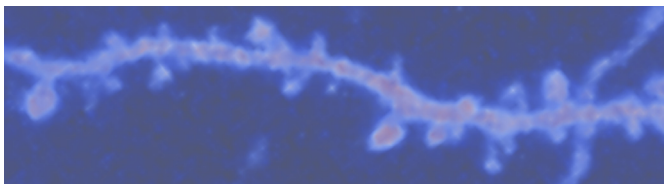
Dendritic Spine Detection and Segmentation for 4D 2-Photon Microscopy Data Using Statistical Models from Digitally Reconstructed Fluorescence Images



Inauguraldissertation
zur
Erlangung der Würde eines Doktors der Philosophie
vorgelegt der
Philosophisch-Naturwissenschaftlichen Fakultät
der Universität Basel

von

Clemens Blumer
aus Basel (BS) und Engi (GL)

Basel, 2013

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät auf Antrag von

Prof. Dr. Thomas Vetter
Universität Basel, Dissertationsleiter

Prof. Dr. Volker Roth
Universität Basel, Korreferent

Basel, den 15.10.2013

Prof. Dr. Jörg Schibler
Dekan

creative commons

Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 2.5 Schweiz

---

Sie dürfen:

das Werk vervielfältigen, verbreiten und öffentlich zugänglich machen

Zu den folgenden Bedingungen:

Namensnennung. Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen (wodurch aber nicht der Eindruck entstehen darf, Sie oder die Nutzung des Werkes durch Sie würden entlohnt).

Keine kommerzielle Nutzung. Dieses Werk darf nicht für kommerzielle Zwecke verwendet werden.

Keine Bearbeitung. Dieses Werk darf nicht bearbeitet oder in anderer Weise verändert werden.

- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter welche dieses Werk fällt, mitteilen. Am Einfachsten ist es, einen Link auf diese Seite einzubinden.
- Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die Einwilligung des Rechteinhabers dazu erhalten.
- Diese Lizenz lässt die Urheberpersönlichkeitsrechte unberührt.

Quelle: http://creativecommons.org/licenses/by-nc-nd/2.5/ch/ Datum: 3.4.2009

Abstract

The brain with its neurons is a complex organ which is not yet fully decoded. Many diseases and human behavior are affected by the brain. Therefore, neurobiological experiments are conducted. In neurobiological experiments, the imaging of neurons is a key technology. Thanks to 2-Photon Microscopy (2PM) it is feasible to image the volume of labeled, living, pyramidal neurons. Moreover, in a second channel a different marker can label specific structures or proteins. Small structures like synapses are not visible in 2PM. However, the size of spines has a relation to the strength of synapses. Therefore, the focus lies on the study of dendritic spines. Imaging over time is possible and generates multiple fluorescence images. The analysis of fluorescence images is difficult, time consuming and error-prone even for experts. Furthermore, the reproducibility of manual analysis is not guaranteed. Therefore, the automatic detection, segmentation and tracking of dendritic spines in 2PM data is required.

We will introduce a full pipeline to detect, segment and track spines in time series from 2PM data. We train a statistical dendrite intensity and spine probability model trained with 2-Dimensional (2D) data from Digitally Reconstructed Fluorescence Images (DRFIs). DRFIs are synthetic images which can be computed from geometrical shapes of dendrites and their spines. Electron Microscopy (EM) reconstructions contain the shape of dendrites and spines. Automatic EM reconstructions are a difficult task. However, due to specially prepared samples manual reconstruction becomes feasible. The computation of DRFIs using Serial Block-Face Scanning Electron Microscopy (SBFS-EM) data is a novelty. This concept enables us to overcome the issue of expert labeled spines in fluorescence images. In many image analysis approaches the backbone of dendrites is the central part of the approach. However, the backbone only exists by definition and it is difficult to place it in fluorescence images. The dendrite intensity model is based on 2D slices which are orthogonal to the backbone. Therefore, it is feasible to optimize the backbone of any dendrite such that it is optimally located with respect to the backbone definition and for the model at the same time. We are able to predict the spine probability for 2D slices at every pixel. This is enabled by the information transfer from the SBFS-EM domain to the fluorescence image domain. In combination with further features, which are computed over multiple slices, a robust spine prediction is feasible. This prediction can be projected back to the original 3-Dimensional (3D) space of the image. Thus, a prediction and segmentation of spines in 3D is possible.

Imaging time series of dendrite pieces is a challenging task. The same structure (region of interest) must be imaged using the same parameters at each time point. Due to handling of the sample (e.g. storing the samples in an incubator) between different imaging steps, the corresponding region of interest must be located at each time point. Accordingly any movement of the region of interest must be compensated in translation and rotation with respect to a coordinate system established at a specific time point. Rotations around the optical axis of the microscope (z-axis) are more probable than rotations around the x- and y-axis because of the solid glass plate (Petri

dish) on which the sample is arranged. In contrast, translations are feasible in all directions. Furthermore, the imaged dendrites are alive. It is therefore possible that their shape and orientation in space changes over time. This introduces transformations which are non-rigid. The estimation of such changes is not tractable as only dendrite changes should be compensated, but not spine changes. Therefore, we will use a rigid registration which enables translation and rotation. Additionally, tracking of the spine candidates over the registered time points is required because some movement of spines is possible. Successful tracking of spines enables to trace intensity changes of spines. The tracing of intensity changes is feasible for multiple image channels and opens the possibility of manifold applications.

Based on the detection and segmentation of spines we will finally demonstrate the successful solution of image analysis for fluorescence time series in its whole complexity in some practical applications. In one experiment we show the ability of detecting spines which have a presynaptic bouton. For this detection and classification single time point images with multiple image channels are used. In a second part we demonstrate the successful detection and segmentation of spines in time series. Furthermore, we demonstrate in time series the possibility to detect spines having an Endoplasmic Reticulum (ER). In experiments like this, the whole complexity of image analysis for fluorescence time series must be solved.

In summary, our contribution is the introduction of a concept to compute DRFIs using SBFS-EM data. This enables the information transfer between image domains. Related to the computation of DRFIs we introduce statistical models of dendrite intensity and spine probability maps. The statistical models are central to detect and segment spines. Furthermore, the challenge of analyzing fluorescence image series is solved by the spine prediction followed by registration and tracking of spine candidates. Finally, the application on real biological experiments demonstrates the practical use of automatic image analysis for fluorescence images in single time points and time series.

Acknowledgements

This thesis was supported by many people. First of all I want to thank Thomas Vetter and his whole research group GraVis for the professional and social time we shared together. The conceptual idea of using statistical models was inspired by the Morphable Model that is used and developed in the group.

From the beginning my project and research was motivated by the subject of interdisciplinarity, which was given by Thomas Oertner and his lab. I would like to thank him for the collaboration and the insights into neurobiology I gained due to him. From his previous lab at the Friedrich Miescher Institute for Biomedical Research, FMI Basel, I would like to thank very much Cyprien Vivien. He and Christel Genoud (also FMI) generated the data for the approach using SBFS-EM data and the evaluation with Correlative Light Electron Microscopy (CLEM) data. From the Oertner lab at the Center for Molecular Neurobiology Hamburg, ZMNH, I especially want to thank Simon Wiegert and Alberto Perez Alvarez for their inputs in software requirements, usability, testing and providing data and results from experiments.

Contents

Notation

A	Matrix
a	Vector
$\vec{v}$	2D and 3D Vector
p or $p(\dots)$	2D and 3D Point
$P(\dots)$	Probability of, $0 \leq P(\dots) \leq 1$
t_i	Time point i of time series
I	Image
s_d or $s_{d,i}$	2D dendrite intensity slice
s_s or $s_{s,i}$	2D spine probability map
M_d or $M_{d,k}$	Dendrite intensity model
M_s or $M_{s,k}$	Spine probability model
$\|\cdot\|$	Euclidean Norm
$\|\cdot\|$	L^1 Norm
$*$	Convolution
$a \cdot b$	Multiplication of a and b
$A \circ B$	Element-wise multiplication of matrix A and B (Hadamard product)
$a \circ b$	Element-wise multiplication of vector a and b (Hadamard product)
$\mathscr{F}$	Fourier transformation
$\mathscr{F}^{-1}$	Inverse Fourier transformation
$\#(\dots)$	Number of elements
$\approx$	Approximation
$\overline{a}$ or μ_a	Average of a where a is a list or a vector
DRFI$(\dots)$	Digitally Reconstructed Fluorescence Image
kB	KiloByte (10^3 byte)
MB	MegaByte (10^6 byte)
GB	MegaByte (10^9 byte)

1. Introduction

Image analysis of biological data is a growing field. This is mainly driven by the fact that health and biology are very important topics for humans and the technical evolution of the last century linked to research opens a wide range of new possibilities. However, these new possibilities require in turn new technical solutions and, therefore, a link back to technical research (and image analysis) is given.

We will introduce the problem and the motivation to solve it in Section 1.1. In Section 1.2 we will give a view on the prior work of the field. The field is not very old, but different approaches were tried to solve the problem though. Finally, in Section 1.3 we will introduce our contribution. We will also discuss how our approach is different from prior work and how it solves partially different problems.

1.1. Motivation

Health is of great importance for people. The Latin sentence *"Mens sana in corpore sano"* ("A healthy mind in a healthy body") from the poet Juvenal is often cited even in our modern society. It shows that the most important asset is health. In 2010 the worldwide health market had a volume of about US$856 billion [13]. This corresponds to about 1.4 % of the worldwide gross domestic product (GDP). This shows that people are aware of the health problem and are also keen to promote research. This is underlined by the fact that in 2010 the pharmaceutical industry has invested more than US$70 billion in research and development. For more details about further facts of the health industry we refer the interested reader to [13]. Health is central in the human life and a lot of effort is made to improve health and to fight diseases. Different genetic diseases as well as diseases which are more likely by genetic predisposition (e.g. Alzheimer disease) exist. These diseases and the care and therapy of them cost a lot of money. Moreover, the burden for the affected patient and his family can be incredibly large. Many diseases affect the brain or are related to it. However, the human brain is the most complex organ and we do not know exactly how it works. Therefore, a lot of basic research tries to decode the human brain. This decoding of the brain is needed to understand the healthy brain and its difference to the brains of sick people. Furthermore, the difference between healthy and ill organs and their interaction with the diseases is central. Therefore, research that examines the impact of diseases on the brain and neurons is of great importance. In biological experiments with neurons, it is feasible to study the difference between neurons of healthy and ill persons and animals respectively. This opens the possibility to have various technical requirements at the same time, which advances the technical research.

In recent years, the imaging of neurons using optical microscopy has almost become the standard approach. Microscopy like Laser-Scanning Confocal Microscopy (LSCM) and 2-Photon Microscopy (2PM) is available in many laboratories. Imaging became easier. Image quality and preparations were improved. All these facts make imaging

attractive and popular. Furthermore, the concepts of preparation and imaging are almost standardized. Therefore, the production of data is less time and resource intensive than it was in the past. The concept of imaging living neurons enables to make time series. This strongly increases the amount of data to be analyzed. However, the analysis of fluorescence images is in general a very time consuming and error-prone task. Moreover, the result of analysis is of interest for biologists rather than the time consuming segmentation process itself. However, the motivation to use computer science and image analysis has increased. The need of automatic image analysis to detect and to segment neuronal structures and to compute different statistical values requires algorithms which are specially adapted to the problem. Fluorescence images are very characteristic images. Image processing and analysis of fluorescence images cannot be compared directly to the image analysis of photos from cameras (photography). Furthermore, image analysis of fluorescence images is not well studied compared to other subjects. Beside all the traditional, well established algorithms in image analysis, there is a high demand for algorithms to analyze fluorescence images.

In summary we can say that the combination of a rather new research field, i.e. image analysis of fluorescence data, combined with biological basic research related to human health is a very important, interesting research area which is in demand.

From a biological point of view it is interesting to study the changes of dendritic spines in living neurons over time. This very specific requirement is an interesting application to demonstrate the power of image analysis and our concepts. We can paraphrase our task to enable to detect, segment and track spines of living, pyramidal cells of rats over time in fluorescence images from 2PM.

1.2. Prior Work

Prior to this work, different approaches for spine segmentation and image analysis of fluorescence as well as 2PM data were tried. In many respects, most of these are similar to classical methods of image analysis. However, the combination of such approaches and their application to the problem can be very unique. We will introduce here the most important prior work which is applied to spine detection and/or segmentation in fluorescence images from 2PM or LSCM.

Most approaches for spine detection, segmentation or just neuron tracing use a skeleton or backbone of the dendrite [10,11,32,34,37,44,45,45,50,58,60–62]. The huge number of approaches using the backbone or skeleton in some way shows the importance of the backbone. It demonstrates that the backbone can be a central key for dendrite representation and spine detection. However, approaches which focus on backbones have some disadvantages as most of them produce a high rate of False Positive (FP) and False Negative (FN) results if the backbone is not found correctly.

The detection of the spine tip followed by a segmentation (mostly region growing) is another method which is used by different approaches [24, 35, 50, 62]. The tip of the spine is very characteristic because it has a high curvature and the largest distance to the backbone. However, the difficulty is to exclude false spine tips (FP) which are mainly generated by background noise.

Since 2008 almost all approaches mainly use 3-Dimensional (3D) information or the 3D fluorescence image. The approaches based on Maximum Intensity Projection (MIP) are being phased out. The reasons for this change are mainly the evolution of hardware

and the information loss by MIP. Since about 4-6 years powerful computers have enough main memory to load and process even larger, 3D fluorescence images. As the computational power has also increased, hardware is not a bottleneck anymore.

Spines can appear detached in fluorescence images because of photo bleaching and image noise. In reality there is always a connection to the dendrite. Because of detached spines, many approaches have a dual spine detection or segmentation method for attached and detached spines [3, 10, 11, 34, 61, 62]. Many of these approaches use the side branches of the skeleton to detect spines. Most approaches with dual spine detection use data from LSCM and not from 2PM. This is reasonable because in LSCM images the structures appear slightly sharper and the probability of detached spines is higher.

Fluorescence images are strongly smoothed by the Point Spread Function (PSF). Therefore, the idea of deconvolution is very interesting but also complex. About half of the approaches (see also Table A.1 in Appendix A) use deconvolution. This ranges from the integration of simple deconvolution concepts up to the use of some advanced third party software to do deconvolution before applying the image analysis pipeline. The fact that mainly third party deconvolution software is used shows its difficulty. Deconvolution is another research field. Based on the detection quality of the different approaches it is impossible to conclude if deconvolution is required or not to get best spine detection and segmentation results. It shows more that the algorithms are optimized to work on the corresponding data. Optimization on data is also underlined by the fact that almost all approaches use LSCM or 2PM data but not both. Some approaches mention the use of both data types, but in general results are only demonstrated using one type.

In Appendix A in Section A.1 we present an overview of most prior work in Table A.1. We list the most important properties of each spine detection and segmentation approach to give an overview of the different directions of research.

1.3. Contribution

On the one hand, our contribution to the problem of spine detection and segmentation is very specific, but on the other hand, it yields also some more generally applicable ideas and concepts. Our main contributions are divided into the three parts Digitally Reconstructed Fluorescence Image (DRFI), model computation and practical application. An overview of the whole approach is shown in Appendix A in Section A.2. In Chapter 3 we introduce as a novelty the method of computing synthetic fluorescence images, called DRFIs, from Electron Microscopy (EM) reconstructions. We introduce the full pipeline of computing synthetic fluorescence images from real geometrical shapes. Furthermore, the use of the Correlative Light Electron Microscopy (CLEM) offers new validation possibilities. The computation of DRFIs enables us to introduce in Chapter 5 a statistical dendrite intensity and spine probability model based on a larger amount of training data which is not expert labeled. In Chapter 6 we introduce the application of the new model to the problem of spine detection and segmentation and extend the approach to time series in Chapter 7. Additionally, in Chapter 8 we discuss the distribution of spines in space and the impact of the distribution in space to spine detection and segmentation algorithms. To sum up we can say that beside our contribution to the detection and segmentation of spines in 4-Dimensional (3-Dimensional

plus time) (4D) fluorescence data we also introduced new concepts. These new concepts can be summarized as statistical models based on CLEM data.

In difference to prior work our approach is based on a statistical model of intensities and probabilities. Furthermore, we demonstrate the use of spine detection and segmentation on time series and multi-channel fluorescence images. Even a combination of both applications to five dimensional data (4D + multi-channel yielding 5D(x, y, z, t, λ)) is feasible. We present results of time series analysis which enable to track changes of spines over time in Section 7.3. These results are fully based on automatic analysis. Moreover, we demonstrate the success of a two channel analysis to detect spines with special properties in Section 6.6.2 (single time point) and Section 7.3.2 (time series). The demonstration of these two applications shows the practical importance of automatic spine analysis and the overall value of our approach. Additionally, we are not aware that anyone has ever before demonstrated the automatic analysis of spines over time and with fluorescence images with multiple channels.

