means clustering depending on the dimension of the embedding

of the clustering, red means low stability. A high stability indicates clusters, even if data points are randomly sub-sampled, and therefore structure in the data. *MDS* shows low stability for all dimension for $k > 4$. For $k \leq 4$, there is a high stability for all dimensions. This means, that additional dimensions do not add additional structure to the data. Therefore it can be inferred that the clusters are artifacts from the first dimensions. Diffusion maps shows stable clusters until a 15 dimensional space. The amount of stable clusters increases with the dimensionality of the embedding, so it can be concluded that additional dimensions contribute to the structure. Because one can infer from Figure 6.8 (a) and (b), that a minimum amount of $15-20$ dimensions is needed, only Diffusion maps can identify structure that consists of more than 4 groups.

In conclusion, Diffusion maps can identify more structure in the data than *MDS* because of a more stable clustering and the contribution of additional dimensions to the structure. A 15 dimensional embedding allows a mapping of local similarity measures according to the ground truth as well as an investigation of the structure of the dataset.

6.4 Evaluation of similarity visualization

For the visualization of similar brain areas, 9 annotations of the *EXT* neuron of the **[D2-EXT-VIS]** data are available. The sets consist of 6-12 images with at least two *VT-lines*. The annotations are given as selected regions within the *mushroom body* which are similar for each subset. The evaluation is based on the overlap of the annotation and the heatmaps generated by the method of Chapter 5. They are visually compared if the annotations correspond to the similarity visualization.

Research Question

Is it possible to identify local structures that exhibit strong grouping behavior? Do these structures correspond to knowledge about genetic characteristics of the individuals? In particular, can a structure that exhibits specific differences in the population be identified?

Validation method

The method is validated by either using Diffusion maps for the embedding or Multidimensional Scaling, and compare them to the ground truth. The visualization is based on a heatmap of the average distance of the mapped data, which was introduced in Section 5.3. Therefore the brain is split into 2200 windows according to Chapter 5. The amount of windows is chosen as a trade of between performance and accuracy. To set the distances in relation to the entire **[D2]** dataset, also a U-test is performed on the data and also visualized in a heatmap (according to Section 5.3).

The mapping is computed for the **[D2]** while the validation is performed on the 9 subjects of the **[D2-EXT-VIS]** dataset. To improve the readability, only a selection of 3 sets (**[SVGT2]**,**[SVGT3]** and **[SVGT9]**) in this experiment.

The performance of the similarity visualization is tested by a qualitative comparison of the automatic identified similarity regions by Algorithm 5.2 and the a priori annotation of the expected similar morphology/neuron annotated by an expert.

Results and Discussion

Figure 6.9 shows the similarity heatmap of the diffusion map based method in the left column (a,d,g), the annotated *Maximum Intensity Projections* (MIP) for visual comparison in the middle (b,e,h) and the *MDS* based method in the right column (c,f,i). The MIPS include only slices which contain the annotated, similar regions (yellow). To compare them to the ground truth, the heatmaps are also limited to the slices of the ground truth. The entire results of this experiments can be found in Appendix B.

In the first row of Figure 6.9, the results for **[SVGT2]** can be seen. The ground truth specifies the *heel of the mushroom body* as similar region. The visualization of the Diffusion map based method (a) has dark blue areas which overlap with the yellow annotations. As one can see, also a big triangular area is blue-colored. This is not in contradiction with the ground truth because it is not available for this region too. The *MDS* method indicates the annotated region to, but with less contrast and not as smooth compared to (a).

To obtain statistical reliability for Diffusion maps, also a one-sided U-Test was computed. The blue areas are significant (p-value≤ 0.1) similar for the set in contrast to the rest of the entire dataset. Figure 6.10 leads to the conclusion, that the annotated areas are significant different to the other brains.

The next test case, **[SVGT3]** consists of two round structures on the edges of the cut. Both of them can be seen in the Diffusion maps visualization in Figure 6.9 (d). *MDS* performs worse, only a part of the left structure is considered to be similar. The U-Test heatmap in Figure 6.10 proves the similarity visualization regarding to the annotation.

[SVGT9] can be detected via Diffusion maps and *MDS*, but in different ways. While Diffusion maps can detect two lobes as similar and a separated dissimilar region between them, *MDS* merges them two one similar region. Also the U-Test identifies the region between the lobes as different to the population.

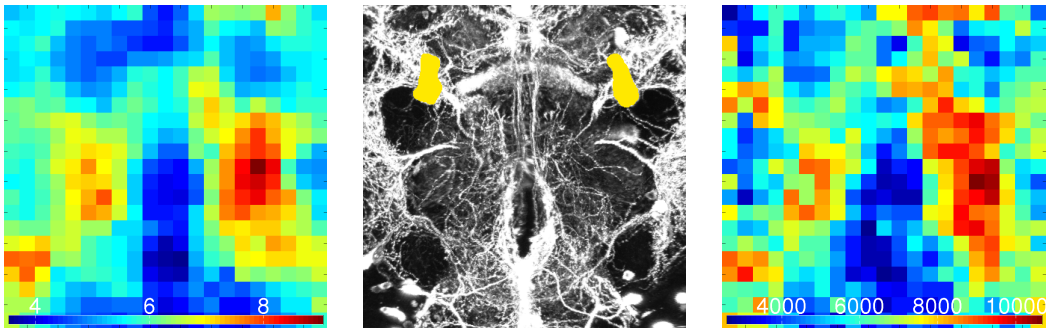
In conclusion, both methods are able to visualize similarities of the testsets but *MDS* performs worse compared to Diffusion maps. For **[SVGT2]**, the similarity was detected stronger for Diffusion maps. *MDS* was only able to identify the left annotation on **[SVGT3]**. On testset **[SVGT9]**, both methods detected the ground truth, but *MDS* merged them to one similar region.

6.5 Evaluation of embedding-based retrieval

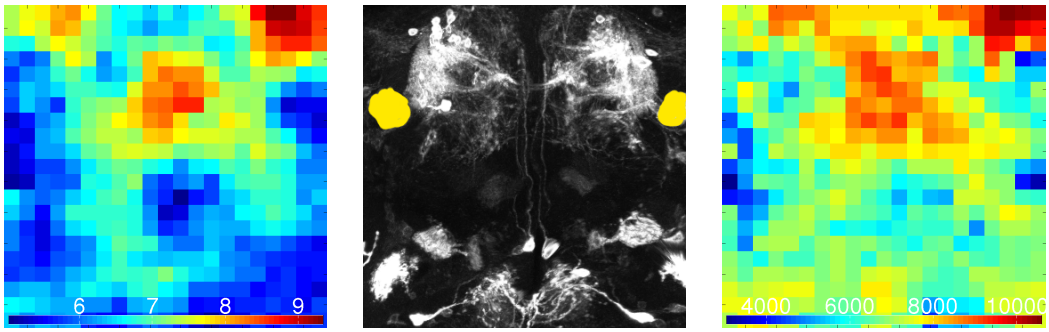
This experiment is about performing an retrieval not based on the distance function, but on the mapped data of **[D1]**. For the the mapping, a ground truth of similar structures is necessary to evaluate if they develop a clustering. Since a manual annotation of similar substructures is an extensive task, only the small set **[D1]** with one annotated structure (*pIP10* neuron with the substructures L_pIP10, R_pIP10, arborization A,B and C) is available. Like in Section 6.2 these structures are used as query patterns. Therefore brains which also contains *pIP10* are expected to be in the top-ranked retrieval results.

Research Question

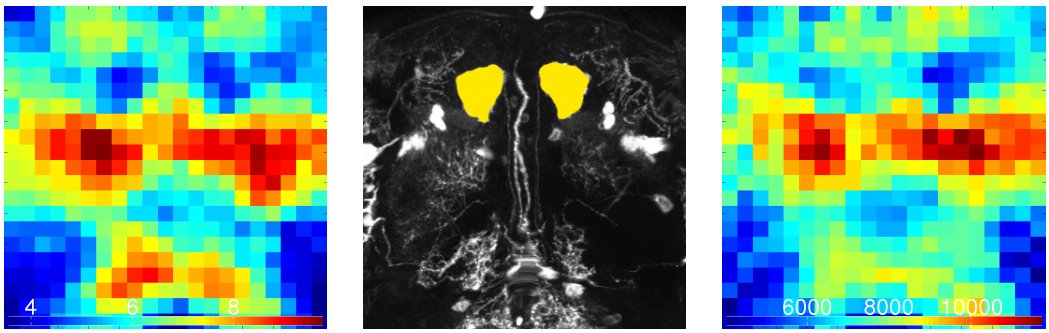
Can embedded data or clustering used for content based image retrieval? Is it better than the distance function based retrieval?



(a) SVGT2: Diffusion maps (b) SVGT2: MIP (c) SVGT2: MDS



(d) SVGT3: Diffusion maps (e) SVGT3: MIP (f) SVGT3: MDS

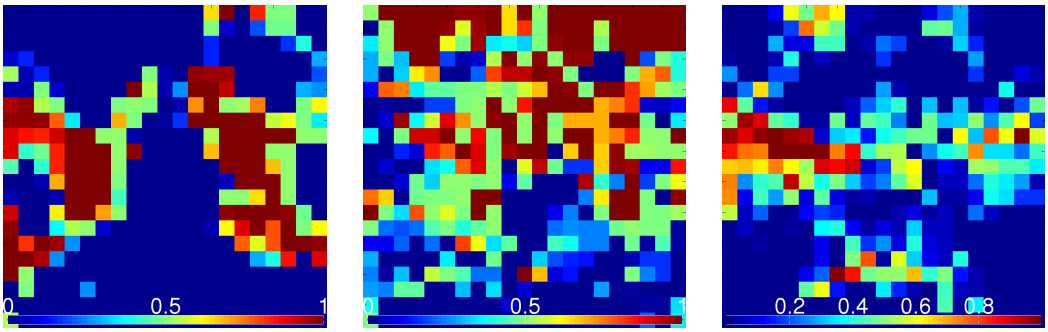


(g) SVGT9: Diffusion maps (h) SVGT9: MIP (i) SVGT9: MDS

Figure 6.9: Similarity visualization via Diffusion maps and Multidimensional Scaling: Mean fitness of every window of the optimized solutions in the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity).

Validation method

Retrievals based on the distance on the mapping and *hierarchical distance* are performed according to Algorithm 4.2. To compare the results to the distance function based retrieval of Section 6.2, the distance matrix is calculated for the areas defined by the segmentations L_pIP10,



(a) SVGT2 (b) SVGT3 (c) SVGT9

Figure 6.10: Similarity visualization via Diffusion maps: p-values of every window of the optimized solutions in the layer (z-axis) of the mushroom body. Blue means low p-value (high similarity), red high-pvalue (low similarity).

R_pIP10, arborization A,B and C for the dataset [D1]. The resulting classification error is then used for validation.

Results and Discussion

Table 6.2 shows the results of the retrieval. In the first column (*distance matrix*), the retrieval was computed by choosing the query-row (the row that corresponds to the query brain) of the distance matrix. As one can see, the classification error is for all test cases above 50%. This is caused by the symmetrization of the distance matrix (see Equation 4.2), which is necessary for the diffusion map algorithm. Without this step, the row of the distance matrix is analog to the distance function. The results for using the row of the not-symmetrized matrix is shown in the second column (*distance function*).

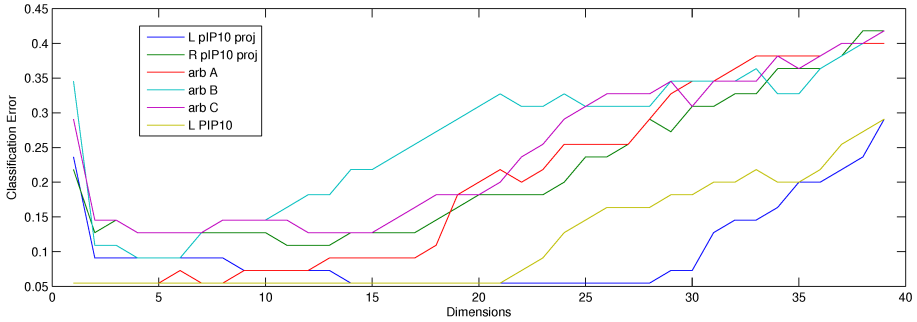
Column *distance on mapping* contains the best classification error for all possible dimensions. The relation between the dimensions of the underlying embedding and the classification error can be inferred from Figure 6.11 (a) were the error is minimized for low dimensions between 3 and 10. This concludes, that the mapping can represent the true relationship between the *pIP10* and the *not pIP10* group of **[D1]** depending on the dimension. The last column (*hierarchical distance*) includes the classification error of the hierarchical clustering based approach. While the mean classification error for the distance function is 0.08%, both mapping based methods have mean classification error of 0.09%. The mapping based distance minimizes the classification error for all cases between 4 and 5 dimensions, the error increases with the amount of dimensions. The clustering approach also increases the classification error at higher dimensions, the minimum is between 5 and 13 dimensions.

In conclusion, one can say that the retrieval works on the embedded data (mean classification error 9%) for distances (mean classification error 9%) and for the clustering, but it does not

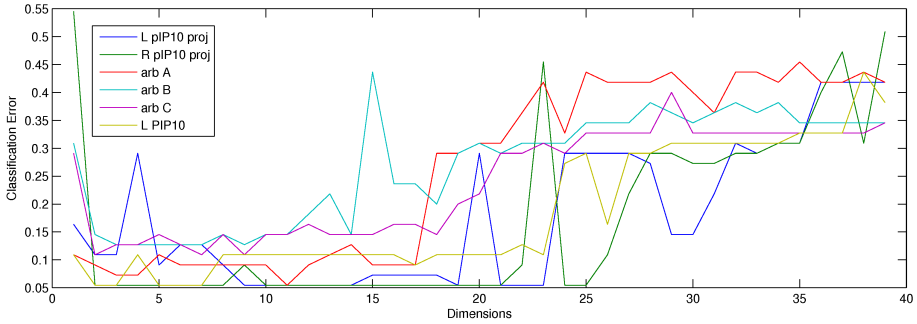
	distance matrix	distance function	distance on mapping	hierarchical distance
proj. L_pIP10	0.51	0.09	0.09	0.10
proj. R_pIP10	0.56	0.03	0.12	0.05
arbA	0.54	0.07	0.05	0.07
arbB	0.51	0.16	0.09	0.12
arbC	0.52	0.10	0.12	0.12
L_pIP10	0.50	0.03	0.05	0.05

Table 6.2: Classification error of the classification by the distance function, the distance matrix, the distance on the mapping and the hierarchical distance (for minimal mean classification error)

improve the results compared to the distance function (mean classification error 8%).



(a)



(b)

Figure 6.11: Relation between dimensions of the underlying embedding and the error of the distance based classification (a) and the hierarchical clustering based classification (b)

6.6 Evaluation exploratory analysis of data

In a final experiment this embedding approach is applied for an exploratory analysis of the **[D2-DPM]** and the **[D2-EXT]** test data. In the focus of interest is the structure of the data, more precisely groups of brains with different mutations. It is investigated, if clusters in the data reflect true relations of *VT-lines* for the test cases.

Except for the *VT-line* labels, no a priori information is known about the structure of the set, therefore this experiment represents a completely exploratory analysis of the data. Similar to the experiment of Section 6.3, the embedding is generated for the **[D2]** dataset, but but only the subsets **[D2-DPM]** and **[D2-EXT]** are analyzed. The maps are quantitatively investigated for clusters/groups by k-means clustering. The smaller datasets enables a the qualitative evaluation by visual verification of the clusters. This represents an exploratory investigation of the population given by **[D2]**.

Research Question

Is it possible to find meaningful clusters in the data for which no a–priori information is given?

Validation method

The exploratory analysis consists on two parts: A cluster analysis and a 2D plot of the dataset. The first part is a k-means clustering which depends the amount of classes k and the dimensionality of the embedding. They can be estimated by using a combination of *NNM*, *Inter/Intra VT-line distances* and *Cluster Stability Index* for the subset of interesst. The methods are explained in Section 4.2. At First, the *NNMs* and *Inter/Intra VT-line distances* are computed for the data set. They give an idea about the minimal amount of dimensions needed. The *CSI* is applied for different k and dimensions (analog to Section 6.3). The parameters can then be chosen by using a k/dimensionality combination which maximizes the *NNM* and minimizes the *CSI* for dimension with significant differences between the *Inter VT-line distances* and the *Intra VT-line distances*. The data is then visualized by the usage of *MDS* on the mapped data and plotting the first 2 two embedding dimensions in a 2D plot. This method is described in Section resulting split of the data is interpreted by an expert regarding to structure and clustering.

	2D	3D	4D	5D	10D	15D	20D	total
[D2-EXT] Mushroom Body	5	9	11	14	17	19	20	21
[D2-EXT] Central complex	4	6	13	16	18	20	20	21
[D2-DPM] Mushroom body	3	4	4	3	4	4	5	7

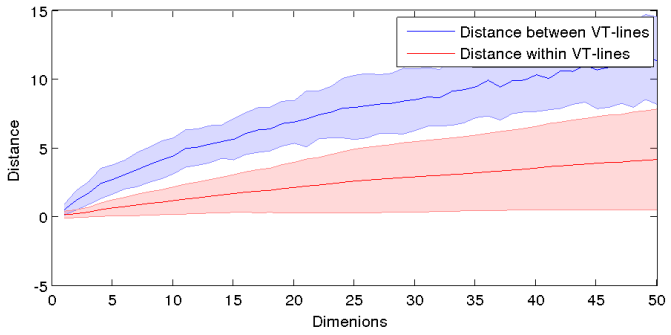
Table 6.3: Amount of nearest neighbor mutations for the 3 test cases at different dimensionality (maximum amount of *NNM* for $t = 1, 2, 3$

Results and Discussion

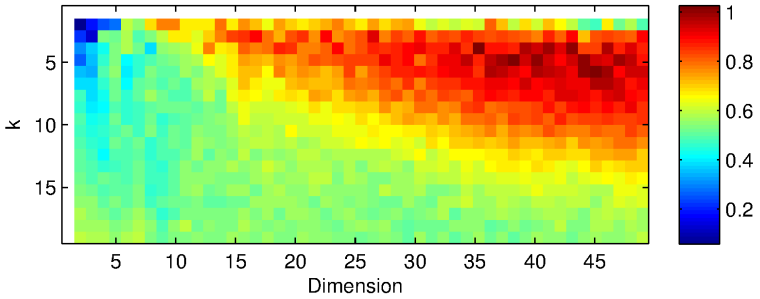
Three test cases are investigated:

1. **[D2-EXT]** in the *Mushroom body* area For the **[D2-EXT]** data set, the dimensionality of the embedding and the amount of clusters are estimated by calculating the amount of *NNMs*, the *Inter VT-line distances*, the *Intra VT-line distances* and the *CSI* for different parameter variations. Table 6.3 infers, that at least $5 - 10$ dimensions are needed to cover the most *NNMs* (14-17 of a total amount would be 21). A significant difference in the distances between *VT-lines* and within *VT-lines* can be already achieved for 3 dimensions according to 6.12 (a). In Figure 6.12 (b), the stablest clustering for dimensions ≥ 5 is at $k = 4$ for 8 dimensions.

   The clustering can be seen in Figure 6.13. Because it is a 2D plot of a 8 dimensional clustering, some clusters are overlapping in the plot (like cluster 1 and 4. Obviously, the relations within this cluster lie in a higher dimension. 19 out of 21 *VT-liness* are not split between two or more clusters, which can be interpreted as a good sign that the clustering represents the true structure. To get more evidence about the relation between *VT-lines*, one brain per *VT-lines* is displayed for every cluster in Figure 6.14. As one can see, the cluster do not represent a special structure. The reason for this is, that the data set is inhomogeneous: The *Mushroom body* area is large enough (in relation to the total brain, see Figure 6.2) to contain more than one neuronal structure but there is only one distance function. So the neuronal structures compete. For example, if there is a high similarity for one neuronal structure in two *VT-lines* (low distance), but two structures which are completely different (high distance), the two structures would overrule the similar one. There are two possible solutions for this problem. On the one hand, one can reduce the size of the area which is compared and therefore the amount of competing neuronal structures. This is done in the **[D2-EXT]** - *Central complex* test case. On the other hand, one can choose a set, which is known to be different in only one or two characteristics. The **[D2-DPM]** - *Mushroom body* test case is an example for that.



(a) Distribution (given by the mean and the standard deviation) of the *Intra VT-line distance* (red) and the *Inter VT-line distance*



(b) Cluster Stability Index vs dimension of the embedding

Figure 6.12: Performance of the mapping and the clustering of the *EXT* neuron (in the *Mushroom body* area) depending on the dimension of the embedding

2. **[D2-EXT]** in the *Central complex* area Due to the issues of the first test case, the experiment on the **[D2-EXT]** was also computed for a smaller area, the *Central complex*. As it can be seen in Figure 6.2, the *Central complex* overlaps with the *Mushroom body*. Because the area is smaller, less neuronal structures, and therefore a better clustering is expected. As for the other test cases, the parameter is done by calculating the amount of *NNMs* and the *CSI*. A minimum of $5 - 10$ dimensions can be concluded from Table 6.3 and Figure 6.15 (a). For this range, the best amount of clusters according to the cluster stability (Figure 6.15 (b)) can be identified as $k = 5$ for 6 dimensions.