2. Background

This thesis covers a field which connects computer science to biology and neurobiology. It is a typical interdisciplinary work. There exist basic theories and technologies in both fields. In this background chapter we give an overview on the most important topics. The background chapter describes topics from both disciplines and some that are relevant for both fields. We describe the biological background (neurons) and experimental setups in Section 2.1 and Section 2.2 respectively. In Section 2.3 we will discuss fluorescence images, their generation and related issues. Section 2.4 introduces Electron Microscopy (EM) and in more detail Serial Block-Face Scanning Electron Microscopy (SBFS-EM). Finally, we will move to background knowledge of image analysis and will introduce ideas and concepts of statistical models in Section 2.5. The background of statistical models will be very general because an applied introduction of statistical models is given in Chapter 5, Section 5.3.

2.1. Neurons

Neurons are one of the most important cell types of any living structure. Multiple neurons build the brain. It permits individuals to process the input coming from the environment. Furthermore, the brain (at least the human one) makes it possible to have thoughts and to plan abstract processes. The key numbers of the human brain are very impressive. They underline why it is so difficult to understand the way the brain works. The human brain is composed of 10^{11} neurons. Every neuron has on average about 7000 synapses. The total number of synapses of a human, adult brain is estimated to be in the range of 10^{14} to 5×10^{14} synapses. In Table 2.1 we introduce the most important structures of neurons with focus on this work.

The basic structure of neurons is well known since the 20^{th} century. Figure 2.1 shows a schematic diagram of a neuron. Current research tries to study and understand how multiple cells work together. This should help to answer the question how a brain works. For studies like this neither dead nor single cells are of interest. It is tried to understand how multiple, living cells work together as a system. Moreover, the relation of neurons to diseases with all their aspects is tried to understand. Therefore, the research of living networks of neurons contributes to the field of system biology.

Basic communication from cell to cell is already well understood. The signal travels from cell to cell by releasing vesicles from the first cell (presynaptic) via axon to a second cell (postsynaptic). At the second cell the dendrite receives most of these signals at spines. If a transport of vesicles takes place, then a synapse is built between the axon and dendrite. The release of vesicles is triggered by an action potential. The synapse is a very interesting structure of neurons because it connects two cells. We refer the interested reader to the corresponding literature, e.g. [16], for a deeper understanding of neurons and their signal processing.

Structure	Description
Action Potential (AP)	An action potential is a short change (1 ms to 2 ms) of the electrical membrane potential. The membrane potential deviates from the rest potential. Action potentials are binary signals. No strong or weak action potentials exist. However, the frequency of action potentials corresponds to the signal strength.
Axon	Axons are like dendrites long tubular structures. However, at axons the signal is emitted and is transmitted by synapses from one cell to the next.
Dendrite	Dendrites are long tubular shaped structures which build a large tree-like construct. Most signal input to the cell arrives over dendrites, more exactly at synapses which are frequently located at spines.
Endoplasmic Reticulum	The Endoplasmic Reticulum (ER) is a dynamic structure of membrane [51]. Spines containing ER have enhanced synaptic strength and are preferential sites for a particular type of synaptic plasticity called mGluR-dependent depression [28].
Membrane	The membrane builds the surface of the cell. Under certain situations ion and vesicle can pass trough the membrane.
Mitochondrion	Mitochondria deliver most of the energy for the cell. Therefore, mitochondria are also called "cellular power plants".
Nucleus	The core of the cell. Contains most of the cell's genetic material.
Soma	A cell is structured into soma (cell body), axons and dendrites. Many important structures like mitochondria and Endoplasmic Reticulum are enclosed by the soma.
Spine	Spines are the protrusions located at dendrites. Most synaptic connections between cells take place at spines. Spines can be very densely distributed along dendrites. A cell has about 200 000 spines.
Synapse	The communication point between two cells is called synapse. A signal from one cell to the next travels over a synapse.
Vesicle	Vesicles are containers for different transmitters. Vesicles can be transported and released from cell to cell. Multiple vesicles together form a so called bouton in the presynaptic axon.

Table 2.1.: Most important structures of a neuron with focus on this thesis.

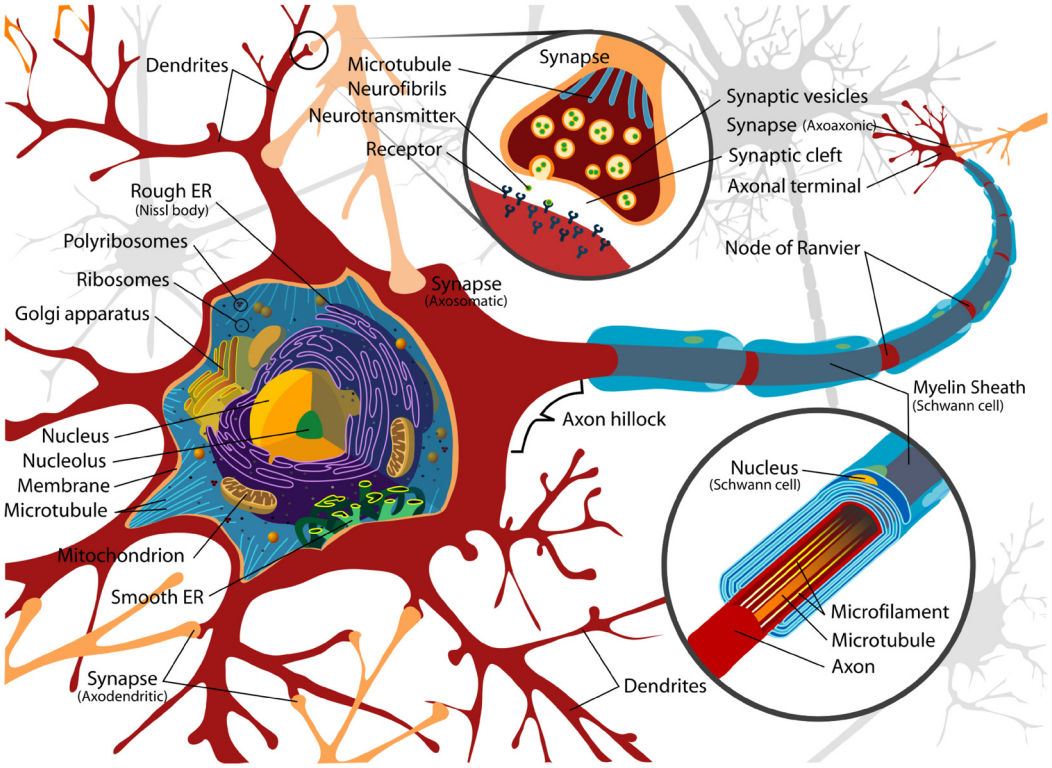


Figure 2.1.: Schema showing the different parts of a neuron [56]. Signals travel by axons over synapses to dendrites of other neurons.

2.2. Biological Experiments with Neurons

In the previous section we introduced basic knowledge about neurons. In this section we will introduce how neurons can be used for biological experiments to get new insights on reasons of disease (e.g. Alzheimer disease) or the functionality of the brain. Most experiments with living and/or stimulated neurons have the goal to compare the reactions of healthy and sick cells or how the cells react in general on a stimulation. Imaging of living neurons results in worse image quality than imaging fixed neurons. However, living neurons can be stimulated and also time series imaging is possible. For experiments with living neurons fluorescence imaging and imaging approaches like 2-Photon Microscopy (2PM) (see Section 2.3.2) become crucial. Many studies are conducted with cells from animals like mouse or rat. These animals serve as object of study because their brain is simpler. Nevertheless they possess the ability to learn. Often the cells are cultured and not directly studied in the animal. Neither to culture human cells nor direct imaging of human neurons in the brain would be feasible or generally allowed.

The overall goal of all (biological) experiments is to gather data and observations under a controlled environment to validate theories, test new theories or even conclude new theories. This means that experiments are a tool for scientists to carry out their work and research.

In principle most experiments have a similar concept and structure. The goal is to conduct experiments in which living cells under eventually special situations can react and during and after the reaction measurements (imaging) takes place. In some experiments also a single time point is used to get a measurement and make conclusions. Roughly described the following steps are conducted:

- Culture of cells.
- Insertion of fluorescence into some cells.
- Imaging of a baseline.
- Recording of images during and after the stimulation or treatment.
- Measurement of statistical values at interesting locations (e.g. spines) and tracking of changes over time.

Culture of Cells

Culture of cells, called cell culture, is the complex process of breeding, growth and storing of cells in a controlled environment. The history of cell culture is quite old and goes back to the 19th century. Cells are cultured in a controlled environment in a (cell) incubator with fixed parameters like temperature, humidity and gas environment. An incubator is basically a storing location for the cells with heating and is attached to a controlled gas mixture. The cells can grow directly in their tissue or also in an alternative cell growth material. Cells are fed by media mostly derived from animal blood. Therefore, it can happen that the cells are contaminated with viruses. It is difficult to maintain healthy cell cultures. Another danger is cell line cross-contamination. By cross-contamination from one cell line to another cell line a contamination takes place. In summary, cell cultures make it possible to work with controlled cells but also require an experienced and faithful scientist or technician.

Insertion of Fluorescence

Neurons are not fluorescent themselves. Therefore, it is required to insert the fluorescence in any way. There exist different concepts. Basically fluorescence can be inserted by a dye or using fluorescent proteins. The protein can be transferred by Deoxyribonucleic acid (DNA) using a gene gun or viruses. We refer the reader for further details about fluorescence to Section 2.3 and Section 2.3.2.

Baseline Imaging

The study of measured values requires a baseline. Cells are living structures and each preparation is individual (even though same parameters can be used). Therefore, images and results cannot be compared directly. Furthermore, microscopy has many parameters and most probably some are changed, e.g. imaging depth. Therefore, for each experiment a baseline is imaged when the neuron is under a so called relaxed situation. Then a comparison between different images, cultures and experiments is possible by normalizing by the baseline. In our application it is possible to use the average dendrite intensity as baseline and to normalize by it.

Imaging During Experiments

Imaging is not part of the experiment. However, is the essential technique to measure the state of the cell. Imaging can be seen as taking measurements of different parameters. Imaging maps the state of the neuron or parts of it to a number which then can be compared. Images can be taken before, during and after a stimulation of the neuron. The difficulty of imaging neurons multiple times is given by the following reasons. The danger of photobleaching (see Section 2.3.2) is much higher if cells are repeatedly imaged. Additionally, the risk that the cells die or are contaminated is much higher because of the multiple interactions with the cells.

Statistical Values of Interesting Structures

Imaging of neurons does not deliver a single number. It is a measurement at different locations (sampling) which are uniformly distributed over the dimensions and space. These measurements are the basis for all conclusions. However, it is required to calculate measurements for certain structures. Processing or comparing the whole image at once does not allow to conclude anything. After imaging the huge work of data analysis starts and automatic image analysis has a central role. It is tried to calculate an intensity value for known structures like spines. These statistical values of structures can then be normalized with the baseline and a comparison between the different measurements (images) becomes possible.

2.3. Fluorescence Image

Fluorescence images are well accepted in the neurobiological field. Imaging of cells using fluorescence imaging is often done. Basically, fluorescence images are structured like all classical 3-Dimensional (3D) images. However, they have a very specific and characteristic Point Spread Function (PSF). In principle also fluorescence imaging is

a light microscopy approach. In contrast to normal images from photography or light microscopy, in fluorescence imaging the properties of objects are differently mapped to the image. There does not exist a direct mapping function. We will introduce the properties of fluorescence images in Section 2.3.1. In Section 2.3.2 2PM in particular and Multi-Photon Microscopy (MPM) in general will be introduced. The important issue of the PSF will be discussed in Section 2.3.3, followed by the ideas of deconvolution in Section 2.3.4.

2.3.1. Fluorescence Image Properties

Fluorescence images are structured like common 3D images. They have three dimensions (x, y and z) and are composed of pixels or voxels. In fluorescence image no edges are visible in difference to images constructed from photographies. With this aspect fluorescence images are more similar to medical images like Magnet Resonance Imaging (MRI) (also called Magnet Resonance Tomography (MRT)) or Computer Tomography (CT) images. However, the issues of fluorescence images are still different from medical images. In addition to the lack of any edge information in the images, also the z-resolution is much worse than the x- and y-resolution in high-resolution fluorescence images. It can be seen as a smoothing of the image and no high-frequency information is part of the image. Unluckily, the PSF is also elongated in the z-direction. Therefore, the application of real 3D image analysis approaches and algorithms is quite difficult and requires mostly to care in detail about the issues with the z-resolution. In Figure 2.2 we show a xy-, xz- and yz-slice of a 3D fluorescence image. In this image the issue of the differences between horizontal (x, y) and axial (z) resolution and sampling is well visible. In z-direction the intensity changes are not continuous anymore. There exist also intensity changes along the object which are caused by the object (neuron) properties as well as the distribution of fluorescent dye or protein. This is best visible in the image showing the xz-slice. Furthermore, the object (dendrite) only has very few pixels in the axial direction. This is highlighted by the typical image resolution (image spacing) of fluorescence images. Typically the z-spacing is about 0.5 µm while the x- and y-spacing are in the range of less than 0.1 µm. This effect is not only a sampling issue but also a limitation of the maximal resolution possible using 2-Photon Microscopy (see Section 2.3.2). Furthermore, the intensity in the image does not correspond to light reflected from the object (not like in classical photography). It is the detected fluorescence that is emitted from the object because of the excitation introduced by light. As further issue the image is taken from a living object. Depending on the preparation and microscope settings huge differences in image intensities can arise in the image (and over multiple images). This means that even for the same structure depending on the image location, imaging settings or preparation different image intensities can be measured. The dynamic of the image intensity depending on the image location is given by the location dependent PSF.

We can summarize the most important properties and issues in image analysis of fluorescence images by the following list:

- No direct mapping of object to the image.
- Imaging resolution limits are different for x-/y- and z-axis.
- Different sampling in z-axis because of different resolution limits.
- No edge information (high-frequency) is in the images.

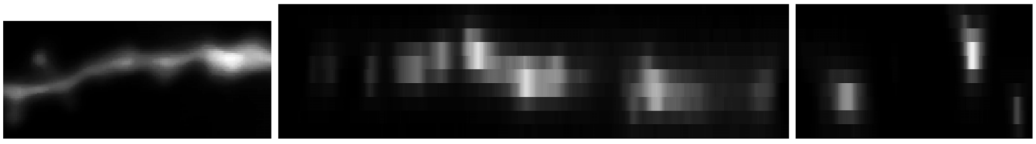


Figure 2.2.: From *left* to *right* we present extracted 2D slices through xy, xz and yz. The worse resolution in the z-direction in difference to the xy-direction is well visible.

- Intensity information is not constant over the image.

All these properties of the image require to use image analysis approaches that are specially designed for fluorescence images.

2.3.2. 2-Photon Microscopy

Fluorescence images are generated by Laser-Scanning Confocal Microscopy (LSCM), 2-Photon Microscopy (2PM), Multi-Photon Microscopy (MPM) or some similar microscopy techniques. In the basic concept all these imaging approaches are similar. They mainly differ in the way the excitation state to emit fluorescence is achieved. First, we will introduce the basic concept of fluorescence microscopy. Then we discuss in more detail 2-Photon Microscopy.

The concept of fluorescence microscopy to generate an image can be split into the following steps:

- Enclose fluorescence in object.
- Excite fluorescence.
- Detect emitted fluorescence.
- Repeat excitation and detection of emission by scanning of sample.

Enclosed Fluorescence

The idea of fluorescence microscopy is based on the property that the object to be imaged has a fluorescent protein or dye enclosed. In 1961 Osamu Shimomura et al. detected and studied Green Fluorescent Protein (GFP) in jellyfish (*Aequorea victoria*) [48, 49]. However, their focus was not on GFP and the discovery of it was a fortunate event. Almost 40 years later, in 1994, Martin Chalfie et al. were able to insert GFP into other species (*E. coli* and *C. Elegans*) than jellyfish [9]. This was the first use of GFP as a marker protein. The final success of fluorescent proteins is thanks to Roger Tsien. He studied GFP in detail and was able to improve it too [52]. In 2008 Osamu Shimomura, Martin Chalfie and Roger Tsien received the Nobel Prize in chemistry for their work leading to the scientific success of GFP and related proteins. For more details about GFP and it worth to get a Nobel Prize we refer the reader also to [21]. This successful story allows biologists worldwide to enclose fluorescent proteins in living structures. The volume or parts of the structure can be made visible by using fluorescence. The fluorescent protein is injected genetically using a so called gene gun (biolistic particle delivery system). Heavy metal particles, e.g. gold particle, are used as medium. The particle are coated by plasmid DNA in which the gene for the fluorescent protein is

part of. Alternatively the dye can be inserted directly using patch pipettes. If the fluorescence should bind to special proteins the gene gun approach is used. Due to binding to specific proteins labeling of different structures is feasible.

Excitation of Fluorescence

The enclosed proteins are not self-emission. The protein must be transferred into an excitation state. Additional energy (light) is required to reach an excited state and fluorescence is emitted. A pulsed laser is focused into the cell by using an adapted light microscope. Photons from the laser hit the fluorescent protein, bring it in to an excited state and the fluorescence is emitted. The excited state is only reached if the laser has the correct wavelength. The required wavelength depends on the used fluorescent protein.

Detection of Fluorescence

The emitted fluorescence must be detected to compose an image from it. Detectors (Photon Multiplier Tube (PMT)) are located at different places and the emitted fluorescence is transformed into a electronic signal and then an intensity value of the image. The measured value is stored in the image. The difficulty of signal detection is to detect as much of the fluorescent emission as possible but only from the focused point. There exist different approaches to ignore fluorescence which does not come from the focused point depending on the imaging system.

Scanning of Sample

To produce an image the sample is scanned (rasterized) by the laser. At each location the emission process is started and fluorescence emission takes place. Additionally, the focus enables to measure at different depths of the sample and to construct from that a 3D image. The name Laser-Scanning Microscopy (LSM) is used for this kind of microscopy because of the scanning process of the laser. More in detail, the laser beam is moved over the horizontal plane which is the scanning or rasterizing. The laser beam is deviated using controllable mirrors to achieve the horizontal movement. For the third dimension the focal point is changed. This changes the depth in which the fluorescence is excited and emitted.

2-Photon Microscopy in a Nutshell

The core principle of fluorescence imaging is introduced in the previous paragraphs. Here we will discuss in detail how the process of imaging works for 2-Photon Microscopy. Figure 2.3 shows all important parts of 2PM and the differences to other approaches. In Figure 2.3, *a*, the concept of getting from the ground state to an excited state is illustrated. Two photons hit within short time (about $0.5 \times 10^{-15}\,\mathrm{s} = 0.5\,\mathrm{fs}$) a molecule in the 2 Photon Absorption (2PA). Together they have the energy to bring the molecule in the excited state. This happens during the laser pulse which takes about $10^{-13}\,\mathrm{s}$ and is repeated every $10^{-8}\,\mathrm{s}$. After reaching the excited state, the normal fluorescence emission takes place. The fluorescence decay lasts for about $10^{-9}\,\mathrm{s}$. The concept of two hitting photons enables to take a laser with higher wavelength (less energy) and nevertheless to have in sum of the two photons the required energy to reach the excited

state. In the schema additionally the effects of Second Harmonic Generation (SHG) and Coherent Anti-stokes Raman Scattering (CARS) are shown which we will not discuss in more detail. In Figure 2.3, *b*, the difference between single-photon (linear) and multi-photon (non-linear) excitation is shown. In single-photon excitation blue light is used. In contrast, 2-Photon Microscopy uses near-infrared, red light. Red light has with 630 nm to 790 nm a larger wavelength than blue light with 420 nm to 480 nm. With increasing wavelength of the light the energy of a single photon decreases. In the single-photon approach a large region is fluorescent, also outside the focal plane. In 2PM only in the focused region fluorescence is emitted. This is because the probability that within the required short time two photon hit the same molecule is only given at the point in focus. The schema of the 2-Photon Microscopy is displayed in Figure 2.3, *c*. A pulsed, Near Infrared (NIR) laser is the energy source. The laser beam is passing different lenses. A mirror changes the xy-position which is the scanning (xy-scanner). After a semi-transparent mirror the laser is focused into a single point in the specimen. At this focused location the 2-Photon Excitation Fluorescence (2PEF) takes place. The fluorescence is emitted isotropically. Second Harmonic Generation (SHG) takes also place in forward direction of the laser beam. The emitted fluorescence is coupled out from the laser beam with mirrors or the semi-transparent mirror and focused using collection lenses. Finally, the fluorescence arrives at the Photon Multiplier Tube (PMT). The PMT converts the fluorescence into a signal. This signal is saved as a pixel value of the fluorescence image.

Photobleaching

The number of times a molecule can emit fluorescence is limited (depending on the protein). If this number is reached the fluorescence emission does not occur anymore. This is then called photobleaching. Photobleaching motivates only to excite the molecule if the fluorescence emission is detected and collected. The multi-photon approach has the advantage that a focused point is emitting fluorescence. This is because in short time two photons must hit one molecule what only happens at the focal point. All the emitted fluorescence is wanted and collected. This reduces the danger of photobleaching compared to LSCM. In the single-photon approach also outside of the focal plane fluorescence is emitted. The unwanted fluorescence from out of focus is removed by a pinhole. This means that some molecule reach an excited state and emit fluorescence which is not collected. The number of emission of these molecules is reduced without any use. This increases the danger of photobleaching for LSCM compared to 2PM.

2-Photon Microscopy vs. Laser-Scanning Confocal Microscopy

In Table 2.2 we compare some properties of 2-Photon Microscopy and Laser-Scanning Confocal Microscopy. These are the most important properties by which 2PM outperforms LSCM in many applications. The main advantages are the focused excitation instead of the pinhole and the possibility to image deeper in the tissue.

For a further, deeper, technical background about 2-Photon Microscopy we refer the interested reader to [26] and for Laser-Scanning Confocal Microscopy to [40].

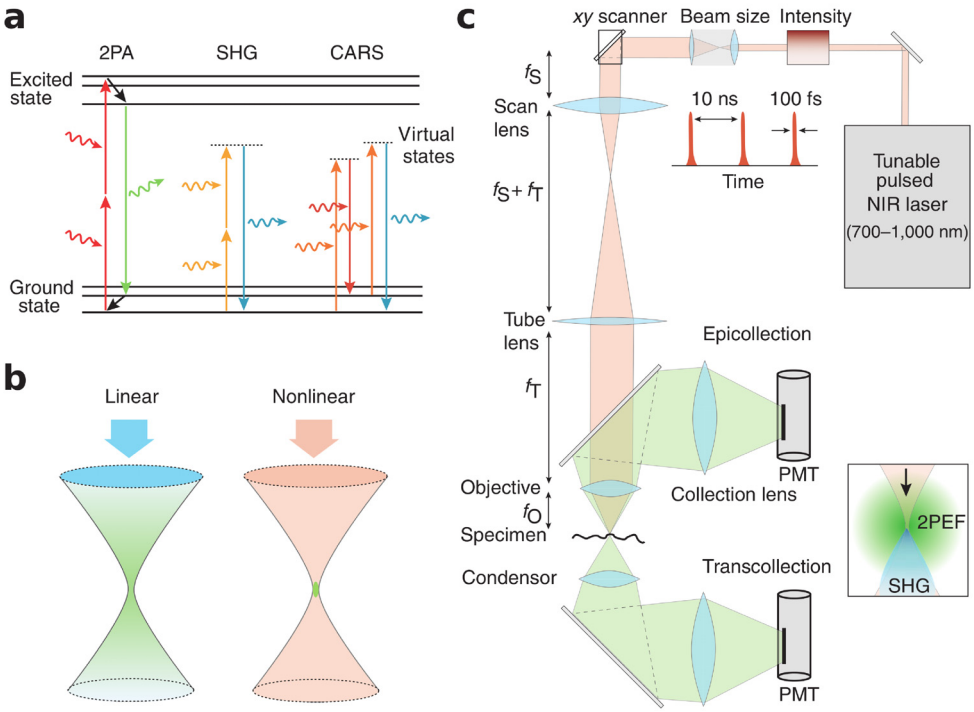


Figure 2.3.: Schema describing the functionality of 2PM. *a:* Jablonski diagram showing the 2 Photon Absorption (2PA). Additionally, showing Second Harmonic Generation (SHG) and Coherent Anti-stokes Raman Scattering (CARS) which result in a virtual state. *b:* Linear and non-linear signal generation and spatial distribution. In the linear case a not focused point is emitting fluorescence. *c:* Schema of 2-Photon Microscopy with all required parts. Figure from [26].

	LSCM	2PM
Excitation	1 Photon	2 Photon
Wavelength	420 nm to 480 nm	630 nm to 790 nm
Pinhole	yes	no
Focused Excitation	no	yes
Maximal Imaging Depth	100 µm	1000 µm

Table 2.2.: Comparison between 2PM and LSCM properties which are different for both approaches. In most cases 2PM outperforms LSCM. Therefore, it is often the better choice and started in the lasts year to be heavily used all over the world.

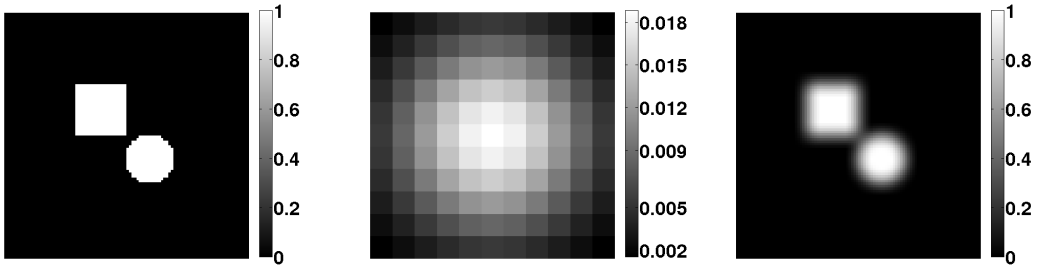


Figure 2.4.: Effect of the PSF on a binary imaging showing a circle and square. All the edges are smoothed out. *Left:* Binary input image. *Middle:* Arbitrary PSF (Gaussian). *Right:* Resulting image of applying the PSF.

2.3.3. Point Spread Function

Each imaging system has a PSF. The PSF is the function that describes how an ideal single point is mapped to the image by the imaging system. Different formulated, the PSF is the response of the imaging system to a point. The geometrical shape and texture of the real object and the PSF composed together as an image I' are described by a convolution:

$$I' = K_{\mathrm{PSF}} * I \tag{2.1}$$

where K_{PSF} is the PSF as kernel and I the image with the original information and structure (before imaging). Figure 2.4 shows the effect of the PSF on a binary image showing a circle and a square. As main effect we can see how the high frequencies (edges) are smoothed out. The effect of how the image is changed depends on the PSF. If the PSF is a Gaussian distribution it is identical to filter the image with a Gaussian kernel.

The PSF of each imaging system is very specific and individual. This is true for any camera system but also for imaging systems like 2PM. In the case of 2PM the PSF is very large. It influences the resulting images heavily. For image analysis, in difference to other systems, deconvolution or to keep the process of imaging with the PSF in mind is important.

The theory of PSF for 2PM and LSM in general is well studied. We will introduce the most important terms and their relation to the PSF.

Numerical Aperture

The Numerical Aperture (NA) is one of the characteristic numbers of a microscope. The NA is described by the refractive index and the half angle of the cone of light. The cone of light is built by the diameter of the lense and the focal distance (see Figure 2.5 for visualization). The NA is a dimensionless number. It is given by:

$$\mathrm{NA} = \eta \sin \theta \tag{2.2}$$

where η is the refractive index and θ is half of the angle of the cone of light. The NA can be approximated by using directly the focal length and lense diameter:

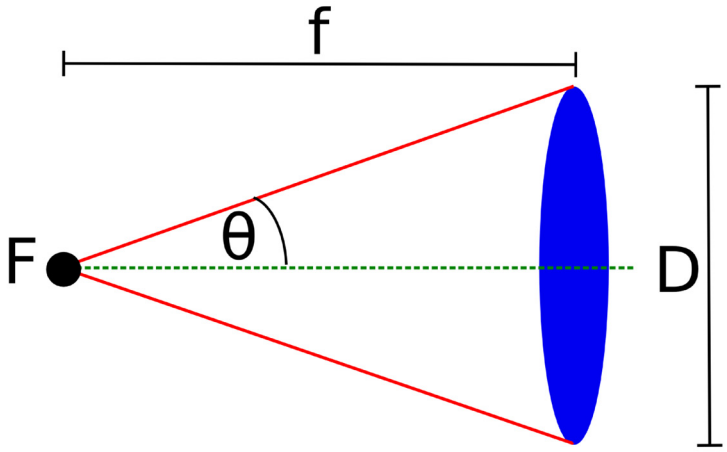


Figure 2.5.: Schema to visualize the cone of light. F is the focal point, f is the focal length and D is the diameter of the lense.

$$\mathrm{NA} = \eta \sin\theta = \eta \sin\left[\arctan\left(\frac{D}{2f}\right)\right] \approx \eta\frac{D}{2f} \tag{2.3}$$

Typical values for the NA of 2PM or LSCM are in the range of NA $= [0.8, 1.4]$. The larger the NA the smaller the diffraction limit gets.

Airy Disk

Airy disks describe the pattern generated when a laser uniformly illuminates a lense. This means that even a perfect imaging system is not able to represent a light point perfectly. In Figure 2.6 a 1-Dimensional (1D) airy disk is displayed. In 2D a circular structure with high intensity is located in the center. The central peak is surrounded by concentric bright rings. The intensity distribution h is computed by:

$$h(r) = \left(2\frac{J_1(r)}{r}\right)^2 \tag{2.4}$$

where r is the radial distance and J_1 is the first Bessel function. The airy disk helps to describe the diffraction limit. The diffraction limit is described by the Rayleigh Criterion and generally accepted. An imaging system is able to resolve two objects if the maximum of the first airy pattern is more far away than the first minimum of the second airy pattern.

Diffraction Limit

The diffraction limit describes if two objects just can be resolved by an imaging system. For classical light microscopy the theoretical diffraction limit was introduced by Ernst Abbe in 1873. It is given by:

$$d_{abbe} = \frac{\lambda}{2\eta\sin\theta} = \frac{0.5\lambda}{\mathrm{NA}} \tag{2.5}$$

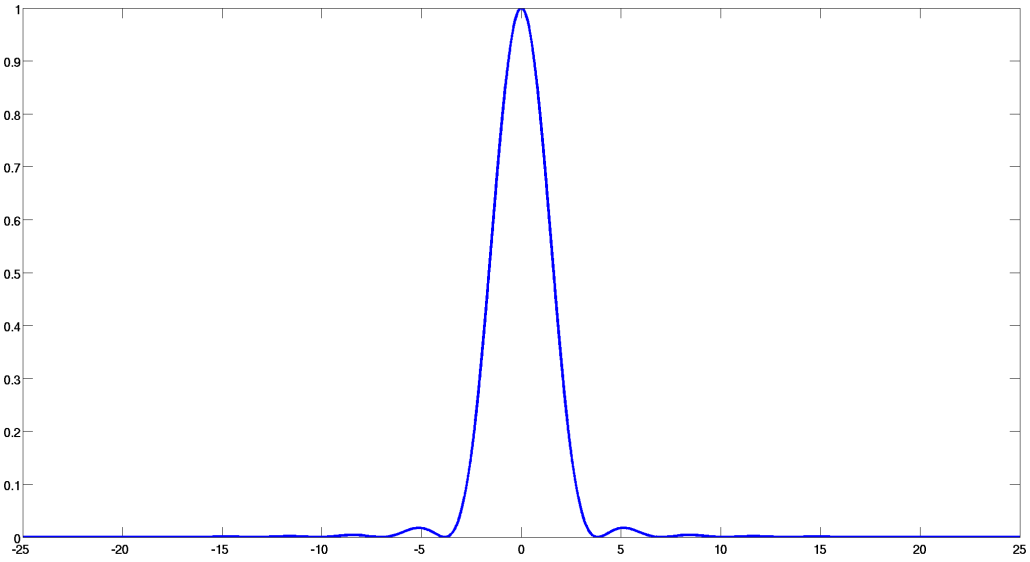


Figure 2.6.: Airy disks describe how a perfect imaging system would represent a light point. The airy disk is shown in one dimension.

This theoretical limit is not reached. We introduced the Rayleigh Criterion which describes this situation with the airy disk. The overlap defined by the Rayleigh Criterion corresponds to the first minimum circle and its distance is defined by:

$$d_{airy} = \frac{1.22\lambda}{2\eta \sin\theta} = \frac{1.22\lambda}{2\mathrm{NA}} = \frac{0.61\lambda}{\mathrm{NA}} \tag{2.6}$$

where λ is the wavelength.

The resolution limit of LSCM is reduced by a factor of about 30 % because of the narrower intensity PSF. We are concluding the reduced radius which is given by:

$$d'_{airy} = \frac{0.427\lambda}{\mathrm{NA}} \tag{2.7}$$

The reason for this effect is that the pinhole can be closed up to the first airy disk or also less (about 0.25 airy units). However, in practice this is often not achievable. Generally the signal to noise ratio becomes too worse. Especially for weakly fluorescent samples it is not possible to close the pinhole too much [15]. In theory the diffraction limit of 2PM would be about twice of that of LSCM. In practice it shows that the limits of LSCM are not reachable, but in contrast the one for 2PM are feasible. Therefore, both microscopy approaches have similar resolution limits [15]. Feasible limits for both approaches are estimated to be about 250 nm for commercial systems like mentioned by [15].

The measured PSF is computed different. The measured Full Width Half Maximum (FWHM) for 2PM is the squared illumination PSF in contrast to the non-squared for LSCM, see [63]. This holds also for the diffraction limit.

The z-resolution (axial) is computed different as the xy-resolution. Typically the axial resolution is about half of the xy-resolution. The typical values for the $\frac{1}{e}$-radius ω of 2PM are computed by the following formula as introduced in [63]:

$$\omega_{xy} = \begin{cases} \frac{0.320\lambda}{\sqrt{2}\mathrm{NA}} & \mathrm{NA} \leq 0.7 \\ \frac{0.325\lambda}{\sqrt{2}\mathrm{NA}^{0.91}} & \mathrm{NA} > 0.7 \end{cases} \tag{2.8}$$

$$\omega_z = \frac{0.532\lambda}{\sqrt{2}} \left[\frac{1}{\eta - \sqrt{\eta^2 - \mathrm{NA}^2}} \right] \tag{2.9}$$

The FWHM is computed by multiplying the $\frac{1}{e}$-radius ω with $2\sqrt{\ln 2}$.