   Figure 6.16 shows the first two embedding dimensions derived by *MDS* of the mapped data. As one can see, 5 clusters can be visualized also in two dimension, although the clustering was performed on 6 dimensional data. To investigate the data one brain per *VT-lines* is displayed for every cluster in Figure 6.17 in a similar way as for the first testcase. *Cluster 1* shows to vertical *projections* in the middle of the complex. *Cluster 2* shows a round structure in the center which is particular strong in the third and the fifth *VT-line*. The third cluster contains only one *VT-line*. At a closer look, the lower two horizontal structures can be also found in the brains of *Cluster 4*. In Figure 6.16, *Cluster 3* is also

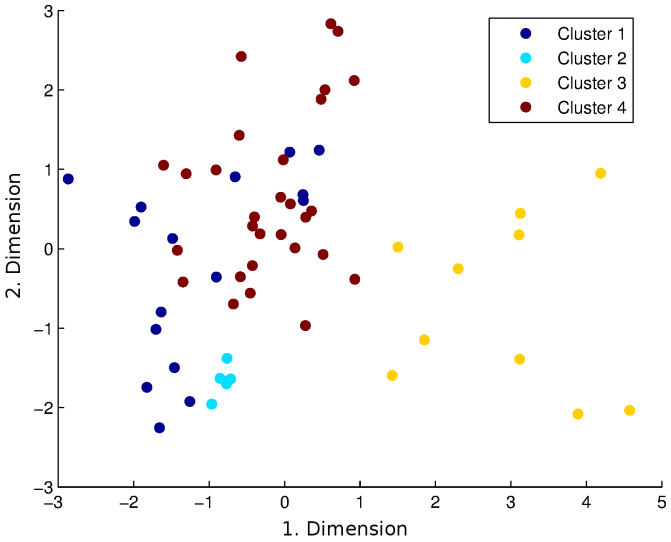


Figure 6.13: First 2 embedding dimensions of the **[D2-EXT]** (*Mushroom body*) set

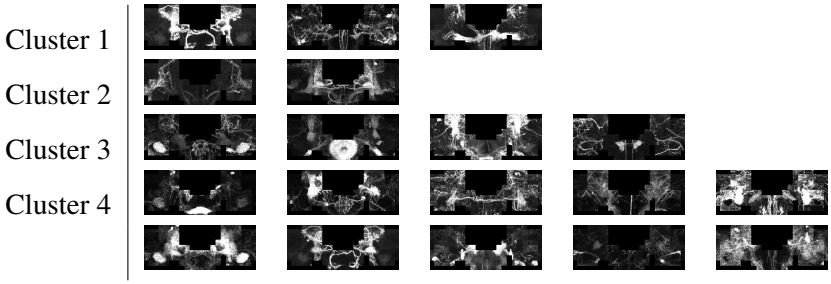
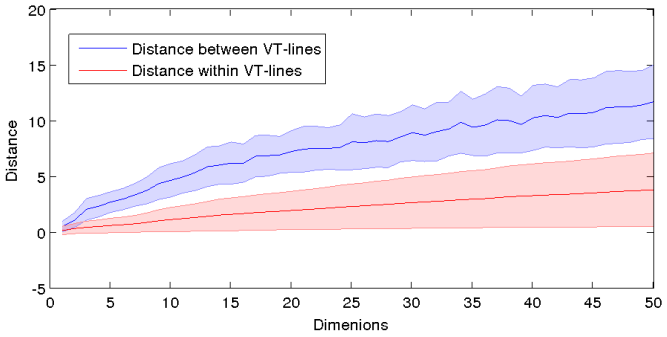
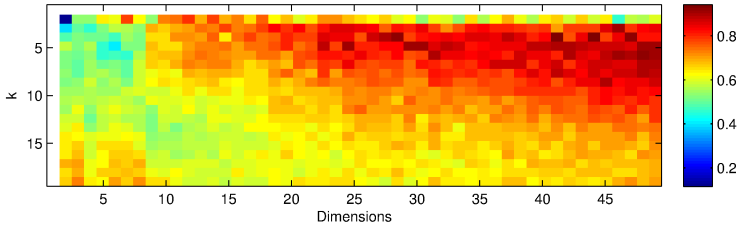


Figure 6.14: Clustering of the **[D2-EXT]** dataset (*Mushroom body* area). For every *VT-line* which is not split between clusters, one representative brain is shown (so every image represents one *VT-line*)



(a) Distribution (given by the mean and the standard deviation) of the *Intra VT-line distance* (red) and the *Inter VT-line distance*



(b) Cluster Stability Index vs dimension of the embedding

Figure 6.15: Performance of the mapping of the *EXT* neuron (in the *Central complex* area) depending on the dimension of the embedding

mapped close to *Cluster 4*. The reason for the separate cluster are the upper two round structures, which can not be found in *Cluster 4*. The special characteristic of *Cluster 4* is the semi-circled *projection* in the middle, and the two horizontal structures which are also in *Cluster 3*. *Cluster 5* represents the rest but also some similarities: The second, third and fourth *VT-lines* have two vertical *projections*, which explains the mapping close to *Cluster 1*. The first, second and fourth show two round structures in the middle.

In conclusion, a smaller area with less characteristics (compared to the first test case) is better suited for a clustering on the mapping.

3. **[D2-DPM]** in the *Mushroom body* area

   For the *DPM* neuron, 4 out of 7 *NNM* can be found for at least a 3 dimensional space according to Table 6.3. Figure 6.18 (a) shows, that the only significant difference between the *Inter VT-line Distances* and the *Intra VT-line Distance* can be found between 11 and 18 dimensions. For this range, two stable groups are identified in Figure 6.18(b) with the stablest clustering for 11 dimensions. Figure 6.19 reveals the structure of the data in a 2D plot. Three separated groups can be identified visually although the clustering defines only two groups in a higher dimensional space. To investigate the plot, one brain per *VT-line* is visualized according to their cluster membership in Figure 6.20. All *VT-lines* except one are not split by the clustering. In principle, *Cluster 1* shows the *Mushroom*

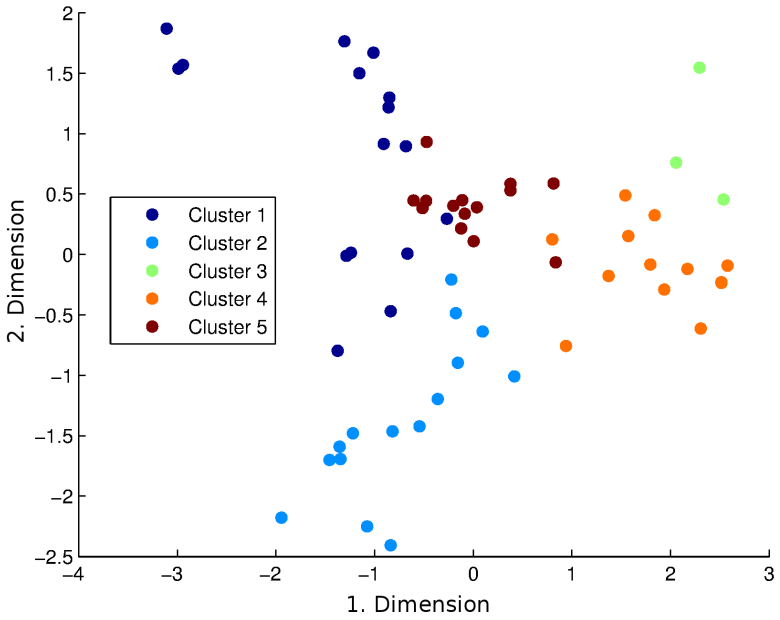


Figure 6.16: First 2 embedding dimensions of the **[D2-EXT]** (*Central complex*) set

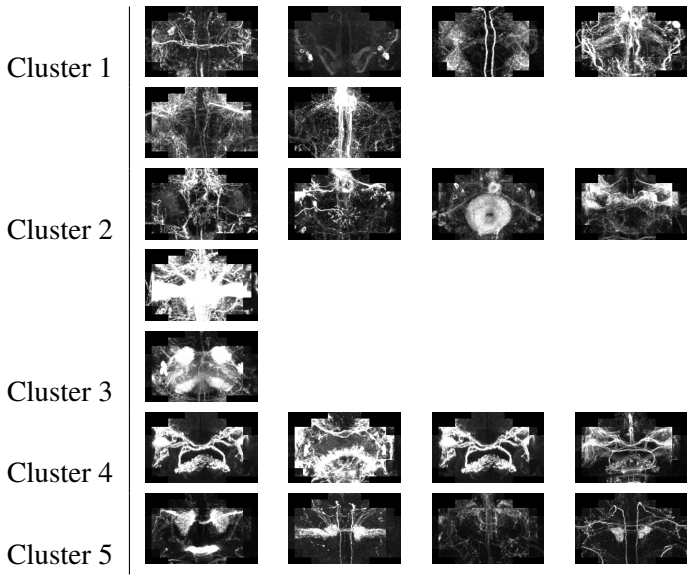
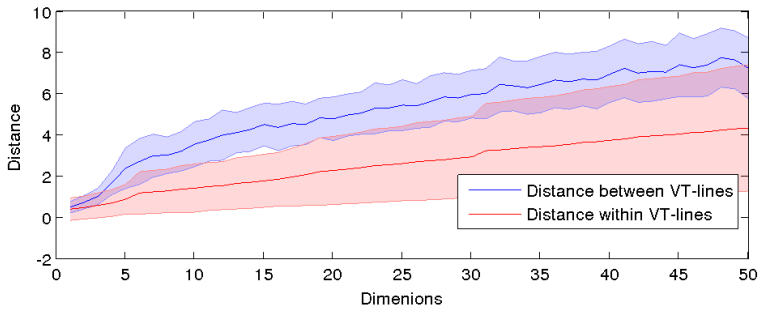
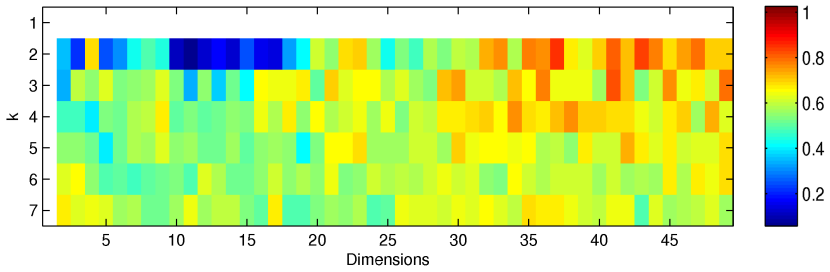


Figure 6.17: Clustering of the **[D2-EXT]** dataset (*Central Complex* area) . For every *VT-line*, one representative brain is shown (so every image represents one *VT-line*



(a) Distribution (given by the mean and the standard deviation) of the *Intra VT-line distance* (red) and the *Inter VT-line distance*



(b) Cluster Stability Index vs dimensionality of the embedding

Figure 6.18: Performance of the mapping of the *DPM* neuron depending on the dimension of the embedding

body with two separated parts. The first 3 *VT-lines* corresponds to the points above the blue line in Figure 6.19, while the last two are below the blue line. The difference between these sub groups are the two vertical neurons between the two parts. *Cluster 2* consists also of two sub groups. The first two *VT-lines* show the two parts of the *Mushroom body* merged together (above blue line), the third one is not merged and below the line. All brains of *Cluster 2* have a *projection* on the left and right region outside the body. These are common in *Cluster 2*, but not in *Cluster 1* and are therefore the separating feature. In conclusion, the clustering is driven by those external neurons. Because there are less characteristics compared to the first testcase, the clustering resulted in separated groups although the mapping is also based on the *Mushroom body* area.