Estimation of the Point Spread Function

The PSF can be estimated analytical with all the knowledge about the imaging system and physical laws or experimental by measuring in images. The important properties of the PSF were introduced above. Additionally, it is worth to mention the scattering and the fact that deeper in tissue not the same amount of light (from the laser) arrives. With all this knowledge an analytical PSF can be computed (what also is done in practice). However, the experience showed that experimentally measured PSF are more accurate and represent better the characteristic of each imaging system. We will introduce how both approaches work and discuss the advantages and disadvantages of both.

The theoretical PSF can be computed using the above described properties of the airy disk and other knowledge about the system and be approximated. It is common and validated that the PSF can be approximated quite accurate by a Gaussian distribution [59, 63]. Therefore, it is feasible to just model the PSF of 2PM by a 3D Gaussian distribution. From the FWHM directly the standard deviation σ of the Gaussian distribution can be computed for the xy- and z-direction. The mean value is $\mu = 0$. This approximation is simple and requires no complex computation. As extension to this it is also shown that for LSCM the theoretical PSF can be computed even more accurate by using e.g. information about the imaging location [38].

The experimentally estimated PSF has implicitly all the properties of the imaging system integrated. However, the result is depending on the imaging parameters and the way the experiment is conducted. The idea is to image an object smaller than the theoretical diffraction limit and thus to get from it the PSF. In practice this is made by using small latex beads which are then imaged. Each imaged bead results in its individual PSF. From all the imaged beads an averaged PSF is computed.

Both approaches have their advantages and disadvantages. The theoretical approach is much simpler because no imaging is required and the Gaussian distribution is easily computed. Furthermore, if the PSF is used under different environmental parameters afterwards, then it is not possible to have the correct experimentally estimated one. Therefore, it often makes sense to use the theoretical one for image analysis approaches.

In Section 3.4 Figure 3.3 shows a synthetic PSF approximated using a Gaussian distribution. In Figure 2.7 we show an example of an experimentally measured PSF by imaging beads. Different beads are visible in the volume rendering (Figure 2.7, *left*). In Figure 2.7, *middle*, the central xy-plane is shown and *right* we present the central slice through xz. The intensity distribution looks similar to a Gaussian distribution. However, it is much less smooth. Furthermore, the maximal intensity is not centered and the symmetry properties are not fulfilled. However, the figure shows slices through a single bead and not an averaging of multiple beads. Some of these artifacts could be removed by averaging. Then the estimated PSF is more like a Gaussian distribution.

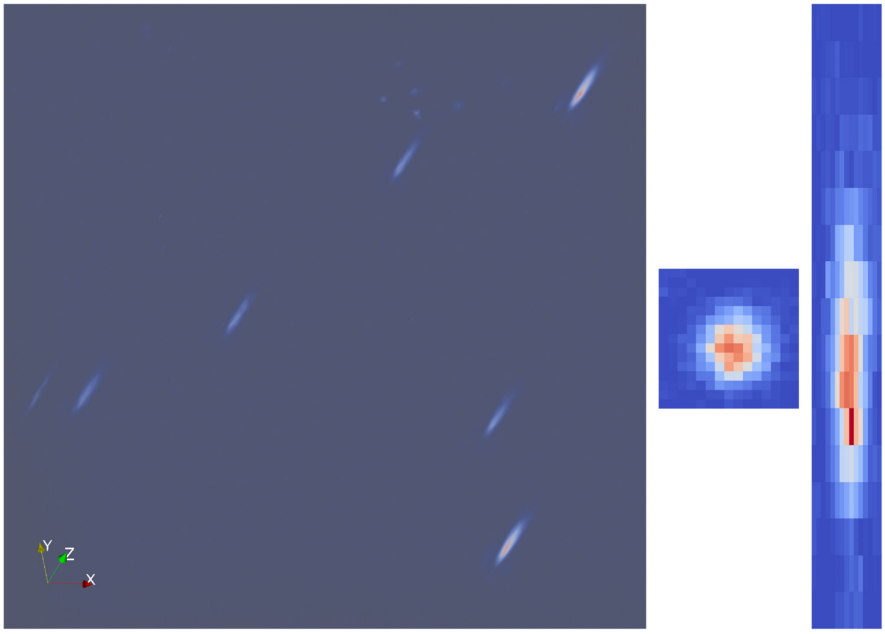


Figure 2.7.: *Left:* Volume rendering of imaged beads. *Middle:* Slice centered and parallel to xy-plane. *Right:* Slice centered and parallel to xz-plane. The asymmetry of the experimentally measured PSF is well visible.

2.3.4. Deconvolution

Deconvolution is the approach to invert the PSF. It can be described as finding and applying the inverted PSF. This removes the effects of the PSF. After deconvolution the real structures like without the PSF should be visible or at least better visible as before. Deconvolution can be divided into approaches integrating knowledge from the imaging process and blind deconvolution. Blind deconvolution approaches try to directly estimate the deconvolution from the image. The other deconvolution approaches use parameters from the imaging system.

Advantages of Using Deconvolution

In deconvolved images structures are visible in their original representation. Disturbing artifacts like the strong smoothing effect of the PSF are compensated. Edges and surfaces become better visible and related to that more high frequency properties are part of the image. In theory the real shape and intensity distribution becomes visible. Image analysis should become simpler and more robust because different objects are even better separated (see also the effect of the PSF presented in Figure 2.4).

Disadvantages of Using Deconvolution

In addition to the advantages the deconvolution has some disadvantages too. One main drawback is the computational cost of most approaches. Furthermore, most approaches require many (variable) microscope parameters to be known or a good estimation of the PSF is required to get good results. The use of simple deconvolution approaches mostly does not give good results. Well performing tools are difficult to use, often not

integrable in individual software and/or expensive. Also the quality of the deconvolved image must be controlled. Not always successful results are given. In cases like this a further analysis would not be possible or the image analysis approach must be working on deconvolved an non-deconvolved images. However, if the system is working on both kind of images it is probably not required to try to deconvolve the images at all. These disadvantages and the fact that deconvolution is a field at its own are also the main reason not to use deconvolution in our approach and this thesis.

Tools for 2-Photon Microscopy Deconvolution

The deconvolution of fluorescence images taken by different microscopes is quite complex. Furthermore, the effective implementation is time consuming. Therefore, there exist well known tools like Huygens Software [1]. Also most microscope manufacture have their own software for deconvolution. The advantage of software like Huygens is that it works independent from any microscope type. To outperform or perform similar well as well established deconvolution tools is not very feasible without investing a lot of time and to have a very good knowledge about deconvolution. Therefore, most projects having image analysis as main goal do no deconvolution or use one of these (professional) tools.

Conclusion

Most deconvolution approaches are optimized for LSCM data and do not perform as well on 2PM data. Furthermore, the PSF of 2PM smooths a bit more the data. Therefore, the deconvolution is more difficult in general. To process deconvolved and not deconvolved information is not feasible. It is required that all information in a pipeline is used in one or the other state but not mixed. Deconvolved data is not identical to data which has not applied a PSF at all. The deconvolution improves the image quality but does not find the ground truth before imaging. Otherwise the deconvolution would be able to find in fluorescence data the geometrical correct reconstruction (similar to reconstructions from EM data). All these different aspects are to be considered if one decides to use deconvolution or not. Just to apply any deconvolution approach without thinking about the effects for the planned pipeline is a dangerous idea. As stated in Equation (2.10), in addition to the convolution with the PSF also a noise term N changes the image [17]. In Equation (2.10) I is the final image and G the geometry respective image prior to imaging and its effect of the PSF. Even if the real PSF is known, a deconvolution is not straightforward because of the noise. Also the noise would have to be removed.

$$I = N\left[\mathrm{PSF} * G\right] \tag{2.10}$$

We do not use any deconvolution in our pipeline because of all these aspects of the deconvolution. The idea of deconvolution would not fit to our concept of using data from multiple modalities.

[1] Huygens Software, Scientific Volume Imaging (SVI) - `http://www.svi.nl/HuygensSoftware`

2.4. Serial Block-Face Scanning Electron Microscopy

Fluorescence images from 2PM and similar imaging techniques do not provide real information about the geometry or shape of the imaged object. There exists no algorithm (even independent of the cost) which would be able to reconstruct from fluorescence image a shape identical to the real geometry in all details. In difference to this, EM data is very high resolved in all axis. Detailed structures of the imaged object are visible. The drawback is, that EM is a (financial) expensive and time consuming technique. Furthermore, EM only can image dead samples. However, exactly these differences between fluorescence images and EM data makes it interesting to use both information together. If from the same structure both images are available, this is called correlative data or Correlative Light Electron Microscopy (CLEM). We will use the advantages of both imaging approaches as well as CLEM. Therefore, we introduce fundamental knowledge about EM and SBFS-EM in particular.

2.4.1. Electron Microscopy and Data

EM data are high resolved intensity images. Biological structures like neurons are fixed and prepared for EM. It is feasible to generate isotropic images with a higher resolution than in fluorescence imaging (e.g. 2PM). In current EM resolutions up to about 10 nm are feasible. In high resolved images it is feasible to see the real geometry from tiny structures like spines. A typical approach for imaging biological structures is to use Transmission Electron Microscopy (TEM). In approaches like TEM one 2D slice after the other one is generated (using a microtome) and imaged. These slices are independent each one from the others. The slices cannot directly be composed together as a 3D image (slice by slice). It is required that the slices are aligned to their neighboring slices and also corrected for distortion effects. Furthermore, stitching is required if larger regions are imaged. This is quite difficult and requires a larger effort (see e.g. [33]). However, EM is able to resolve the required structural information of dendrite and spines. To overcome the issue of distortion and stitching the use of SBFS-EM is recommended. SBFS-EM will generate similar data like TEM. Moreover, the issues of alignment, distortion and so on are already solved.

2.4.2. Serial Block-Face Scanning Electron Microscopy - Technology

The preparation of tissues for SBFS-EM is similar to the preparation for other EM techniques (e.g. TEM). The tissue is fixed and does not live anymore. Time series imaging is not possible because of the fixation. The fixed tissue sample is cropped and inserted to the microscope. The surface of the block is sampled with an electron beam and an image generated. This image is the first slice. Then an ultrathin slice (50 nm or less) of the fixed tissue block is planed down with a diamond knife. The new surface is again sampled with the electron beam and generates the next image. All the 2D images together are the slices of the final resulting 3D image. Figure 2.8 shows a schematic diagram of the imaging process and the mechanical design (Computer Aided Design (CAD)) of the in-chamber microtome.

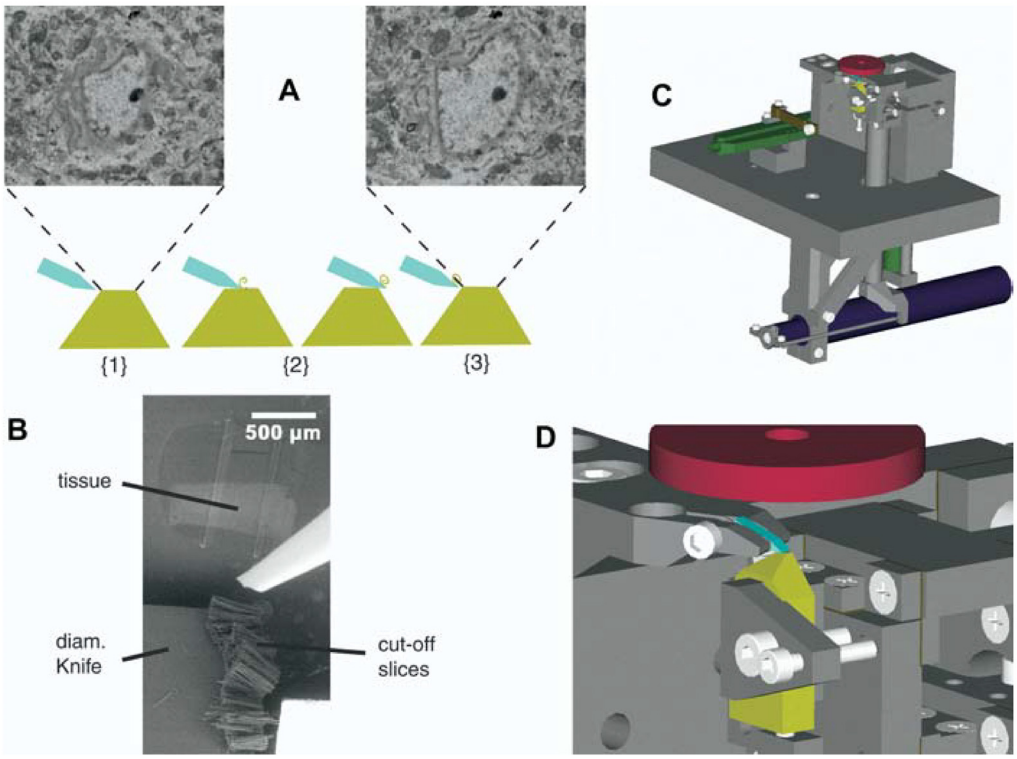


Figure 2.8.: Schematic diagram of working principle of the SBFS-EM. *A:* Process of imaging {*1*}, cutting of a ultrathin slice {*2*} and imaging of the next slice {*3*}. *B:* The cut-off ultrathin slices pile on the diamond knife. *C and D:* Mechanical design of SBFS-EM as CAD schema. Figure from [19].

2.4.3. Advantages of SBFS-EM

In principle SBFS-EM generates images in a similar way like other EM techniques. However, the processing of the generated slice images to build a 3D volume is much simpler or not required at all. Depending on the size of the imaged region no stitching, registration nor alignment is required. Generally the registration from the SBFS-EM is better than the resolution [19]. The 2D image slices just can be composed together to a 3D image. This makes the use of SBFS-EM simple and fast. It becomes feasible to analyze the 3D volume without additional image correction. Therefore, SBFS-EM allows to focus on image analysis. Image correction like distortion correction and alignment are not required. The lack of distortion correction and alignment is clearly one of the most important advantages of SBFS-EM over TEM which requires these steps (see e.g. [33] about the importance of distortion correction of TEM data).

2.5. Statistical Models

A basic concept of current image analysis is to learn from data. It is tried to learn the properties (or features) of an image directly from training data. Then for a test image a prediction or classification is made. The idea is that due to enough representative data a system is capable to classify if something is similar (and how similar) to the learned class. Centrally important for approaches like this are statistical models. They try to model the object class. However, instead to manually build a model describing the object class the model is composed from data and compresses the learned information thanks to statistical methods into a statistical model. A statistical model spans the space of samples in which then can be worked.

The essential properties of statistical models and differences to other models are introduced in Section 2.5.1. In Section 2.5.2 we will introduce fields in which statistical models are used. Finally, we introduce the basics from Principal Component Analysis (PCA) and well known approaches using PCA in Section 2.5.3. In contrast to the theory the application of the PCA to compute our models and integration in our pipeline is introduced in Section 5.3.

2.5.1. Basics

Here we discuss the basics of statistical models with respect to image analysis. However, many of the discussed subjects can be directly or indirectly applied to other subjects too.

Every image can be seen as a number of measurements (pixels). Therefore, the work with images, vertices or any other kind of measurement can be identically traded with respect to statistical models. The basic idea behind statistical models is that given an example some properties of this example are important and descriptive for the object class and other ones are not or less. If all the variables of the examples are imagined in a high dimensional space every sample has its individual location. They span together a space (like a dense cloud). It is feasible to move inside this spanned space and staying within the object class. Moving outside the point cloud corresponds to move away from the class. The effect how examples are distributed within space is schematically visualized in Figure 2.9. Every *red dot* symbolizes an example in the high dimensional space. For visualization purpose only a schematic diagram in two dimensions is shown.

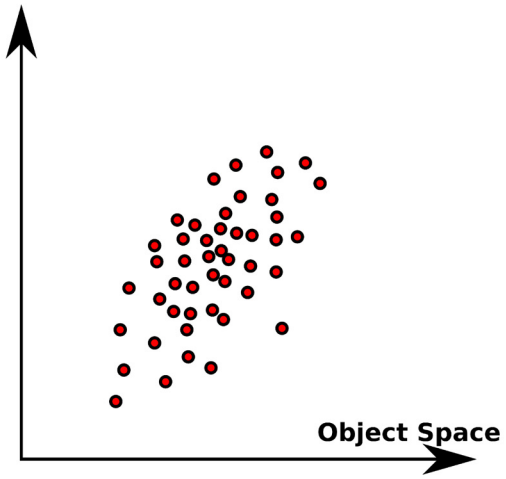


Figure 2.9.: All examples can be visualized as points in a high dimensional space. Within the class they build a kind of cluster. It is feasible to interpolate within the examples to get further objects of the class.

In 1995 the Active Shape Model (ASM) marks a milestone of statistical (shape) models in 2D [14]. This work was then extended to Active Appearance Model (AAM) in 1998 [20]. In 1999, finally, statistical shape models for 3D were introduced as 3D Morphable Model (3DMM) [5]. ASM are using the concept of snakes. However, these are constrained to the point distribution model learned earlier. AAM is an extension to ASM and adds the information about appearance within the object. The next milestone was to learn 3D models which was done by the 3DMM. The principal idea to learn from training data how information can be distributed (point distribution) is similar. However, in the 3DMM everything is done in 3D. There is neither use nor relation to the concept of snakes. All these models have in common that the compressed knowledge is based on training data and statistical information gathered from the training data. This means that some variable build a distribution function that can be used to model an object or class. Instead of a single variable this core idea is then applied to a high dimensional representation. A common approach to represent all the different variables is to use PCA. PCA transforms the variable into a new space in which new variables are used. These variables can be sorted by their importance as well as the variance of the training data they capture. The basics of PCA will be introduced in Section 2.5.3.

2.5.2. Application Areas

Statistical models and statistical shape models have a large field of application in computer vision. In industry applications automated production lines can perform better or are more flexible using models. Similar to that industry robots even start to work outside of a fixed environment and are used e.g. in pipelines or drains to make a diagnosis about the maintenance state. First car assistance systems using statistical models are integrated in purchasable cars. In the medical domain statistical models support to segment, label (atlas labeling) or detect structures in medical images (e.g. x-ray, CT, MRI or other image data). This research field is heavily studied from industry and academia. The new technologies open the possibility of fast and reliable diagnosis and

further support. Current research takes also place in the field of building models of individual body parts or bones, e.g. [1]. Models like this clearly enable to detect abnormalities of body parts or to predict the original shape of broken bones. Moreover, even during operations it is feasible to use information from models for diagnosis. A step further go automated or semi-automated surgeries. They use for instance augmented reality [18] or even surgery robots. In surgeries objects must be recognized reliably under different illumination, orientation and individuals. Furthermore, also within an object class the shape can change because of breathing, heart beat or muscular contraction. Therefore, approaches without the flexibility of a statistical model are less suitable and have most likely worse performance. In the field of entertainment applications like games, social media, movies and similar things are to be mentioned. In all these entertainment fields it is tried to integrate the human in some aspects. This is done by enabling persons to interact in a natural way or by integrating the appearance of people for different uses (e.g. facial avatars). In both situations working systems for any human or human part (e.g. arm and legs) in a unknown and most probably dynamic environment (individual room/location and illumination) are required. Statistical models have a very good performance in solving problems like this because they generalize very well and are easily adapted to the individual person.

Statistical models serve in all these fields for tasks like detection, segmentation, tracking, abnormality prediction, augmentation and others. In all these tasks the power of a statistical model is to overcome errors which are caused e.g. by noise or occlusion. This is because the models learned what kind of data is representative and the knowledge is integrated into the model.

2.5.3. Principal Component Analysis

The key for statistical (shape) models is to represent a large and normally high dimensional data space (the training data) in a compact representation. This means that the information in the data is reduced to the most important information. Statistical shape models share properties with dimensionality reduction approaches. The most often used approach for statistical models and dimensionality reduction is the PCA. The 3DMM [5] and also older approaches like Eigenfaces [53] are based on PCA. The approach of PCA is a mathematical procedure to transform the high dimensional information into an orthogonal space with linear uncorrelated variables. These variables are the so called Principal Components (PCs). The PCs are sorted by their importance and the first PC has the largest statistical variance. We will introduce here the basics of the PCA and some background. In Section 5.3 the idea of the PCA is then used to compute a dendrite intensity and spine probability model.

A given set of training data s_i (with dimension m) can be written in a matrix of size $m \times n$. The mean example μ from every example s_i is subtracted. In the matrix the rows are the different variables and the columns the examples. The matrix X is then given by:

$$X = \begin{bmatrix} s_1 - \mu \dots s_n - \mu \end{bmatrix} \tag{2.11}$$

The mean μ is computed by:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{2.12}$$

By using the Singular Value Decomposition (SVD) we can decompose matrix X into

$$X = UDV^T \tag{2.13}$$

Matrix U is the matrix with the eigenvectors of the covariance matrix $\Sigma = \frac{1}{n}XX^T$. D is a diagonal matrix with the so called singular values which are the standard deviation captured by the corresponding eigenvectors in U. The columns of matrix U and V^T are unitary and build each of them an orthonormal vector which build the basis vectors. Furthermore, it holds that $UU^T = I$ and $VV^T = I$ where I is the identity matrix. Because of that it holds that $U^{-1} = U^T$ and $(V^T)^{-1} = V$. Now the inverted matrix X^{-1} can be written thanks to the SVD by:

$$X^{-1} = (UDV^T)^{-1} = (V^T)^{-1}D^{-1}U^{-1} = VD^{-1}U^T \tag{2.14}$$

The inverted diagonal matrix D^{-1} is given by the diagonal matrix with values $\frac{1}{d_i}$ on the diagonal where d_i are the diagonal elements from D.

A sample s is written as

$$s = \mu + U\alpha \tag{2.15}$$

where α are the so called PCA coefficients. Instead of using all dimensionality a reduction to the k-th first PCs is feasible and in practice done. This reduction corresponds also to the dimensionality reduction which is used in compression approaches. A new sample can be projected into the PCA space. The PCA coefficients are found by:

$$\alpha = U^T(s - \mu) \tag{2.16}$$

The probability of observing a sample s can be written as

$$P(\alpha|s) = \frac{1}{\sqrt{(2\pi)^k\,|\Sigma|}} \exp\left(-\frac{1}{2}(s-\mu)^T\Sigma^{-1}(s-\mu)\right) \tag{2.17}$$

which is the density distribution of the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$. The probability $P(\alpha|s)$ is the posterior probability of the latent variable α. The fact that the probability of observing an example can be computed is quite important. Thanks to this property it becomes simple to compare samples and we can make conclusions from statistical models by probabilistic reasoning.

In Equation (2.15) we showed how any sample of the class can be represented by the PCs. In Equation (2.16) a set of coefficients α describes the sample in a compact representation. This fact enables to represent samples from the training data and other samples. If not all PCs (less than number of variables) are used a compression takes place. If the number of training samples is less than number of variables this compression (dimension reduction) is anyway given. This is mostly true for models from images or 3D meshes because the number of variables is large (number of pixels) in the range of hundreds or thousands. However, the number of examples is normally

limited and smaller. For example the Basel Face Model (BFM) uses only 200 examples for training, but the number of variables is given by the meshes with 53490 vertices [41]. This shows how statistical (shape) models can represent very high dimensional information in a compact way. The importance of the different PCs is well visualized by showing how much variability each one holds. This is visualized by the variance or the standard deviation they hold which is directly visible in the diagonal elements d_i of matrix D. The diagonal elements are the standard deviation of each PC:

$$\sigma_i = \frac{d_i}{\sqrt{n}} \tag{2.18}$$

The smaller the standard deviation σ_i of a PC is the less important it is and can be skipped. We can clip the PCs after a desired value because the PCs are ordered by their eigenvalues. This gives a compact representation. In Section 5.4, Figure 5.4, an example of the decreasing standard deviations is shown. PCs with smaller standard deviation start to hold more noise and less class information. This means that the data can be described in a compacter way than using all the variables. Furthermore, the faster the values σ_i decrease the better the model is trained and the model better represents the information of the training data.

The decision of number of PCs is a difficult but important task. It is not a priori clear how many components are required or useful. Furthermore, too many components can mean to model noise. This is not wanted at all. In 1954 the original Kaiser-Guttmann criteria was introduced [23] . This criteria requires the eigenvalues to be larger than 1. For different reasons this does not always hold (see [31]). Later the so called Scree Test followed [8]. In the Scree Test the eigenvalues are plotted against the number of components. Then the point where the curve has a sharp bend is localized. This whole test is done manually (visually) by an expert. The test is very subjective because multiple sharp bends are possible. Horn extended the Scree Test to plot a curve with random data. The intersection between both lines (structured and unstructured data) is the point to stop [30]. Another approach is to use components up to a proportion of the total variance. E.g. use all components until 95 % of the total variance is accumulated. This method is supported by statisticians, but even some advise against it (see in [31]). However, it is a simple and statistical based criteria. There exist even more stopping criteria which are summarized in [31]. The number of components is a very important decision while constructing statistical shape models. At least the visual interpretation and decision with the Scree Test should be suggested. Normally, this gives for the experienced user a good hint where to stop. However, also a comparison between the different criteria and how the statistical model performs with the different number of PCs is recommendable. In Section 5.4 we show an example of the proportion of total variance in Figure 5.5. The function for plotting the proportion of total variance is

$$f(x) = \frac{\sum_{i=1}^{x} \sigma_i^2}{\sum_{i=1}^{K} \sigma_i^2} \tag{2.19}$$

where K is the total number of components and σ_i are the eigenvalues from the diagonal matrix D.

In a short summary PCA can be described as a compression method and a dimensionality reduction approach which enables the computation of statistical models and builds a good basis for probabilistic modeling.

3. Digitally Reconstructed Fluorescence Images

Synthetic data and synthetic images are often used to train image analysis systems. Synthetic data represents the original data but with known parameters of all parts of the image generation. Therefore, ground truth data can be generated and all details of the image generation process are under supervision. However, synthetic data only approximates the real image generation process as well as modeled. Therefore, it is not guaranteed that algorithms using synthetic data will have the same performance if tested with real data. The use of synthetic data is often rated as misleading. In many approaches manually labeled data is used instead of synthetic data. It is infeasible to manually label all spines in fluorescence images of dendrite pieces. In many situations even experts are not able to decide if spines are located or not at a certain location. Furthermore, experts are only able to decide binary labels and cannot specify how much intensity is from spines and from dendrite. Therefore, we take the advantage of synthetic data that gives us full control on all information. It is feasible to compute per pixel how much intensity is contributed from spines by using the concept of information transfer. In Section 3.1 we discuss in more detail the usage of synthetic data in image analysis. In Section 3.2 we introduce what assumptions and prerequisites are given to compute synthetic fluorescence images of pieces of dendrite. The geometrical correct shape and the Point Spread Function (PSF) are discussed in Section 3.3 and Section 3.4 respectively. The details about the computation are given in Section 3.5. Finally, in Section 3.6 we present the information transfer between the electron and fluorescence microscopy domain. We call our concept to compute synthetic fluorescence images Digitally Reconstructed Fluorescence Images (DRFIs).

3.1. Synthetic Data in Image Analysis

Synthetic data has many advantages over real data. Therefore, many image analysis systems use synthetic data. We will discuss the basic idea of synthetic data, advantages, disadvantages and illustrate some examples which work with synthetic data.

Basics of Synthetic Data

Synthetic data enables computer scientists to work in a controlled environment. The data is in some meaning simpler because it is a controlled approximation of the real data. Furthermore, the approximation can be done on different levels. Basically, real data is too complex (too high variability) and synthetic data should close this gap to be able to experiment with the data in a controlled way. Synthetic data and its generation can also help to understand the image generation process. This can be helpful for scientists unexperienced with the data. Synthetic data are samples from a model (which is an approximation). A model with parameters is used to generate data.

Advantage	Description
Simplification	Synthetic data is only an approximation and thus a simplification. Therefore, the core issues can be studied better and it is often easier to find a solution.
Known parameters	Changes in data and analysis can be studied well because the parameters to generate data are known. The relation between parameters and results can be studied. It can be evaluated for which parameters improvements are required.
Flexible resolution	Many models enable flexible resolution and sampling of the synthetic data. Often higher resolution than with the imaging system in focus are possible.
Number of samples	Depending on the kind of data, it is quite difficult to have or get enough data (cost, privacy, time consuming imaging or other reasons). Synthetic data can be generated most likely in any required number of samples.
Known segmentation	For synthetic data the foreground and background are known. Even the segmentation of parts is known or can be easily found by experts.

Table 3.1.: Overview of advantages to use synthetic data in image analysis. The focus is on the generation and use of fluorescence images.

The number of parameters describes the complexity of the model. In some situations it is even possible and/or helpful to hold some parameters (e.g. representing physical properties of the microscope). Furthermore, the model has not all real parameters integrated because it is an approximation. Synthetic image data can be computed in any sampling rate in most cases. This enables to generate images with almost any required sampling rate. However, the generation of synthetic data is often optimized to a range of sampling rate in which the approximation of the reality works well. Related to this the level of details that synthetic data provides can vary. Depending on the application it makes sense to have different level of details. In summary a number of parameters and algorithms generate synthetic data that is as similar as possible to real data but with a fully known generation pipeline.

Advantages

Synthetic data provides different advantages for image analysis approaches. Therefore, in many situations image analysis approaches are designed, trained, learned, tested and/or evaluated with synthetic data. In Table 3.1 we list the most important advantages for the use of synthetic data in image analysis. Even further, not listed, advantages are possible depending on the kind of data and application.

Disadvantages

Like every approach or technique also synthetic data has together with advantages some disadvantages. In Table 3.2 we list the most important ones.

Disadvantage	Description
Simplification	Simplification is an advantage and real disadvantage at the same time. The real problem to be solved can be much more difficult. This is because synthetic data is an approximation of real data. Sometimes algorithms or system developed with synthetic data perform very well on synthetic data but fail with real data. There is no guarantee that a system works robust on both data types. The issue of simplification is also in general the main criticism.
Time consuming design	In most cases it is quite time consuming to design and to develop a system that creates synthetic data which is competitive to real data.

Table 3.2.: Overview of disadvantages to use synthetic data in image analysis. The focus lies on the generation and use of fluorescence images.

The most often criticism is clearly that synthetic data is not representing reality and that systems using synthetic data have a weak performance on real data. We also believe that only testing with synthetic data is a bad idea. In most situations systems trained with real data perform better than system using synthetic data. However, in some situations not enough real data is available or the real data does not provide directly all information required (e.g. segmentation). In the case of spine detection the second issue is given which leads to the problem of expert labeling which is difficult.

Examples

The common reason to use synthetic data in different approaches is the complexity of collecting and/or generating data. Examples using synthetic data during training are e.g. in handwriting recognition system [54] or for face detection [25]. Both approaches try to increase their amount of training data by using synthetic data. The need of synthetic data is also given in biological data. E.g. in [22] a framework to simulate biological cells for fluorescence imaging is presented. The goal is to have enough and reliable data. Data like this can serve as ground truth data because no error-prone expert labeling is required. Further fields to use synthetic data are e.g. registration approaches. In registration methods, often artificial data is used to test and validate the method. The known transformations motivate the use of synthetic data.

Synthetic data is used in many fields related to computer science and not just in image analysis. E.g. business systems are tested using synthetic data. Signal processing in general (which includes image analysis) uses synthetic data too. The power of synthetic data is the unlimited amount of data that can be generated and all additional information which is available with synthetic data that makes it usable as ground truth data.

3.2. Concept

The basic idea is to compute DRFIs from the real shape of the objects (pieces of dendrites). This ensures to know what exactly is approximated in the DRFI. The real shape of dendrites is available in Electron Microscopy (EM) data in a high resolution. Current approaches (e.g. Serial Block-Face Scanning Electron Microscopy (SBFS-EM), see Section 2.4) enable to image fixed tissue in high resolution as 3-Dimensional (3D) images. However, the reconstruction of dendrite pieces in EM data is itself a difficult task. Therefore, we use special prepared samples to retrieve the shape of dendrites as described in Section 3.3. The geometrical shape is the basis to compute DRFIs. Dendrites enclose many structures like mitochondria. It is important to take into account where fluorescence in dendritic structures is present. Otherwise no correct representation of the fluorescence in the synthetic data is given. In our approach we expect the volume marking fluorescence to be everywhere within the space enclosed by the membrane apart from enclosed organelles like mitochondria. Mitochondria can be quite large (diameter up to 1.5 µm) and exist numerously in neurons and dendrites. It is possible that mitochondria fill almost completely regions of the dendrite with respect to the diameter. This has a huge impact on the fluorescence images because the intensity in regions like these decreases almost to zero.

The distribution of fluorescent dye or protein is expected to be homogeneous within the dendrite. We do not expect to have an accumulation of fluorescence in certain regions like spines or spine necks. This assumption is quite important because local accumulations of fluorescence would change the intensity distribution in fluorescence images completely. It is expected that fluorescent dye filled dendrites and genetically modified ones have both a homogeneous distribution of the fluorescence. It is not known that some accumulation takes place. This enables us to model a homogeneous distribution of fluorescence in dendrites (excluding enclosed organelles). This task is quite straightforward and much simpler than modeling specific local accumulations.

3.3. Dendrite Shape from Serial Block-Face Scanning Electron Microscopy

Dendrites have a quite complex and variable shape. The shape of dendrite is given by a flexible border, the membrane. Moreover, the surrounding is not empty but also heavily filled by other structures like dendrites, axons and cell bodies. Therefore, the shape of dendrites can be deformed by surrounding structures. Furthermore, small protrusions (spines) make it even more difficult to describe the shape of dendrites. It is not feasible to model the real shape of dendrites without prior data. Therefore, we take the real shape of dendrites as basis to compute DRFIs. The dendrite shape can be reconstructed from SBFS-EM data.