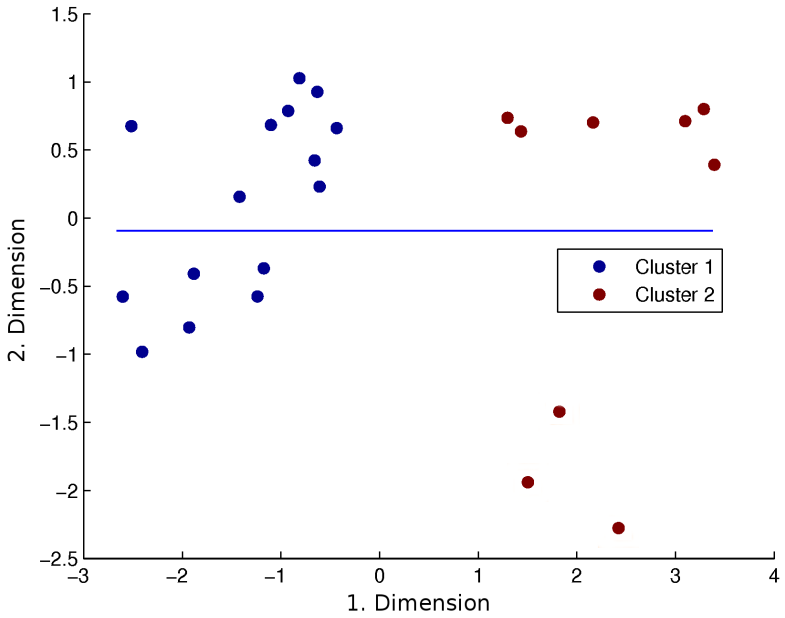


Figure 6.19: First 2 embedding dimensions of the **[D2-DPM]** (*Mushroom body*) set. The blue line is for exploratory reasons in this section.

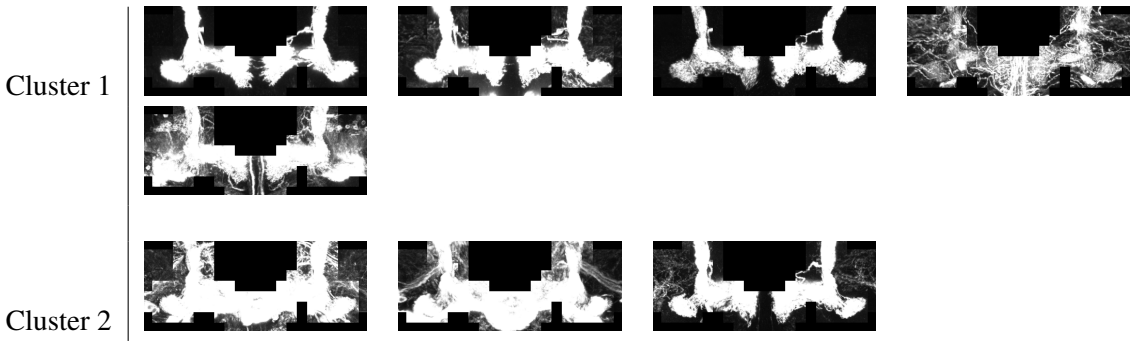


Figure 6.20: Clustering of the **[D2-DPM]** dataset. For every *VT-line*, one representative brain is shown (so every image represents one *VT-line*

6.7 Summary

This Chapter described the experiment data, experiments and their results. The experiment data was divided into 5 datasets which have different characteristics to use it for different experiments.

The similarity measure was validated by using it as a classifier between similar fly brains and not-similar fly brains. Therefore, two groups were annotated by an expert. Different methods for selection of the query pattern were used, such as manual query selection and with or without automatic segmentation within a bounding box. The results were also compared to a voxel-based method. The pre- and automatic segmented method performed best.

The validation of the mapping was done by quantitative measures in two experiments. One experiment investigated the mapping for structure on an entire fly brain population and compared Diffusion maps with *MDS*. By using Diffusion maps it was possible to identify stable clusters, whereby the amount of clusters increased with the dimensionality. In a second experiment, subpopulations were used for an exploratory analysis of the data. Groups of fly brains were found by quantitative methods such as the *CSI* and also visualized in *MIP* for visual verification.

It was also tested, if image retrieval can be performed on embedded data. The results were evaluated in a similar way as for the similarity measure based approach. In conclusion, both approaches show equal results, so the mapping does not improve the retrieval.

The similarity visualization was validated in an experiment by comparing the results with expert-annotated brains. The visualizations were computed by either a Diffusion maps or *MDS* based approaches. The results were verified by the overlap of the heatmaps to the ground truths which were given as *MIP*s with labeled, similar areas. For this experiment, both methods were able to visualize similarities, but *MDS* performed worse compared to Diffusion maps.

CHAPTER 7

Conclusion and Outlook

The aim of this thesis was to develop a method to analyze and visualize variability of drosophila brains. Therefore a similarity measure for neuronal structures was needed to set the brains in relation to each other.

A 3D structure tensor based similarity measure was developed, which was able to compensate shifts in the location and different shapes. Therefore, gradient vector flow was used which smooths the query pattern and diffuses it into the surrounding area. To work with large datasets, computational intensive parts like gradient estimation were put into a pre-processing step to allow a fast on-the-fly computation of the similarity. It has been shown, that the measure can be used for image retrieval. It performs on an annotated data set with a mean error rate of 8% (for pre-segmented data) better than a voxel based method (26.3%). The validation showed, that the performance on tubular structures like projections is better than for arborizations. In addition, the similarity measure was adapted to work within various rectangular windows to use it for the visualization of variations in small populations.

Another challenge was to develop a mapping that represents the similarity of local brain areas. Therefore a distance matrix for the area of interest was generated and mapped by a dimensionality reduction method. It has been proven, that the non-linear method Diffusion maps is able to find more structure in the data set than the linear Multidimensional Scaling. The methods were evaluated by quantitative measures like the amount of Nearest Neighbor Mutations and the Cluster Stability Index, which revealed stable structures in data for low dimensional spaces. Also examples for the clusterings were shown, but they have only limited evidence for biology without annotations.

The third contribution of this thesis was the visualization of similar regions for a set of brains with different mutations. Brains were split into corresponding windows to compute their similarity for the area within. For the reason, that neuronal structures can cover more regions, the windows were merged by multi-modal optimization. The mean similarity for a window, which

was defined by the mean distance of the brains mapped by Diffusion maps or Multidimensional Scaling, was then visualized by a heat map. It has been shown, that the detected similarities overlap with expert annotations, especially for the Diffusion maps based mapping. Therefore it can be suggested that biological relevant structures can be found with this method.

The experiments which were performed in this thesis showed, that the similarity measure can be used for image retrieval and for the visualization of similar regions of a set of fly brains. These methods were proofed by expert annotated data. The mapping shows clusters which can be verified by quantitative methods. In addition, it was qualitative validated for the reason that it is inherent in the similarity visualization.

For future research, it is planned to use the mapping on areas of biological relevance. Biologists can use the proposed method to investigate data were they expect structures. This would generate biological evidence of the mapping method in addition to the quantitative measures. Another focus of interest is to use the methods on different scales. So far, the similarity visualization was used with 2200 windows as a trade of between the visualization of the entire brain and the computational complexity due to the multi-modal optimization. By reducing the area of interest to specific areas like the Mushroom body, the same amount of windows can be used to detect smaller structures.

APPENDIX A

Parameter Optimization

To optimize the results of the algorithms, which are used in this thesis, several parameters can be varied. In this section, the parameters are divided into two groups: Parameter which are responsible for the outcome of the similarity calculation, and parameters which influence the genetic algorithm for multi-modal optimization.

A.1 Calculation of Similarity

The optimization is done by trying possible (and likely) values on the test-set. The parameters which lead to a minimum of the classification error are the optimum. The following parameters were optimized for downscaled images by the factor 4 (due to faster computation):

1. Parameter for Gaussian smoothing:

   Table A.1 shows the classification error of different smooth ranges (size of Gaussian kernel) and σ. As one can see, the best results can be achieved for $\sigma = 2$ for a kernel size of 5.

	range=3	range=5	range=7	range=9
sigma = 0.1	0.1061	0.1030	0.1030	0.1030
sigma = 0.5	0.0939	0.0939	0.0939	0.0939
sigma = 1	0.0909	0.0939	0.0939	0.0939
sigma = 2	0.0909	0.0848	0.0909	0.1030

Table A.1: Mean classification error of different kernels on the classification error

2. Window of the partial derivatives for tensor estimation: Two window sizes were tested. For a 3x3 window, the mean classification error for the test-runs is 0.084 while the error for a 5x5 window was 0.087. So the difference is negligible. Because the computation of partial derivatives of a 3x3 window is faster than for a 5x5, a 3x3 window should be used.

3. γ for tensor normalisation: The γ values 0.01, 0.001, 0.0001 and 0.00001 were tried. A γ value below 0.001 and above 0.0001 leads to the best results.

	L_pIP10 proj	R_pIP10 proj	arb A	arb B	arb C	L_pIP10
$\gamma = 0.01$	0.09	0.07	0.12	0.16	0.14	0.05
$\gamma = 0.001$	0.09	0.03	0.07	0.16	0.10	0.05
$\gamma = 0.0001$	0.09	0.03	0.07	0.16	0.10	0.05
$\gamma = 0.00001$	0.11	0.03	0.07	0.16	0.10	0.05

Table A.2: Classification error of test-runs at different γ values

4. Amount of iterations and μ for gradient vector flow: The amount of iterations is a crucial point of the gradient vector flow because the computation time increases. Figure A.1 shows the effect of the iteration amount on the classification for 5 different μ. For more then one configuration, the mean classification error is minimal at 0.0848. For the reason of computation time, the optimal iteration amount is set to 20. $\mu = 0.04$ or but also $\mu = 0.05$ would be possible.