In Section 2.4 we described the principles of SBFS-EM with its advantages. In SBFS-EM images of neurons all structures are visible and encoded in intensities (see Figure 3.2, *left*). Therefore, it is quite difficult to automatically reconstruct dendrites from SBFS-EM images (and also from other electron microscopy techniques). The manual reconstruction of dendrites is very time consuming and also error-prone. The membrane has the size of only some pixels and also experts happen to make errors in the manual reconstruction. If not all dendrites are required to be reconstructed, then it is possible

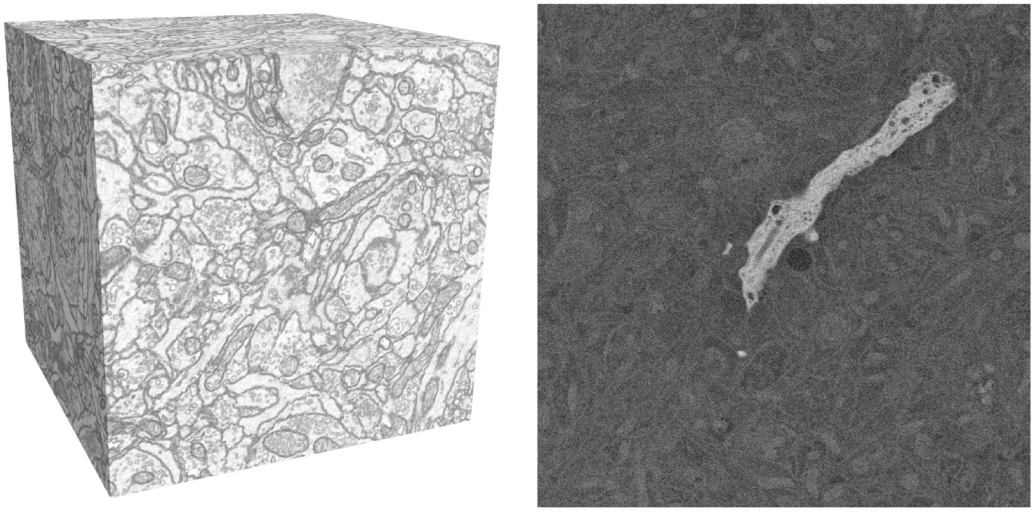


Figure 3.1.: *Left:* Raw data from SBFS-EM without labeling. *Right:* In the raw data from SBFS-EM of a labeled neuron the marked object is clearly distinguishable from the background. In the slice the labeled dendrite (in *white*) has a high contrast to all the other dendrites and can be reconstructed.

to mark a neuron (and its dendrites) during sample preparation. This simplifies the reconstruction process. Figure 3.1, *right*, shows a slice of the raw data of SBFS-EM data with a labeled neuron. The labeled dendrite is distinguishable from the background and can be easily reconstructed manually or by a semi-automatic process. Also visible are enclosed organelles. Especially large mitochondria (e.g. Figure 3.1, *right*, at the left part of the dendrite) are important for the computation of DRFIs.

A semi-automatic approach to reconstruct dendrites is to threshold the raw data by a manual selected threshold. Afterwards, enclosed small objects (small organelles and artifacts) can be closed and the larger enclosed objects (mitochondria) kept. However, this approach is still manually guided and error-prone because the division of the enclosed objects into removable ones and mitochondria is manual and mainly decided by the size. Furthermore, the surface can be very bumpy because of noise and enclosed objects at the surface. This makes a smooth representation of the dendrite impossible. Finally, a huge amount of data has to be controlled after the semi-automatic reconstruction. Often the raw data is required to decide if the reconstruction is correct. Therefore, we prefer to do a completely manual reconstruction of the complete volume or dendrite. This is feasible because only a small number (2-4) of dendrite pieces is required for our further approaches of spine detection and segmentation.

Figure 3.2 shows the different segmentation steps of SBFS-EM data with a labeled neuron for the semi-automatic approach. Figure 3.2, *left*, is the raw data with one cell (and dendrite) labeled. In the *middle* the binary result of applying a manually selected threshold value is shown. *Right*, the result after manual optimization is displayed. All holes except (large) mitochondria were filled. The concave bays at the border still exist. Without comparison to the raw data and neighboring slices it is not feasible to decide which concave bays should be closed.

Results of the finally used reconstructions are displayed in Appendix A.4, Figure A.4.

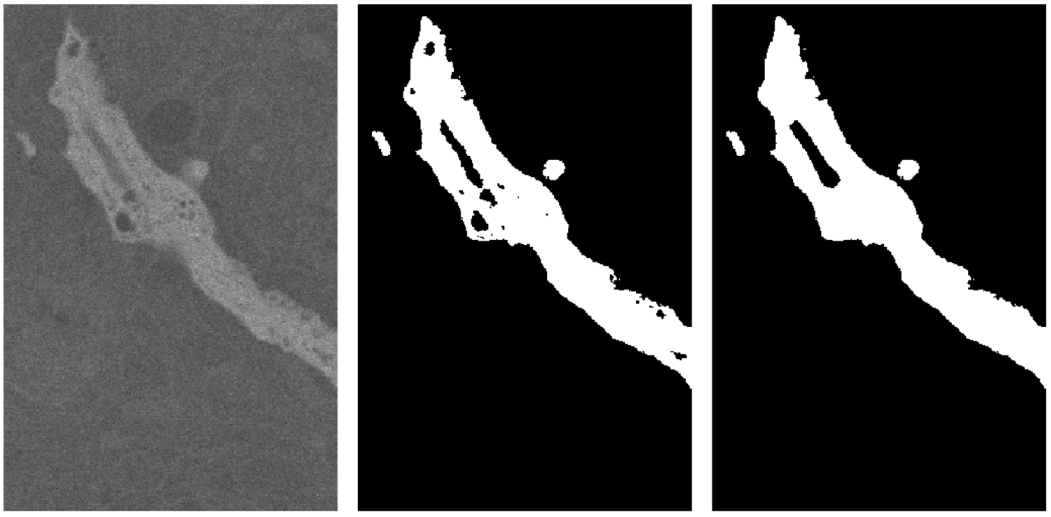


Figure 3.2.: *Left:* Raw data SBFS-EM slice. *Middle:* Thresholded slice using a manually picked threshold. Many holes exist and the surface is very bumpy. *Right:* An automatic hole filling approach and subtraction of enclosed mitochondria generates the reconstruction. However, the bumpy surface must be corrected manually.

3.4. Point Spread Function

The PSF describes how a single object point is mapped to the image intensities. We discussed the PSF and its theoretical background in Section 2.3.3. The differences between the PSF of Laser-Scanning Confocal Microscopy (LSCM) and 2-Photon Microscopy (2PM) were discussed too. Furthermore, we introduced a good estimation for the PSF of 2PM. In our approach we use a Gaussian approximation of the PSF. The standard deviation of the Gaussian approximation of the PSF is given by the Full Width Half Maximum (FWHM), see also Section 2.3.3. The different parameters (e.g. wavelength) enable to generate different PSFs. In Figure 3.3 we show a typical PSF that we use. The main effect of changing the parameters (Numerical Aperture (NA) and wave length) is the change in size of the PSF. However, the basis of the distribution is always quite similar because the parameters do not have a huge value range (e.g. NA $= [0.8, 1.4]$). Figure 3.3, *left*, shows a cut parallel to xy at the center of the PSF ($z = 0$). *Right*, we show the same image for a cut through the xz-plane. The huge elongation along the z-direction is visible. This elongation is one of the main characteristics of the PSF.

3.5. Computation

DRFIs map the process of computing from geometrical correct shapes and the PSF the fluorescence intensities or fluorescence images. The geometrical shape is given by reconstructions from SBFS-EM data. In Section 2.3.3 we introduced the PSF of 2-Photon Microscopy and fluorescence images in general. The correct PSF can be found experimentally or be approximated by a Gaussian distribution (see Section 3.4). For the computation there is no difference if the experimental measured PSF or an

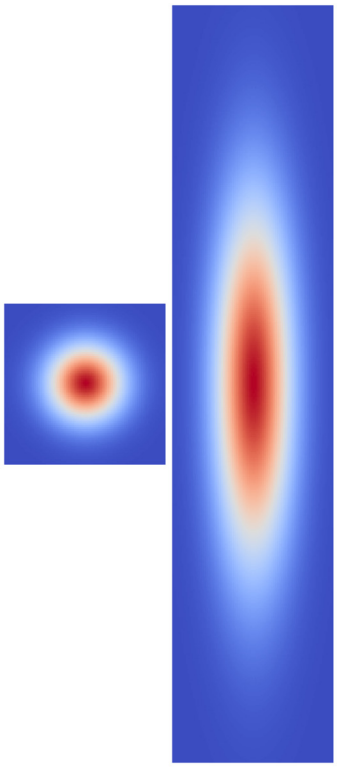


Figure 3.3.: *Left:* 2D slice of the PSF through xy. *Right:* Same representation of synthetic PSF for a cut through xz. The used parameters are NA = 0.9, $\lambda = 810$ and $\eta = 1.42$.

approximated one is used. Both version can be used as kernel.

The PSF describes for every image point or point source the response of the imaging system. The reconstruction (binary image) of the dendrite are the (sampled) image points of the object. This means that the computation of DRFIs is the PSF for every image point (pixel) of the object. Therefore, the convolution of the binary image with the PSF as kernel corresponds to the DRFI. It must be ensured that both images are sampled at the same rate. If $I_{x,y,z}$ is the binary image of the reconstruction and $\mathrm{PSF}_{i,k,l}$ is the point spread function the DRFI is computed by:

$$\mathrm{DRFI}(I_{x,y,z}, \mathrm{PSF}_{i,k,l}) = I_{x,y,z} * \mathrm{PSF}_{i,k,l} \tag{3.1}$$

The approximated PSF can be sampled at any rate. The binary image of the reconstruction is sampled at the rate of the SBFS-EM raw data (or less frequent). Because of the very high sampling rate of SBFS-EM (about 0.02 µm in horizontal direction (x-, y-axis) and 0.05 µm in vertical direction (z-axis)) large binary images are given. For small pieces of dendrite (about 10 µm) and the given sampling rate images have a size of about 500 x 500 x 200 voxels. Convolution is computationally expensive. Therefore, it makes sense to compute the DRFI of large images in the Fourier domain using Discrete Fourier Transformation (DFT) or Fast Fourier Transformation (FFT).

3.5.1. Fast Fourier Transformation and Convolution

The convolution of an image with a kernel is in the Fourier domain an element-wise multiplication of the transformed image and the transformed kernel. There exist many libraries which do a transformation into the Fourier domain by the DFT, e.g. [2]FFTW or the implementation in [3]VNL which is part of [4]VXL. Using the Fourier transformation Equation (3.1) becomes the much less computationally expensive form in the Fourier domain:

$$\mathrm{DRFI}\left(I_{x,y,z}, \mathrm{PSF}_{i,k,l}\right) = \mathscr{F}^{-1}\{\mathscr{F}(I_{x,y,z}) \circ \mathscr{F}(\mathrm{PSF}_{i,k,l})\} \tag{3.2}$$

However, it is required to do the Fourier transformation and the inverse Fourier transformation after the multiplication.

Table 3.3 shows a comparison of computation time for the convolution in the space domain versus the same result using FFT. The first 4 measurements are values from [36]. The fifth measurement is for one typical piece of dendrite we use later on. The image size was $900 \times 900 \times 200$ and the kernel size $59 \times 59 \times 105$. We canceled the direct computation after 4 h. In our computation we used an Intel(R) Xeon(R) CPU X5660 at 2.80 GHz with 96 GB memory and running Ubuntu 12.04 LTS. It clearly shows the advantage of FFT based convolution over direct convolution. Further comparison about the advantage of FFT based convolution over direct convolution can be found in [36]. They show that with a kernel size larger than 20 pixels all compared FFT based convolution approaches outperform the direct convolution. Therefore, the time advantage of using FFT based convolution is huge.

For further information about FFT (and DFT and Fourier Transformation) we refer the reader to the corresponding literature.

[2]FFTW, Fastest Fourier Transformation in the West, `http://www.fftw.org`
[3]VNL, Vision-Numerics-Library in VXL
[4]VXL, Vision-*Something*-Libraries, `http://vxl.sourceforge.net`

	Direct Conv. (s)	FFT Conv. with FFTW (s)
[36], kernel=10	< 0.05	< 0.05
[36], kernel=20	~ 0.05	< 0.05
[36], kernel=30	~ 0.1	< 0.05
[36], kernel=40	> 0.25	< 0.05
Typical Dendrite Piece	$\gg 4h$	410

Table 3.3.: Comparison of computation time for doing a classical convolution or the transformation into the Fourier domain followed by a multiplication and the inverse Fourier transformation.

3.6. Information Transfer

The concept of DRFIs enables the information transfer from EM data or reconstructions from SBFS-EM data to the fluorescence image domain. Due to the information transfer it is feasible to have similar information from one domain in the other domain. This transfer enables to integrate shape structure (given by reconstructions) and the knowledge about the segmentation into fluorescence images. In the case of spines the shape reconstruction in the SBFS-EM domain can be transferred to a representation (probability maps) in the fluorescence image domain. Due to the the information transfer more knowledge than just a binary expert labeling becomes available in the fluorescence image domain. Furthermore, it would not be possible to label information like this (probabilities) directly in the fluorescence images. The transferred information can be used for learning but also for validation. A validation of algorithms within the fluorescence image domain is difficult to almost not possible while it is simple to validate the labels in reconstructions from SBFS-EM data.

With the information transfer we transfer segmentations of EM reconstructions to their corresponding representation in the fluorescence image domain. The following steps need to be fulfilled for the information transfer:

- Object reconstruction (dendrite) from SBFS-EM data.
- Labeling of structure in interest (e.g. spines).
- DRFI of the full object (dendrite including spines).
- DRFI of the labeled objects (spines).
- Computation of the probability map.

Object Reconstruction

The reconstructed object is the basis for which later all the statistics (and probability maps) are computed. It represents the full object. In other words it is the division of the raw data into foreground and background. Depending on the information to be transferred multiple objects can be reconstructed at once.

Labeling of Structure

The structure that is labeled (e.g. spines) is somehow the information that should be transferred to the fluorescence image domain. As input the reconstruction or the binary

image of the reconstruction is used. Even non-expert users are able to label this data because all the structures are easily recognizable.

DRFI of the Reconstruction

The DRFI of the reconstruction is computed as described above. For every voxel under a controlled environment the intensity is known. This enables to compute probability maps with DRFIs of only subparts from the reconstructions.

DRFI of the Labeled Structures

Instead of the whole structure, only from the labeled structures (spines) a DRFI is computed. This DRFI then shows how much intensity from spines is present at which locations in a fluorescence image. This DRFI is computed with the same parameters (e.g. sampling rate) like the DRFI of the whole object. This enables to compute probability maps.

Computation of the Probability Map

The DRFI of the labeled structure describes the intensity distribution for a subpart of the whole reconstruction. The same is valid for the DRFI of the whole object. It is feasible to compute how much intensity of every voxel has as source the labeled objects because the same parameters are used for both DRFIs. Given DRFI(I_d) as DRFI of the complete reconstruction I_d as binary image and DRFI(I_s) as DRFI of the spine label map I_s the probability of a voxel that the intensity is from spine is computed by:

$$P(v(x,y,z) = \text{spine}|I_d, I_s) = \frac{\text{DRFI}_{x,y,z}(I_s, \text{PSF})}{\text{DRFI}_{x,y,z}(I_d, \text{PSF})} \tag{3.3}$$

The complete spine intensity probability map SPM is computed by the element-wise division:

$$\text{SPM}(I_d, I_s, \text{PSF}) = \frac{\text{DRFI}(I_s, \text{PSF})}{\text{DRFI}(I_d, \text{PSF})} \tag{3.4}$$

The further use of the probability maps for the statistical model computation is described in all details in Section 5.2.5.