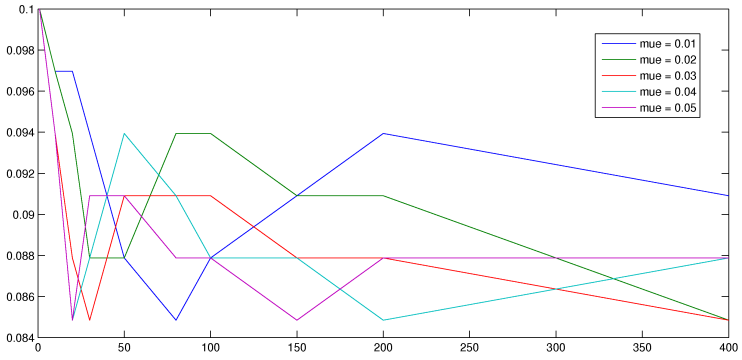


Figure A.1: Effect of μ and the amount of iterations on the mean of the classification error for all 6 testsets

A.2 Multi-modal Optimization

The parameters of the multi-modal optimization are investigated by using **[SVGT2]** as example. In Figure A.2 the amount of window changes of the similiarity visualization after e iterations (epoches) for different population sizes are shown. As one can see, the changes in the windows (total window amount is 2200) start to converge for populations $\leq$ 2000 after 150 iterations, while the population with 3000 individuals converges earlier. The results for the different populations can be seen in A.3. The dark brown windows of represent windows that are not present in the population. This is caused by a too small initial population or too many iterations (depending on the population size). Their amount increases with the iterations because the fitness is optimized. After an infinite amount of muations, all windows except the windows which maximum fitness would be removed from the population. For all populations, the contrast between similar and not similar regions increases with the amount of mutations. Similar regions become more blue, while not similar regions become more red. In conclusion, a population size $\geq$ 2000 with an iteration amount between 50 and 100 should be chosen to achieve the best results with all windows present in the population.

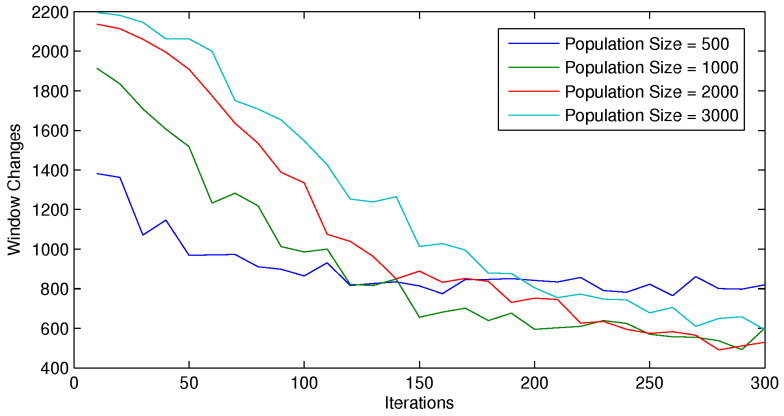


Figure A.2: The amount of window changes after e iterations (epochs) for different population sizes

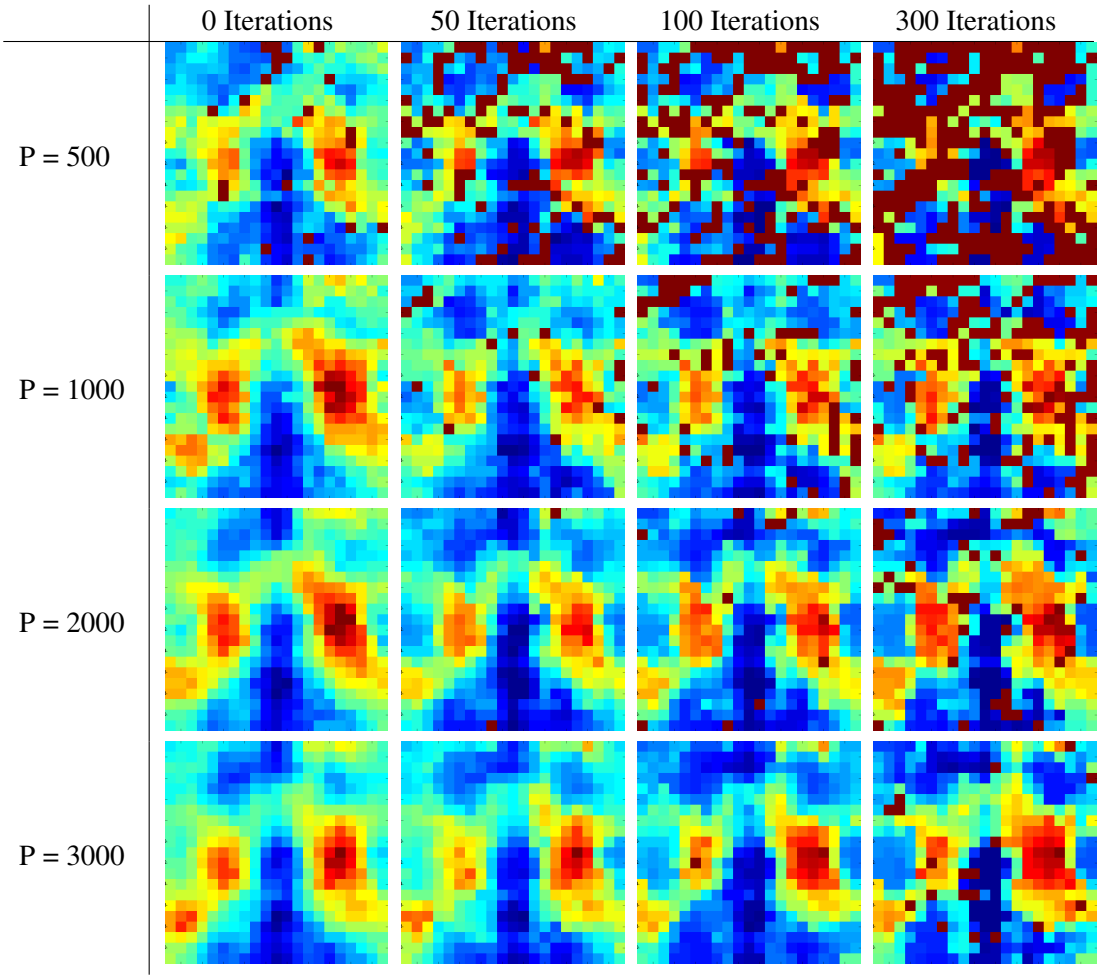
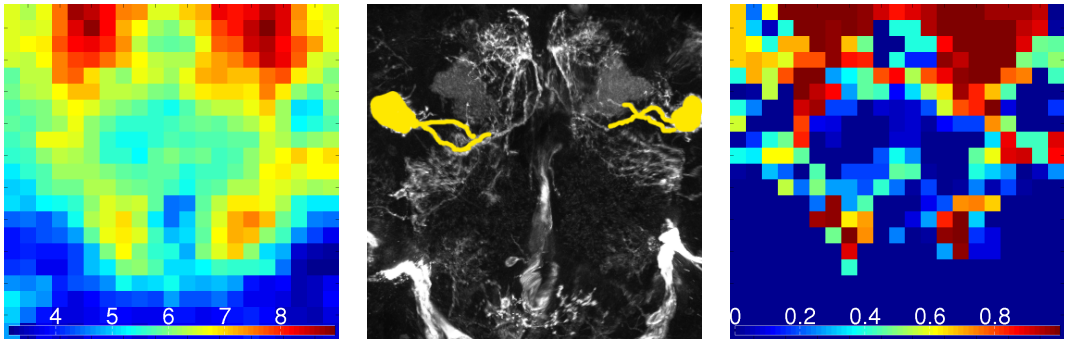


Figure A.3: Ouptut of multimodal optimization for **[SVGT2]** for different population sizes and iterations

APPENDIX B

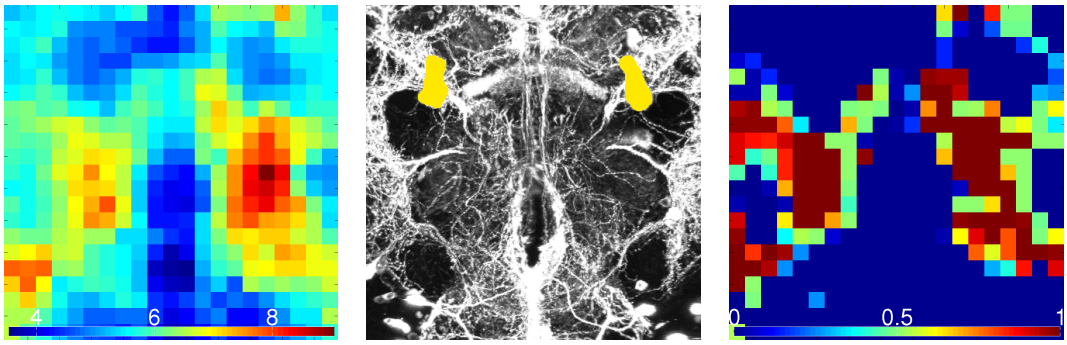
Similarity Visualization Results

Figure B.1 - B.9 show the best fitness (a) and best p-value (c) of every window of the optimized solutions in the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity). The annotations can be seen in the middle (b). P-values of **[SVGT4]**, **[SVGT5]**,**[SVGT6]**, **[SVGT7]** and **[SVGT8]** are log scaled, because all windows were significant.



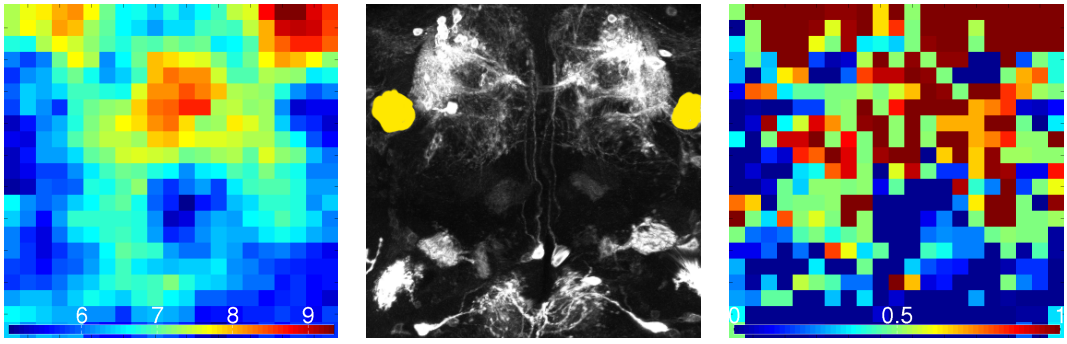
(a) (b) (c)

Figure B.1: **[SVGT1]**



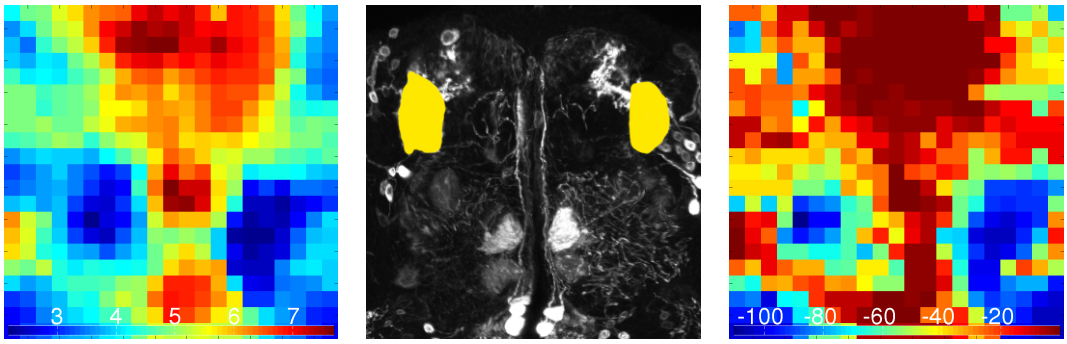
(a) (b) (c)

Figure B.2: **[SVGT2]**



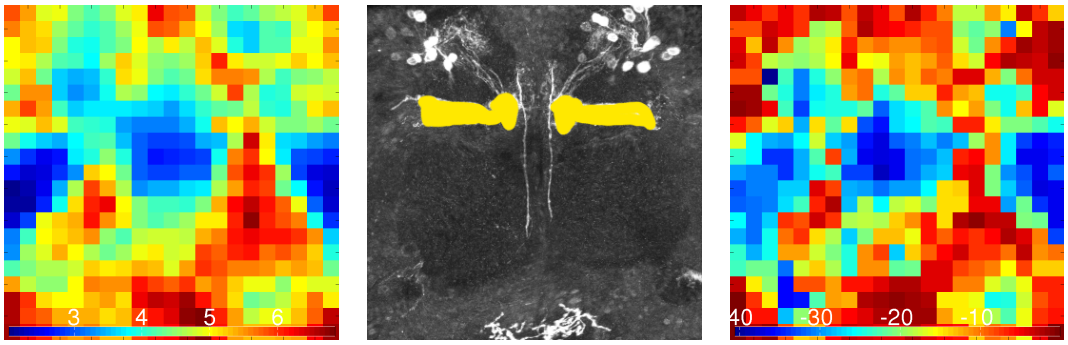
(a) (b) (c)

Figure B.3: **[SVGT3]**



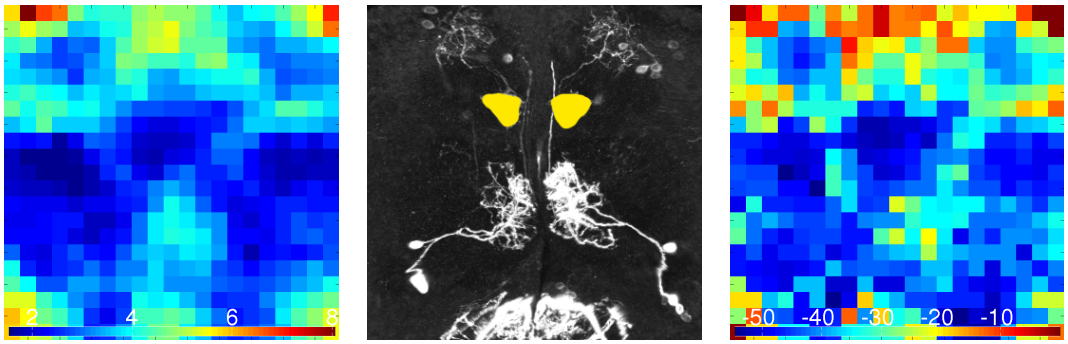
(a) (b) (c)

Figure B.4: **[SVGT4]**: The p-values of (c) are log scaled because every window was significant.



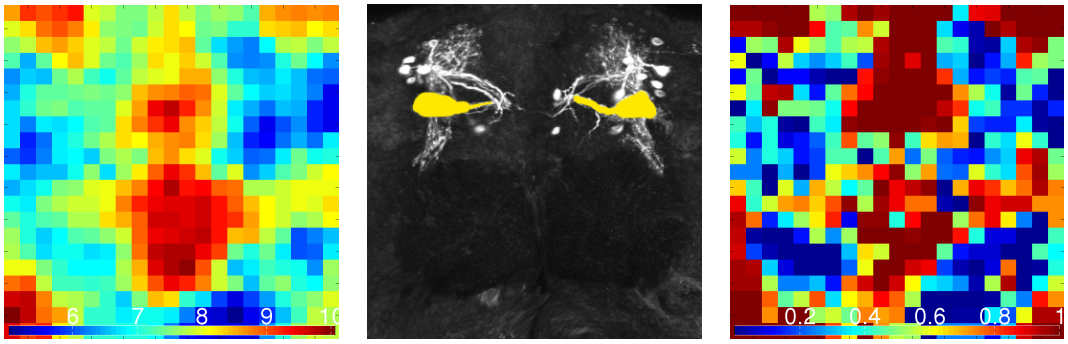
(a) (b) (c)

Figure B.5: **[SVGT5]**: The p-values of (c) are log scaled because every window was significant.



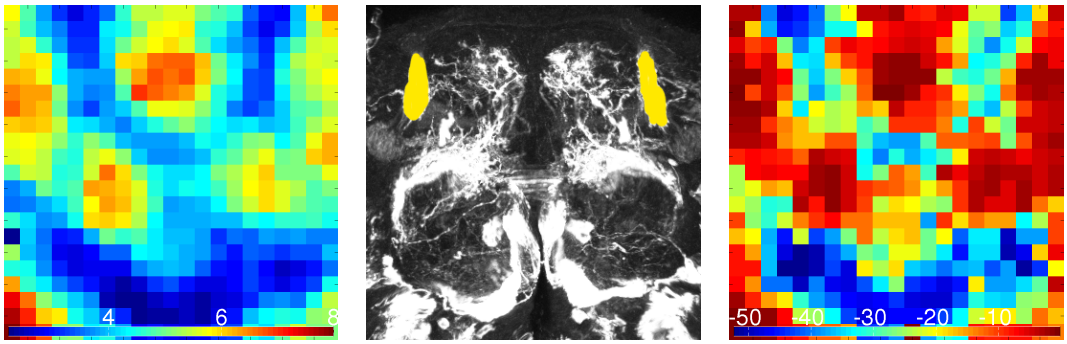
(a) (b) (c)

Figure B.6: **[SVGT6]**: The p-values of (c) are log scaled because every window was significant.



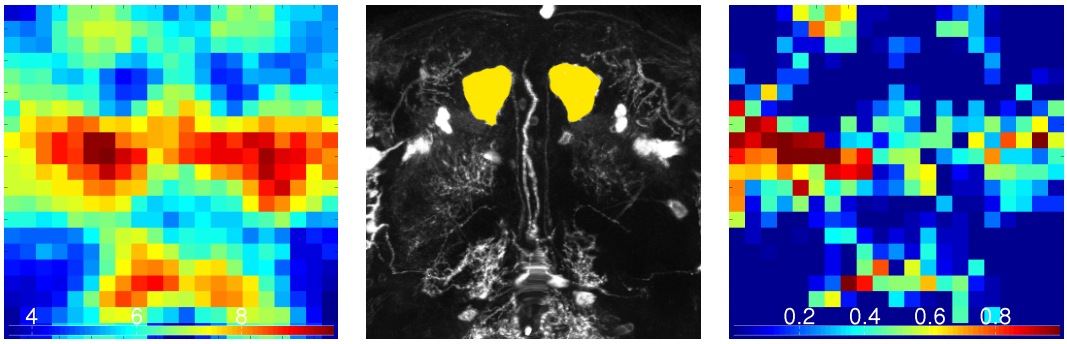
(a) (b) (c)

Figure B.7: **[SVGT7]**



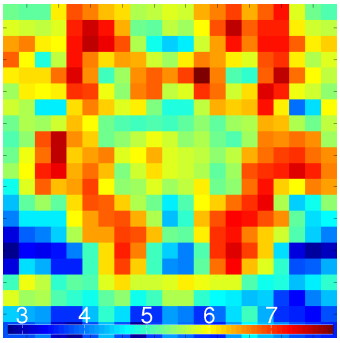
(a) (b) (c)

Figure B.8: **[SVGT8]**: The p-values of (c) are log scaled because every window was significant.

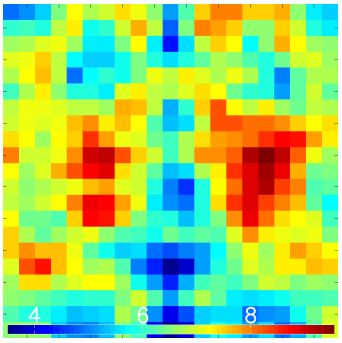


(a) (b) (c)