3.6.1. Information Loss

The information transfer from EM data to the fluorescence image domain is a huge gain because a new kind of information (labels) is available in fluorescence images. However, it is not possible to keep the labels in their original structure like in EM data. In the original domain binary label maps of EM reconstruction are available. The result are probability maps of spine probability per voxel. The same information must be represented as probabilities. Also the shape of the binary label maps is changed by the information transfer. All the effects have as reason the PSF. The information transfer cannot overcome the influence of the PSF. This would only be feasible (by a certain degree) by using deconvolved fluorescence images. The information transfer enables an information transfer which has a kind of information loss. It is not obvious to have probability maps instead of binary segmentations of objects (e.g. spines). However, in

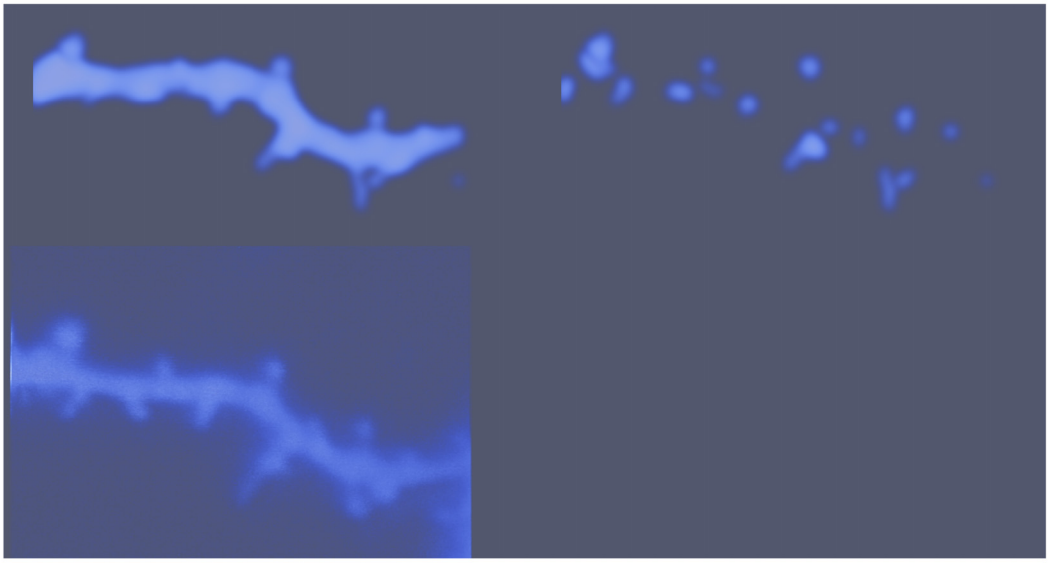


Figure 3.4.: *First row, left:* DRFI of dendrite including spines. *First row, right:* DRFI of spines. *Second row:* Same piece of dendrite as 2PM data.

principle no binary segmentation into object and non-object is feasible for fluorescence images. A binary segmentation of objects in fluorescence images is not the real shape of the object. It is at best the binarized version of the probability map which can be thresholded at different values.

3.7. Result

We present here one example of a DRFI from SBFS-EM. We use the piece of dendrite which is also part of the Correlative Light Electron Microscopy (CLEM) dataset and is called Dendrite4. In Chapter 4, Figure 4.1, the dendrite reconstruction is shown. In Figure 3.4 the resulting DRFI and the 2PM data is shown. In the *first row, left,* the DRFI of the dendrite including spines is shown. *First row, right,* shows the DRFI of only using the spines. In the *second row* the 2PM data is shown. In the 2PM domain only the dendrite including spines is available because spines can not be imaged alone.

4. Backbone of Dendrite

The backbone is an important structure for image analysis of dendrites. In this chapter we will introduce what it is in Section 4.1. Furthermore, we will introduce some fields using the backbone in Section 4.2. In Section 4.3 we will discuss the most simple approach to find the backbone. Our advanced approach to find the optimal backbone is introduced in Section 4.4.

4.1. Definition

Backbone is a common term for an artificial line in dendrites (and axons). The backbone does not exist in reality. It is only a description. The main motivation for a description like this is the modeling of dendrites as tubes. The centerline of the tube corresponds to the backbone. Therefore, the backbone describes different properties. It describes where in the space the dendrite passes through and its curvature. Furthermore, it enables to model dendrites imaged by different systems with a simple model. Figure 4.1 shows how the backbone is defined in an Electron Microscopy (EM) reconstruction. In *blue* the manual clicked backbone is visualized. In *red* spines are marked and in *yellow* enclosed mitochondria. The backbone is placed in the center of the volume of dendrite without spines (but including mitochondria). It can also intersect mitochondria.

In some fields the term skeleton is used as synonym for the backbone. However, in computer science the skeleton for dendrites including spines would result in a backbone (graph) with small side branches entering (representing) the spines. Moreover, there exists the term centerline. This term is often used as synonym for skeleton. In this thesis the term backbone is used and means the backbone visualized in Figure 4.1. The term skeleton is used for the backbone with additional side branches for the spines.

4.1.1. Meaning

Backbone is a frequently used term. However, the meaning can slightly vary depending on the field of research. We introduced how we define the backbone. For our needs the backbone is not more than a (data) structure that holds some properties of the dendrite geometry. Here we introduce what meaning the term backbone has in other fields. Furthermore, we discuss the biological background of the backbone.

Meaning in Biology

The backbone has no explicit meaning in biology. There does not exist a neuronal structure which corresponds to the defined backbone. Therefore, also in images of neurons the backbone is not visible. It is an artificial defined structure of neurons which is used in analysis. This can be in analysis of data from optical imaging (e.g. 2-Photon Microscopy (2PM)), EM or another imaging approach or representation of neurons.

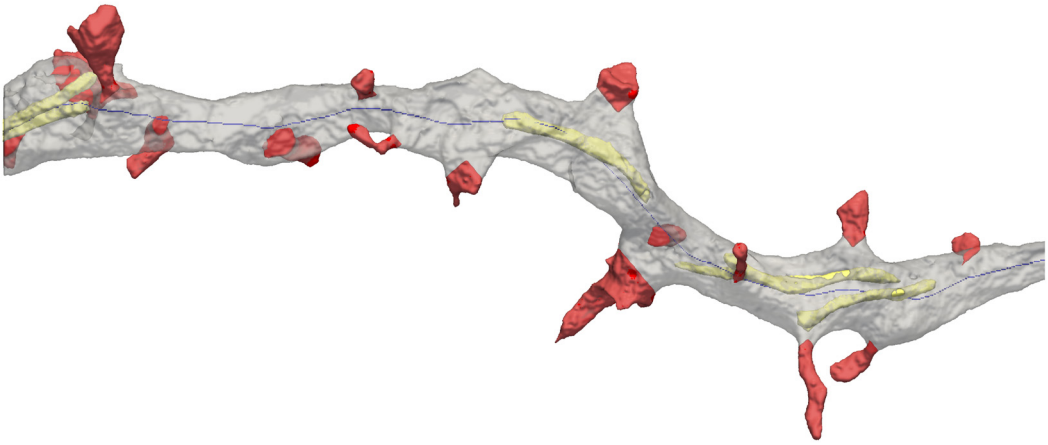


Figure 4.1.: Visualization of the backbone (*blue*) in a dendrite reconstruction from EM including spines (*red*) and mitochondria (*yellow*).

Nevertheless, many related fields use the term backbone. For instance, the change of diameter of the dendrite is easily represented with the backbone. Most applications of the backbone are related to image analysis and similar tasks.

Meaning in Other Applications

In the field of image analysis of neurons the backbone is an often used term. For instance, the backbone describes the task of tracing neurons. Furthermore, many approaches that reconstruct dendrites in images of optical microscopy use the backbone. The domain of spine detection, classification and geometrical properties of spines frequently use the backbone also. All these approaches related to image analysis show the importance of the backbone although it is only an artificial definition.

4.2. Methods Using the Backbone

Different image analysis tasks for neurons use the backbone. In the image analysis of optical microscopy images many approaches are based on the estimation and use of the backbone. In quantitative analysis and image analysis of EM images of neurons the backbone is used as a simple representation of dendrites and axons. Furthermore, the backbone simplifies the description of different properties of dendrites as it serves as well defined orientation points or line.

4.2.1. Dendrite Tracing in Optical Microscopy

For reconstruction of dendrites in optical microscopy images a backbone is computed. In [62] the centerline and backbone is used to compute reconstructions and to detect spines. In [44] the term centerline is used, but in principle it corresponds to our backbone definition. Different approaches use the backbone and an estimated radius to model neurites, e.g. [42]. This enables a light but powerful representation of dendrites if structures like spines have not to be modeled individually.

4.2.2. Neurite Tracing and Reconstruction in Electron Microscopy Images

The importance of the backbone and skeleton is very well illustrated by the research to reconstruct and find the connectome of the brain. In [27], for example, it is tried to reconstruct the brain and find the connectome based on the skeleton. In this research area many approaches try to use knowledge from the skeleton to find the reconstruction (see also [12]). The task to manually find the backbone is very time consuming. Therefore, it is tried to distribute this work and even to structure it as a game [5]. Furthermore, there exists special software to simplify the tracing of neurons [6].

4.2.3. Spine Detection based on the Backbone

The approaches [10, 11, 32, 34, 37, 50, 58, 61, 62] use in images (2-Dimensional (2D) or 3-Dimensional (3D)) the backbone to detect and segment spines. Some years ago the computational power limited the 3D processing. Therefore, some older approaches compute the backbone and segment the spines in 2D, most of them in Maximum Intensity Projection (MIP). In current research most approaches compute all steps fully in 3D. For an overview we refer the reader also to Table A.1 in Section A.1. In the approach of [60] the principle of backbone is used to represent some information. However, the approach is not fully based on the backbone.

4.2.4. Backbone Used to Compute Statistical Values

Many properties of dendrites and their spines can be computed with the use of backbone. In [50] the backbone is used to estimate distances and lengths of spines. The computation of diffusion models often uses tubes similar to the backbone and additionally a radius, e.g. in [7].

4.3. Backbone by Thinning

In image analysis the approach to skeletonize objects is no novelty. Many approaches of dendrite reconstruction and spine detection in optical microscopy images use the skeleton and then remove small side branches that represent noise or spines. How to build skeletons from objects is well studied. There exist many approaches for 2D, e.g. [4,46], and 3D, e.g. [2,6,29]. The thinning approach is the most often used approach to find the backbone or an initialization of it.

Many thinning and skeletonizing approaches use binary input images. The binary structure of the object is reduced to a skeleton which also consists of side branches depending on the shape structure. For a skeleton of a dendrite there exist small side branches for all spines but also for irregularities of the dendrite. The skeleton is a very irregular shape compared to a spline. The skeleton often drifts to spines (in addition to have side branches for spines). Therefore, all approaches using thinning to produce the backbone require to remove all (short) side branches and some smoothing of the skeleton. Fluorescence images have a irregular intensity distribution over the whole image but also locally within the dendrite, e.g. because of enclosed mitochondria. This

[5] Brainflight - `http://www.brainfligt.org`
[6] Knossos - `http://www.knossostool.org`

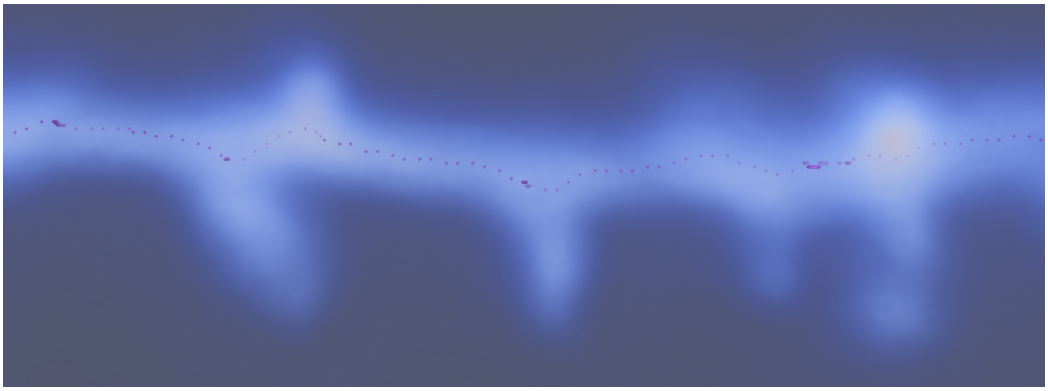


Figure 4.2.: Result of backbone after 3D thinning and removal of side branches. The backbone is visualized by the *pink* voxels.

makes it difficult to get a binary representation of the dendrite which is required for any thinning algorithm.

Figure 4.2 shows a fluorescence image and a backbone computed by thinning in 3D. The fluorescence image was binarized by a threshold. Then the 3D thinning algorithm of Insight Segmentation & Registration Toolkit - `http://www.itk.org` (ITK) [29] was applied. The skeleton was trimmed in a recursive manner to remove all side branches as described in [10]. The resulting backbone is very bumpy and has a strong curvature towards to spine regions where the side branches were located. Furthermore, this approach often fails for larger dendrite pieces where the intensity distribution along the dendrite has large differences. This is because it is not feasible to find a global valid threshold to binarize the image.

In images of larger regions branch points of dendrites exist and also other structures like axons can be present. This makes a simple thinning approach without further knowledge about structures like this useless for spine detection. All these drawbacks and the bad result of the backbone without postprocessing are good arguments not to use thinning. In most datasets only certain dendrite branches or regions should be analyzed. This motivates to let the user pick which regions to analyze and, therefore, makes the use of manually clicked backbones as initialization feasible.

4.4. Clicked and Optimized Backbone by Dendrite Intensity Models

For humans it is a simple task to recognize the interesting pieces of dendrite in fluorescence images. They can click along the interesting pieces more or less precise some points. Branch points, axons and intensity changes are no challenges for humans to solve this task. By manually clicking the dendrite the user has the full control which regions in the image are analyzed. We expect the user only to click more or less accurate along the backbone. The following optimization tries to find the optimal backbone for the use with the statistical models given this rough initialization.

The following steps are required to initialize the backbone and then optimize it by a dendrite intensity model:

- Clicking of dendrite points by the user.
- B-spline approximation of clicked points.
- Computation of normals for all sampled B-spline points.
- Computation of axes of planes orthogonal to the normal at each B-spline point.
- Extraction of planes with center at backbone points and at points shifted parallel to the central plane.
- Computation of the scaling parameters for all extracted planes.
- Computation of the dendrite intensity model probability for all extracted planes.
- Computation of new backbone points from shifted planes and model probabilities.
- Initialization of a new B-spline by positions with locally highest probabilities.
- Iteration over optimization with new backbone points and normals (optional).

4.4.1. Backbone Point Initialization by User

The user clicks points in a volume rendering viewed parallel to the laser axis (z-axis). In this view the dendrite is less elongated by the Point Spread Function (PSF) and more intuitive for the user because it looks similar to the MIP. A depth estimation is done for each clicked point. The z-value corresponds to the location with the highest intensity in the ray clicked by the user.

Given a list $P = \{p_0 \dots p_n\}$ of 2D points clicked by the user where $p_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$ the 3D point $p_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ is estimated by:

$$z_i = \underset{z_j}{\operatorname{argmax}}\{I(x_i, y_i, z_j)\} \tag{4.1}$$

The estimation of the depth is robust under the assumption that the user clicks points where no other structures are passing in different z-levels (crossing below or above). The maximum along the clicked rays is clearly inside the dendrite. Often it is even identical or nearby the expected backbone. Therefore, the simple approach of using the z-position of the position with the maximal value along the ray gives a robust initialized backbone.

4.4.2. B-Spline Through Seed Points

The list of points clicked by the user is given ordered along the dendrite. The user clicks the points along the backbone in order that they can be connected one to the next as backbone. Furthermore, the user can specify different backbone parts for each branch. Each clicked line is then individually approximated by a B-spline. To approximate the B-spline the class itkBSplineScatteredDataPointSetToImageFilter from ITK can be used. The cumulative euclidean distance from point to point is used to distribute the points on the basis. The B-spline line is then finally sampled to a list of points with constant euclidean distance from one point to the next.

4.4.3. Computation of Normals at Sampled B-Spline Points

At each sampled backbone point a vector in direction of the backbone can be computed. We call this vector normal because it will be the normal orthogonal to the plane extracted afterwards. The normal is computed by:

$$\vec{n_i} = \frac{p_{i+d} - p_{i-d}}{\|p_{i+d} - p_{i-d}\|} \tag{4.2}$$

where $d > 0$ is the distance of points used to compute the normal. The normals can also be smoothed for each backbone line. The smoothing ensures a more continuous change of the plane orientations in space. Smoothing is mostly motivated by the larger changes of the z-position along the backbone which can introduce some flickering of the normals. The larger change in z-position is given because of the elongated PSF.

4.4.4. Extraction of Planes Orthogonal to Backbone

For the given backbone points p_i and normals $\vec{n_i}$ the axis of the plane are computed in such a way that the projection of the z-axis $\vec{z}$ onto plane $s_{d,i}$ gives vector $v\vec{2}_i$. Vector $v\vec{2}_i$ is the y-axis in the image. Vector $v\vec{1}_i$ is orthogonal to $\vec{n_i}$ and $v\vec{2}_i$. As a remark: the three vectors $\vec{n_i}$, $-v\vec{1}_i$ (negative direction) and $v\vec{2}_i$ build a right hand system. The vectors are computed as follows. The z-axis $\vec{z}$ is projected on the plane $s_{d,i}$ which is orthogonal to $\vec{n_i}$ and gives $proj_{s_{d,i}}\vec{z}$. The vector $v\vec{2}_i$ is the normalized version $\frac{proj_{s_{d,i}}\vec{z}}{\|proj_{s_{d,i}}\vec{z}\|}$. The projection $proj_{s_{d,i}}\vec{z}$ is computed by:

$$proj_{s_{d,i}}\vec{z} = \vec{z} - \langle z, n\rangle n \tag{4.3}$$

The vector $v\vec{1}_i$ is computed by rotating $v\vec{2}_i$ by $-90°$ around normal $\vec{n_i}$. Finally, the shifted 2D images can be sampled from bottom left edge point $p_{i,\mathrm{BL}} = p_i - \delta v\vec{1}_i - \delta v\vec{2}_i$ to top right edge $p_{i,\mathrm{TR}} = p_i + \delta v\vec{1}_i + \delta v\vec{2}_i$ where δ is the extraction radius. As a remark it is to mention that if the normal gets close to the z-axis some flipping can happen. This has the effect that the continuity of the slices would be broken. However, we normally do not expect to have dendrites like this or that the user clicks points on such dendrites.