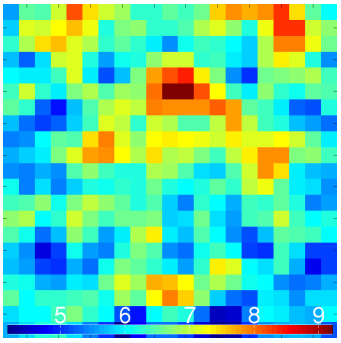
Figure B.9: **[SVGT9]**



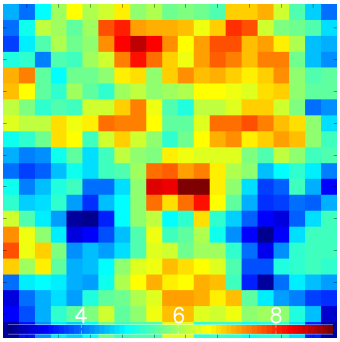
(a) SVGT1



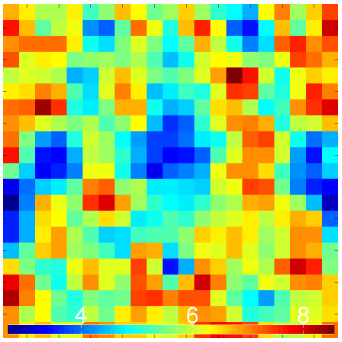
(b) SVGT2



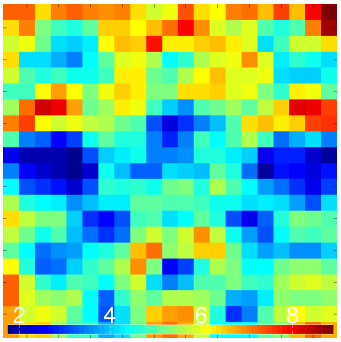
(c) SVGT3



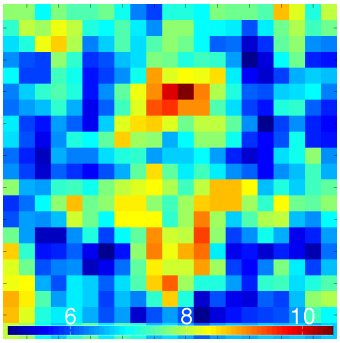
(d) SVGT4



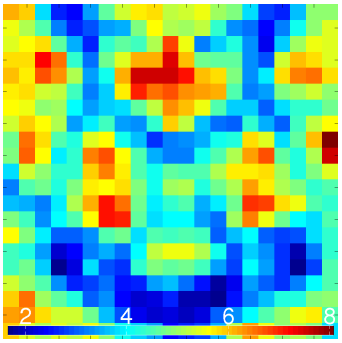
(e) SVGT5



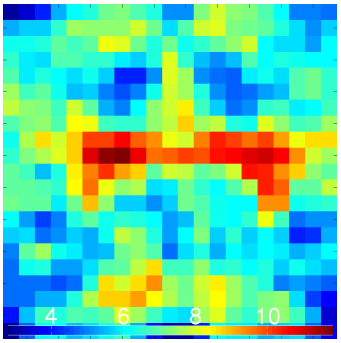
(f) SVGT6



(g) SVGT7

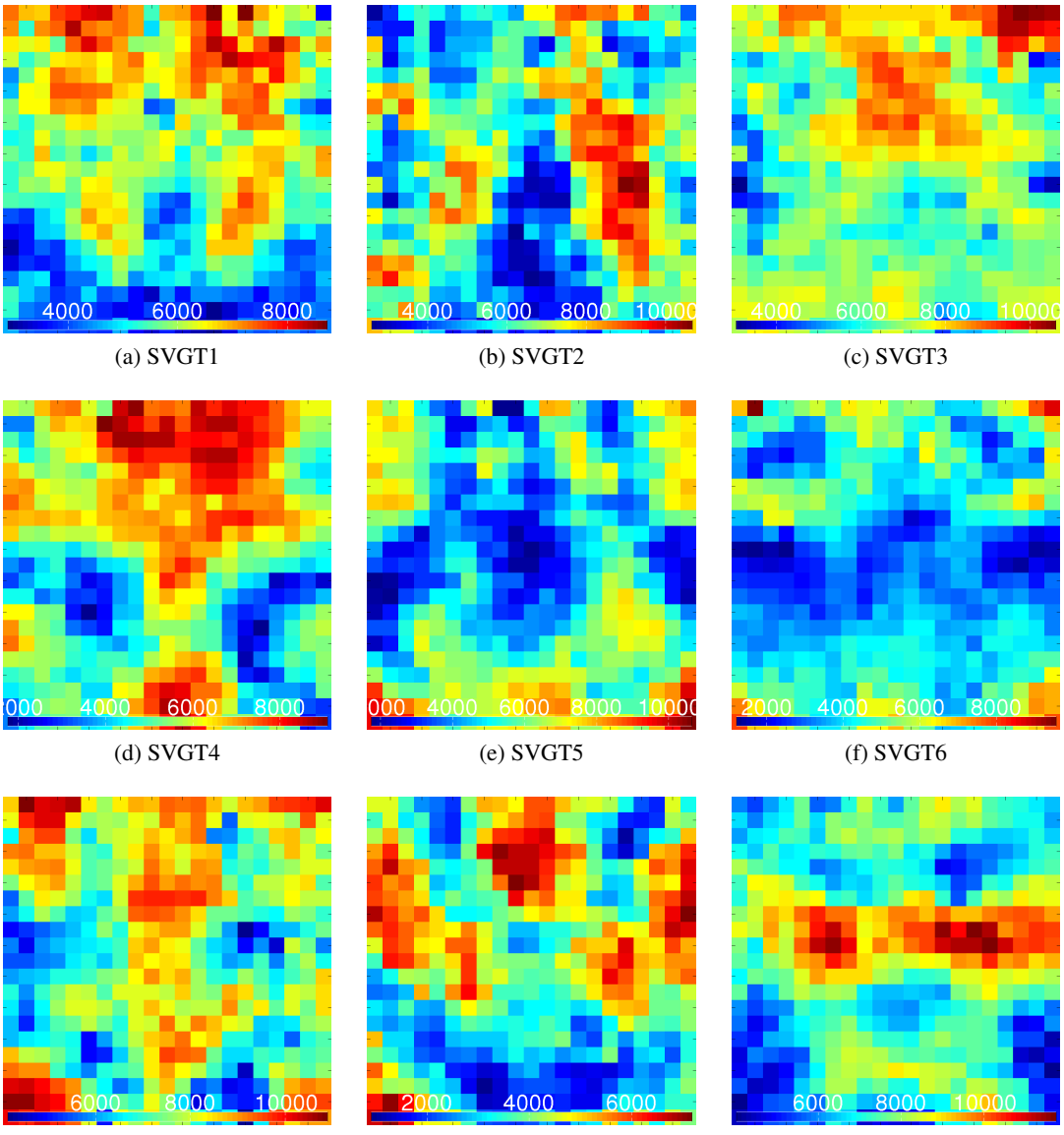


(h) SVGT8



(i) SVGT9

Figure B.10: Fitness of every single window in the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity).



(a) SVGT1 (b) SVGT2 (c) SVGT3

(d) SVGT4 (e) SVGT5 (f) SVGT6

(g) SVGT7 (h) SVGT8 (i) SVGT9

Figure B.11: Similarity visualization via Multidimensional Scaling: Best fitness of every window of the optimized solutions in the layer (z-axis) of the mushroom body. Blue means high fitness (high similarity), red low fitness (low similarity).

APPENDIX C

Glossary

Antibody: A large Y-shaped protein.

Arborization: Efferent fibers of a neuron. Branched. Technical description of *Dendrites*.

Axon: Afferent fibers of a neuron. Biological description of *projection*.

Cell body: Bulbous end of a neuron. Connected with axons (afferent) and dendrites (efferent).

Central Complex: Special brain area in the center of the brain. Can be seen in Figure 6.2 (b).

CSI: Cluster Stability Index [40]. Low *CSI* indicates high stability.

Dendrites: Efferent fibers of a neuron. Branched. Biological description of *arborization*.

DPM neuron: "dorsal paired medial" neuron. Can be found in the *Mushroom body*

Drosophila melanogaster: Fruit fly.

DTI: Diffusion Tensor Imaging

EXT neuron: "external neuron". Can be found in the *Mushroom body*

Genotype: Genetic configuration of a cell.

GVF: Gradient Vector Flow. Used for smoothing and vector spreading of a query pattern. State of the art methods are explained in Section 2.3 and the usage in this thesis in Section 3.3.

Hierarchical distance: The hierarchical distance is defined as the lowest level in a complete linkage clustering dendrogram [63] that connects two points.

Inter VT-line distance: Sum of the distances between the brains of different *VT-lines*.

Intra VT-line distance: Sum of the distances between the brains of the same *VT-lines*.

MDS: Multidimensional Scaling: Linear dimensionality reduction method.

MIP: Maximum Intensity Projection.

Mushroom body: Special brain area which is shaped like a mushroom. Can be seen in Figure 6.2 (a).

Mutation: Alteration of a gene. In the context of this thesis, such alterations are referred to as *VT-line*.

Nearest Neighbor Mutations: Fly brains with the same genetic mutation (so they are from the same *VT-Line*) are samples of the same *genotype* (like clones), so they are more similar to each other than to brains of an other mutation. A *NNM* is a mutation, all of its brains are mapped closest together (brains of the same mutation are their nearest neighbors).

Neuroblast: A cell which will develop into a neuron.

Neuropil: The major part of gray matter of the brain. Surrounds the *cell bodies*.

Niche: Similar individuals which lead to one optimum in the context of multi-modal optimization.

NNM: Abbreviation of Nearest Neighbor Mutations.

Query area: Area for which the similarity measure will be computed.

PCA: Principle Component Analysis. Linear dimensionality reduction method.

pIP10: A special neuron that is related to *Drosophila* courtship behavior [72]. Segmentations and annotations are available in Section 6.1.

Projection: Afferent fibers of a neuron. Technical description of *Axon*.

Spatial grid: Grid that enables development of distinct solutions (spatial selection).

TEP: Tree Edit Distance is the amount of nodes which need to be added/removed from one tree to create a second one [78].

Transgene: A gene or genetic material that has been transfered from one organism to another.

VT-line: Pieces (tiles) which were likely to be active in the nervous system were used to generate transgenic fly lines. Each of these transgenic fly lines contains one unique tile which can be considered as a genetic alteration. In the context of this thesis, such alterations are referred to as *mutation*.

Window grid: Divides fly brains in non-overlapping cubic windows of equal size.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews Computational Statistics*, 2(1-3):37–52, 1986.

[2] ZUSE-Institut Berlin (last accessed March 9, 2012). Amira File Format. `http://amira.zib.de/mol/usersguide/HxFileFormat_AmiraMesh.html`

[3] Julio Barrera and Carlos A Coello Coello. A Review of Particle Swarm Optimization Methods used for Multimodal Optimization. *INNOVATIONS IN SWARM INTELLIGENCE*, 248:9–37, 2009.

[4] Erhan Bas and Deniz Erdogmus. Principal curves as skeletons of tubular objects: locally characterizing the structures of axons. *Neuroinformatics*, 9(2-3):181–91, September 2011.

[5] Shai Ben-David, Ulrike Von Luxburg, and Dávid Pál. A sober look at clustering stability. In *COLT'06 Proceedings of the 19th annual conference on Learning Theory*, volume 4005, pages 5–19, 2006.

[6] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 17:6–17, January 2002.

[7] P. Bertrand and G. Bel Mufti. Loevinger's measures of rule quality for assessing cluster stability. *Computational Statistics & Data Analysis*, 50(4):992–1015, February 2006.

[8] Erich Birngruber, Georg Langs, and René Donner. matVTK - 3D Visualization for MATLAB. *The MIDAS Journal*, pages 1–8, 2009.

[9] H Brand and N Perrimon. Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. *Development (Cambridge, England)*, 118(2):401–15, June 1993.

[10] Kevin L Briggman and Winfried Denk. Towards neural circuit reconstruction with volume electron microscopy techniques. *Current Opinion in Neurobiology*, 16(5):562–570, 2006.

[11] Stefan Bruckner, Veronika Šoltészová, Meister Eduard Gröller, Jiří Hladůvka, Katja Bühler, Jai Yu, and Barry J Dickson. BrainGazer - Visual Queries for Neurobiology Research. *IEEE Transactions on Visualization and Computer Graphics*, 15:1497—-1504, 2009.

[12] Denis Chigirev and William Bialek. Optimal Manifold Representation of Data : An Information Theoretic Approach. *Advances in Neural Information Processing Systems 16*, page 161, 2004.

[13] R Coifman and S Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006.

[14] R J Collins and D R Jefferson. Selection in massively parallel genetic algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 249–256, 1991.

[15] Yuval Davidor. A Naturally Occurring Niche and Species Phenomenon: The Model and First Results. In *ICGA*, pages 257–263, 1991.

[16] D L Davies and D W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[17] Kenneth Alan De Jong. *An analysis of the behavior of a class of genetic adaptive systems, 1975*. PhD thesis, University of Michigan, 1975.

[18] J de la Porte, B M Herbst, W Hereman, and Stefanvan der Walt. An Introduction to Diffusion Maps. *Techniques*, 2008.

[19] Barry J Dickson. `http://www.imp.ac.at/research/research-groups/dickson-group/` Website of Dickson Group, 2012.

[20] Eva Dittrich. *Automatic Segmentation of Retinal Vessels and Measurement of Doppler Flow Velocity in Optical Coherence Tomography Data*. Diploma thesis, Vienna University of Technology, 2009.

[21] J C Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Cybernetics and Systems*, 3(3):32–57, 1973.

[22] Stanley Durrleman, Pierre Fillard, Xavier Pennec, Alain Trouvé, and Nicholas Ayache. Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents. *NeuroImage*, 55(3):1073–90, April 2011.

[23] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[24] M.D. Fox. Segmentation of Edge Preserving Gradient Vector Flow: An Approach Toward Automatically Initializing and Splitting of Snakes. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:162–167, 2005.

[25] Alejandro F Frangi, Wiro J Niessen, Koen L Vincken, and Max A Viergever. Multiscale vessel enhancement filtering. *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1496:130–137, June 1998.

[26] Todd a Gillette, Kerry M Brown, and Giorgio a Ascoli. The DIADEM metric: comparing multiple reconstructions of the same neuron. *Neuroinformatics*, 9(2-3):233–45, September 2011.

[27] David E Goldberg and Jon Richardson. Genetic algorithms with sharing for multimodal function optimization. In John J Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, volume 20, pages 41–49. University of Alabama, L. Erlbaum Associates Inc., 1987.

[28] R W Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[29] Georges R Harik. Finding Multimodal Solutions Using Restricted Tournament Selection. In Larry Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31. Morgan Kaufmann Publishers Inc., 1995.

[30] M Sabry Hassouna and Aly a Farag. Variational curve skeletons using gradient vector flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2257–74, December 2009.

[31] Trevor Hastie and Werner Stuetzle. Principal Curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[32] J H Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[33] Chang-Hwan Im Chang-Hwan Im, Hong-Kyu Kim Hong-Kyu Kim, Hyun-Kyo Jung Hyun-Kyo Jung, and Kyung Choi Kyung Choi. A novel algorithm for multimodal function optimization based on evolution strategy. *IEEE Transactions on Magnetics*, 40(2):1224–1227, 2004.

[34] Institute of Molecular Pathology. Research Report 2011. Technical report, Institute of Molecular Pathology, 2011.

[35] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[36] David N Kennedy, Steven M Hodge, Yong Gao, Jean a Frazier, and Christian Haselgrove. The internet brain volume database: a public resource for storage and retrieval of volumetric data. *Neuroinformatics*, 10(2):129–40, April 2012.

[37] Ken-Ichi Kimura, Tomoaki Hachiya, Masayuki Koganezawa, Tatsunori Tazawa, and Daisuke Yamamoto. Fruitless and doublesex coordinate to generate male-specific neurons that can initiate courtship. *Neuron*, 59(5):759–769, 2008.

[38] Joseph B Kruskal and Myron Wish. *Multidimensional Scaling*. Springer Texts in Statistics. Sage Publications, 1978.

[39] H W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[40] Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-Based Validation of Clustering Solutions. *Neural Computation*, 16:1299–1323, 2004.

[41] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction (Information Science and Statistics)*. Springer, 2007.

[42] Jian-Ping Li, Marton E Balazs, Geoffrey T Parks, and P John Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary computation*, 10(3):207–34, January 2002.

[43] Jane A Loevinger. A systematic approach to the construction and evaluation of tests of ability. *Psychological Monograph*, 61:1–49, 1947.

[44] Artur Luczak. Spatial embedding of neuronal trees modeled by diffusive growth. *Journal of neuroscience methods*, 157(1):132–41, October 2006.

[45] SW Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1995.

[46] Nicolas Y Masse, Sebastian Cachero, Aaron D Ostrovsky, and Gregory S X E Jefferis. A mutual information approach to automate identification of neuronal clusters in Drosophila brain images. *Frontiers in neuroinformatics*, 6(June):21, January 2012.

[47] Christopher Masser. *Construction of an enhancer library for neuronal circuit dissection in Drosophila melanogaster and its employment to identify neurons involved in male courtship behavior*. PhD thesis, University of Vienna, 2012.

[48] Laura J J Nicolaï, Ariane Ramaekers, Tim Raemaekers, Andrzej Drozdzecki, Alex S Mauss, Jiekun Yan, Matthias Landgraf, Wim Annaert, and Bassem A Hassan. Genetically encoded dendritic marker sheds light on neuronal connectivity in Drosophila. *Proceedings of the National Academy of Sciences of the United States of America*, 107(47):20553–20558, 2010.

[49] N Otsu. A Threshold Selection Method from Gray-Level Histograms. *Ieee Transactions On Systems Man And Cybernetics*, 9(1):62–66, 1979.

[50] Damaris Pascual, Filiberto Pla, and J. Salvador Sánchez. Cluster validation using information stability measures. *Pattern Recognition Letters*, 31(6):454–461, April 2010.

[51] Hanchuan Peng, Phuong Chung, Fuhui Long, Lei Qu, Arnim Jenett, Andrew M Seeds, Eugene W Myers, and Julie H Simpson. BrainAligner: 3d registration atlases of. *Nature Methods*, 8(6):2–9, 2011.

[52] A Petrowski. A clearing procedure as a niching method for genetic algorithms. *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 798–803, 1996.

[53] Adaptive Probabilities and Genetic Algorithms. Adaptive Probabilities of Crossover and. *IEEE Transactions On Systems Man And Cybernetics Transactions On Systems Man And Cybernetics*, 24(4):656–667, 1994.

[54] Alexander Rakhlin and Andrea Caponetto. Stability of k-means clustering. In *NIPS*, pages 1121–1128, 2006.

[55] Nilanjan Ray and Scott T Acton. Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Transactions on Medical Imaging*, 23(12):1466–78, December 2004.

[56] Torsten Rohlfing and Calvin R Maurer. Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, 7(1):16–25, March 2003.

[57] J W Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[58] Alexei V Samsonovich and Giorgio A Ascoli. Statistical determinants of dendritic morphology in hippocampal pyramidal neurons: A hidden Markov model. *Hippocampus*, 15(2):166–183, 2005.

[59] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2(2):143–168, 1998.

[60] B Schölkopf, A J Smola, and K R Müller. Kernel Principal Component Analysis. *Advances in Kernel Methods support Vector Learning*, 1327(3):327–352, 1999.

[61] Susanne Schönknecht, Carsten Duch, Klaus Obermayer, and Michael Sibila. 3D Reconstruction of Neurons from Confocal Image Stacks and Visualization of Computational Modeling Experiments. *Bildverarbeitung für die Medizin 2008*, XXI:475, 2008.

[62] L Shapiro and G Stockman. *Computer Vision*. Texts in Computer Science. Prentice Hall, 2002.

[63] R Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[64] Gulshan Singh and Kalyanmoy Deb. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*, page 1305, 2006.

[65] M B Sokolowski. Drosophila: genetics meets behaviour. *Nature Reviews Genetics*, 2(11):879–890., 2001.

[66] Petra Stockinger, Duda Kvitsiani, Shay Rotkopf, László Tirián, and Barry J Dickson. Neural circuitry that governs Drosophila male courtship behavior. *Cell*, 121(5):795–807, 2005.

[67] J B Tenenbaum, V de Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.)*, 290(5500):2319–2323, 2000.

[68] WS Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4):401–419, 1952.

[69] Patrick Alasdair Turner. *Genetic Algorithms and Multiple Distinct Solutions*. Master thesis, University of Edinburgh, 1994.

[70] L Van Der Maaten, E Postma, and J Van Den Herik. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research*, 10(2009-05):1–41, 2009.

[71] Ragini Verma, Parmeshwar Khurd, and Christos Davatzikos. On analyzing diffusion tensor images by identifying manifold structure using isomaps. *IEEE Transactions on Medical Imaging*, 26(6):772–778, 2007.

[72] Anne C Von Philipsborn, Tianxiao Liu, Jai Y Yu, Christopher Masser, Salil S Bidaye, and Barry J Dickson. Neuronal control of Drosophila courtship song. *Neuron*, 69(3):509–522, 2011.

[73] Dhananjay A Wagh, Tobias M Rasse, Esther Asan, Alois Hofbauer, Isabell Schwenkert, Heike Dürrbeck, Sigrid Buchner, Marie-Christine Dabauvalle, Manuela Schmidt, Gang Qin, Carolin Wichmann, Robert Kittel, Stephan J Sigrist, and Erich Buchner. Bruchpilot, a protein with homology to ELKS/CAST, is required for structural integrity and function of synaptic active zones in Drosophila. *Neuron*, 49(6):833–44, 2006.

[74] T A Witten and L M Sander. Diffusion-limited aggregation. *Physical Review B*, 27(9):5686–5697, 1983.

[75] C Xu and J L Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–69, January 1998.

[76] Jai Y Yu, Makoto I Kanai, Ebru Demir, Gregory S X E Jefferis, and Barry J Dickson. Cellular organization of the neural circuit that drives Drosophila courtship behavior. *Current Biology*, 20(18):1602–1614, 2010.

[77] D Zaharie. Extensions of differential evolution algorithms for multimodal optimization. In *In Proceedings of SYNASC'04, 6th International Symposium of Symbolic and Numeric Algorithms for Scientific Computing*, pages 523–534, 2004.

[78] Kaizhong Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(3):205–222, 1996.