In the next step we optimize the backbone. Therefore, not only at the given backbone positions p_i slices are extracted but also shifted along $v\vec{1}_i$ as well as $v\vec{2}_i$. The number and region of useful shifted points of p_i is restricted by the dendrite itself. This is because the center (backbone) should be within the dendrite. It is enough to search about half of the dendrite diameter as maximum because the user clicks points on the dendrite. This can also be reduced by doing an iterative search. For all the extracted slices $s_{d,i}$ and their shifted versions $s_{d,i,j}$ the probability of having found the correct backbone location is computed. This probability enables the optimization of the backbone position.

Figure 4.3, *left*, visualizes how the evaluated points are located on the backbone. Furthermore, it shows the exact orientation in space with the z-axis and the plane vectors $v\vec{1}_i$ and $v\vec{2}_i$. The shifted extraction positions and the corresponding dendrite slices are visualized in Figure 4.3, *right*.

The optimized backbone will then be optimally located for the given dendrite intensity models. Noise of the points clicked by the user is removed and the backbone location is depending on the model. Therefore, the rough initialization of the user is enough accurate.

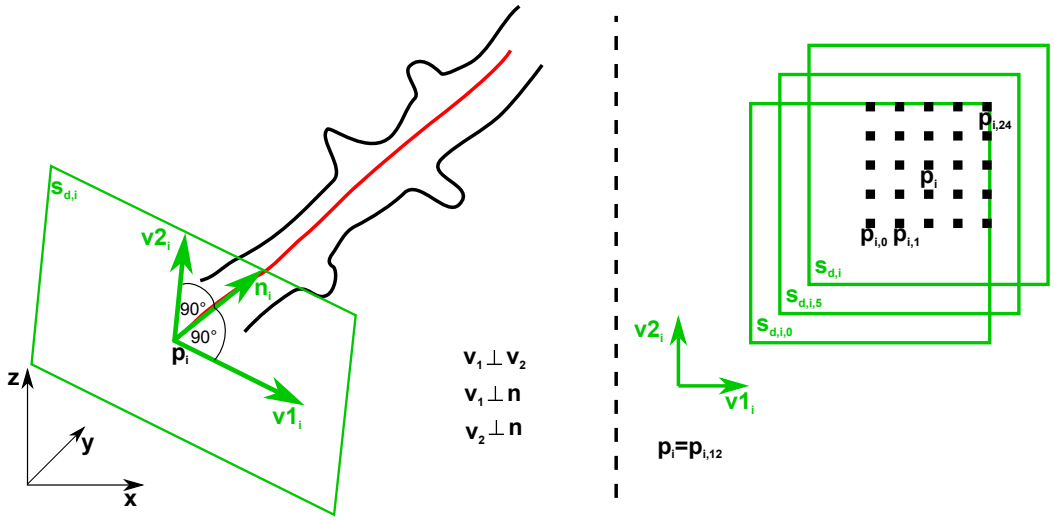


Figure 4.3.: *Left:* Overview of the backbone, the backbone-orthogonal planes and the situation in 3D. *Right:* Parallel shifted points $p_{i,j}$ and the extracted dendrite slices $s_{d,i,j}$.

4.4.5. Scaling and Intensity Normalization of Extracted Planes

For the use with a statistical model the planes must be aligned and scaled to the reference. The alignment corresponds to the different positions parallel to one plane extracted. Therefore, we do not align the extracted planes. The scaling resizes the extracted dendrite cross sections to the same size. This is described in more detail in Section 5.2.4. The intensity normalization is required because of the possible intensity changes over the whole 3D volume. We do a simple normalization that is described in Section 5.2.3.

4.4.6. Probability of Extracted Planes

The probability $P(s_{d,i,j}, M_d)$ describes how probable the given image (or extracted slice) and model M_d are. The probability is computed by:

$$P(s_{d,i,j}, M_d) = P(s_{d,i,j}|M_d)\, P(M_d) \tag{4.4}$$

The model probability $P(M_d)$ is 1 if one model is used. By using multiple models it can be computed directly from the data by the distribution of the samples. The best location of the backbone is defined by the probability $P(s_{d,i,j}, M_d)$. How the posterior probability $P(s_{d,i,j}|M_d)$ is computed is described by Equation (5.17) in Section 5.3. As extension the probability is computed for multiple models M_d^k which are composed for different oriented spines and training data without spine. The best location given all shifted planes and models is found. Furthermore, over all backbone points also the best locations are used as new seeds. Finally, a new B-spline can be approximated through all the computed seed points. For further details about computation of the model probability we refer the reader again to Section 5.3.

4.4.7. Backbone Position Interpolation within Shifted Planes

The optimal backbone position is estimated for the computed probabilities. The location with highest probability is selected from all shifted positions. At the given location a new location is computed using the neighbors (8-neighborhood) and its probabilities as weighting. Finally, the new found positions over all backbone points are sampled ordered by the probability distribution of all the interpolated backbone positions. After each sampled point the other interpolated backbone points within a given radius are removed from the sampling pool. This ensures that the seed points are not to dense and the B-spline has some flexibility and smoothness. The approximated B-spline has a smooth shape which is well adapted to the natural shape of dendrites.

4.4.8. Initialization of B-Spline by Interpolated Locations

B-Splines are a flexible and smooth representation of the backbone. We approximate the seed points by a B-spline. The seed points are the points clicked by the user or later on computed during the backbone optimization. Instead of using all optimized backbone points the use of the B-spline ensures the natural shape of the backbone. Finally, the backbone and thus the backbone points are the B-spline or sampled points from it.

4.4.9. Recomputation of Plane Parameters

As last step the plane parameters must be updated. The normals $\vec{n_i}$ and the corresponding vectors $v\vec{1}_i$ and $v\vec{2}_i$ of the new backbone are updated. The optimization can be repeated or the found backbone be used for spine prediction depending on the expected precision of the points clicked by the user.

4.5. Probability Distribution of Shifted Planes

The probability distribution $P(s_{d,i,j}, M_d^k)$ of the shifted planes for the models M_d^k describe the optimal location of the backbone. Figure 4.4 shows the probability distributions for all the models with $k = \{1, \ldots, 9\}$ where $k = 9$ is the model without spine and the other ones encode different spine orientations. The x- and y-shift are computed for seven positions with a shift delta of 0.1 µm in x- and 0.3 µm in y-direction where the x- and y-direction correspond to $v\vec{1}_i$ and $v\vec{2}_i$ respectively. The number of shifted locations and shift delta was estimated manually. Figure 4.4 shows for all the models how the probability distributions are smooth and direct to zero at the borders. Furthermore, the elongation along the laser axis is recognizable in the y-axis which is $v\vec{2}_i$. This is because the change along this axis is slower. The example shows also how in all cases a clear maximum is recognizable. This motivates to take the location with the maximum over all shifted positions and models. To get a more robust new location the local neighborhood is used to interpolate the position. The main motivation not to average over all shifted locations is that for planes with spine in the directions of spines more far away from the optimal backbone high model probabilities exist. The new backbone positions would drift in the directions of spine (similar to the issue of the thinning approach).

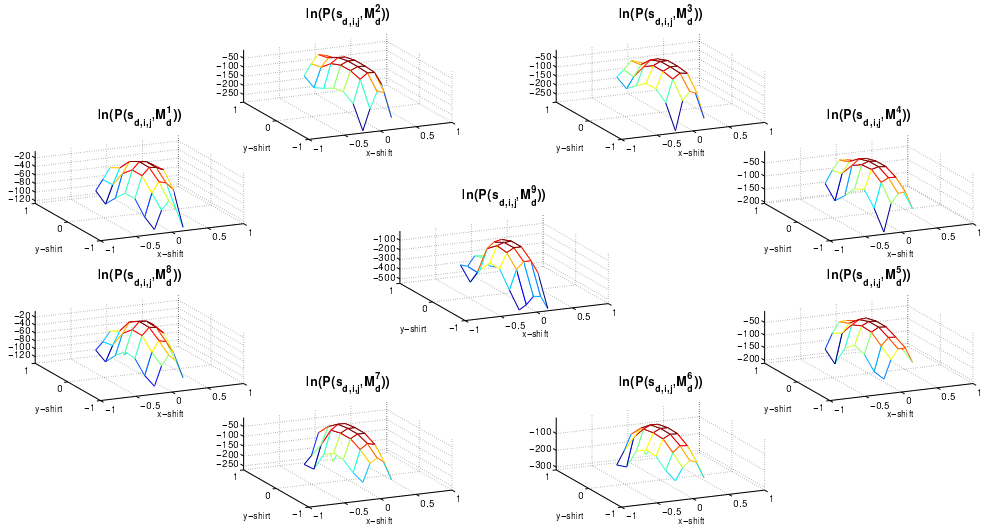


Figure 4.4.: Probability of shifted planes $s_{d,i,j}$ and models M_d^k. The plots are arranged corresponding to the spine orientation the models encode (see also Figure 5.11).

Figure 4.5 shows the effect of drifting away if a spine is present in one direction. The optimal backbone would be located at the center (x-shift and y-shift equal zero), but the probability distribution maximum is located more in positive x-shift direction in which direction also a spine is located. This effect is present for all models more or less, even for the one having encoded spines in that direction.

The effect that even the probability distribution for the model with the correct spine orientation drifts away in direction of spine is not understandable at first. The correct model has the spine encoded in that direction and also for the mean of that model the spine in this direction is recognizable. Nevertheless, there are larger changes from the mean model sample required to get the spine correct represented than to have only represented the mean and drift into spine direction. The larger the spine is the more this effect is visible.

If the shifted position is outside from dendrite intensity the probability converges to zero. For positions within the dendrite (intensities larger than zero) the probabilities build a smooth surface. Taking the average position weighted by the probabilities does not work for all cases. In regions without drift to a spine the weighted new position is correct. However, in regions having a drift into direction of spine the weighted average position would be wrong. In such cases the position with maximal probability is more accurate. In Figure 4.5, *bottom right*, it is visible that the maximum is still located at position $xshift = 0, yshift = 0$. The backbone should also be located at that central location. Therefore, better results of the interpolated backbone position are computed using the maximum of all the shifted position probabilities and to interpolate in a small neighborhood (8-neighborhood) to get the optimal position better resolved than the used shift delta. Of all probabilities $P(s_{d,i,j}, M_d^k)$ the optimal new position $p_{i'}$ is found by the position and model of the maximal probability which is defined by:

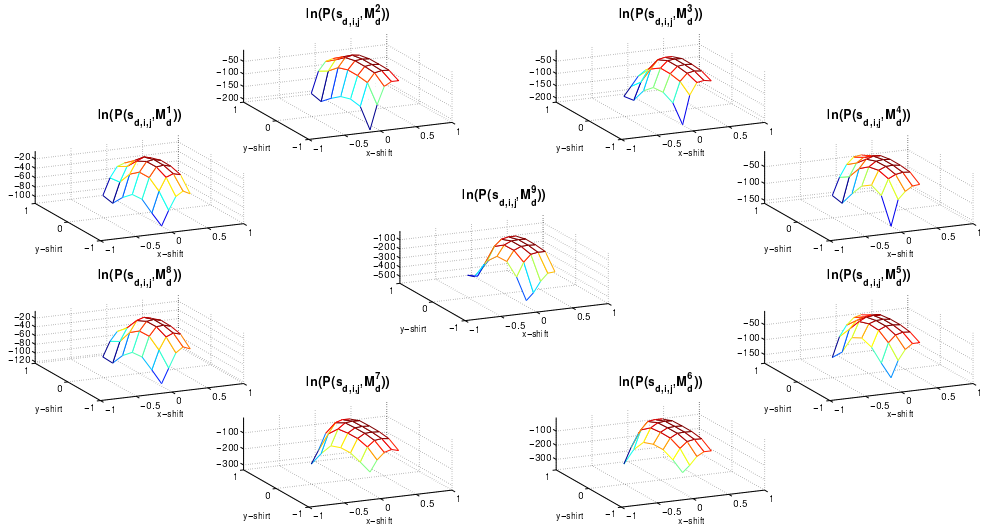


Figure 4.5.: Probability of shifted planes $s_{d,i,j}$ where the maximum is drifting in the direction of a spine (positive x-shift direction).

$$\underset{j,k}{\operatorname{argmax}}\{P(s_{d,i,j}, M_d^k)\} \quad \forall j,k \tag{4.5}$$

Then for the position of the maximum the neighborhood is interpolated by using the probabilities:

$$p_{i'} = \sum_{l=neighbors(j)} p_{i,l} P(s_{d,i,l}, M_d^k) \tag{4.6}$$

4.6. Backbone Optimization Results

The optimization of the backbone results in an optimal backbone with respect to the models. The optimization ensures that the user only has to click points on the dendrite but not perfectly located. Wrong clicked points are improved by the optimization process if they lie within the given search radius or do not become seed points for the B-spline. An interpolation between seed points finds the optimal location for regions without seed point.

Figure 4.6 shows the result of the optimization process. The *top row* shows the first iteration and the *bottom row* the second iteration. In the first iteration the initialization (*pink vectors*) is done by the manually clicked points. Clearly visible in the right part the backbone initialization is very imprecise. The highest probability of all shifted positions of every backbone position are shown as *gray vectors*. At completely off positions the probability is very low or zero. In contrast, at correct initialized locations probabilities greater than zero are visible. In *green* the optimized backbone is displayed. It is quite well adapted to the dendrite. At wrong initialized positions a larger shift within one iteration is not possible because of the search radius. In the *bottom row* the same process is repeated, but the result of the first iteration is used as initialization

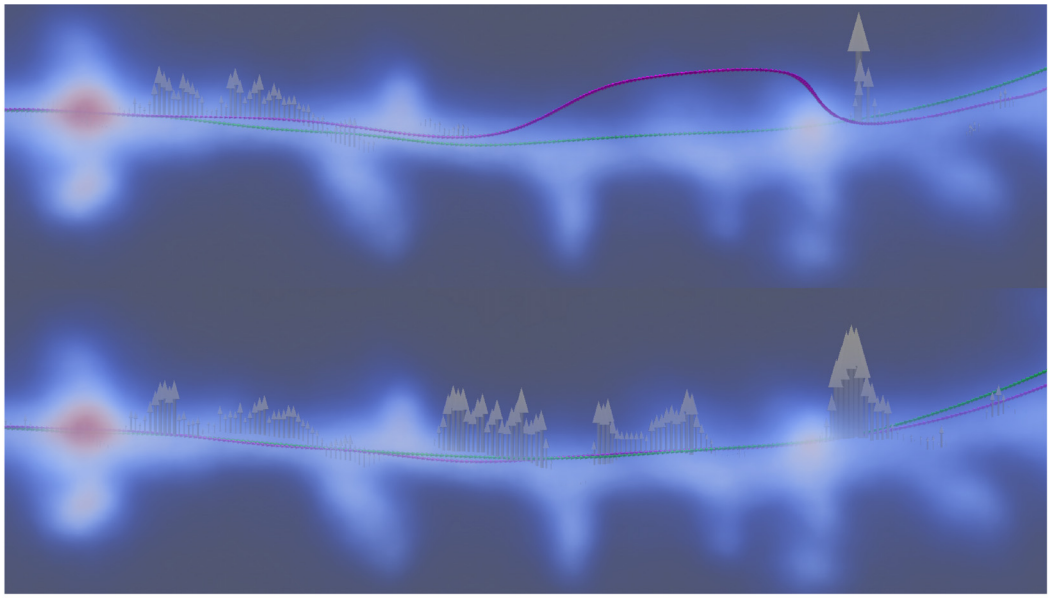


Figure 4.6.: *Top:* 1st iteration, the manual initialized backbone (*pink*) finds the new best locations (*gray*, vector length encodes probabilities) and becomes the resampled new backbone (*green*). *Bottom:* 2nd iteration, instead manual initialization the backbone from the previous iterations is used.

(*pink vectors*). In this step for all the backbone points a good location is found (*gray vectors*) and the optimized backbone (*green vectors*) is located at the expected position.

Further examples of backbone optimization can be found in Appendix A.3.

5. Statistical Dendrite Intensity and Spine Probability Models

An important novelty of our approach are the statistical dendrite intensity and spine probability models. These two models incorporate most of the previous steps in form of a model. Statistical models are concepts from learning approaches. In our approach we use a learning method based on data and given labels. This is a supervised learning approach. A novelty is the way we get our learning data. Moreover, the use of corresponding models to generate a prediction of spine probability per pixel was never done before.

We introduce the concept in Section 5.1. In Section 5.2 we describe how the training data is generated. The computation and combination of the models is described in Section 5.3. In Section 5.4 we present results of the models and in Section 5.5 we present the extension to spine orientation dependent models. The practical use of the models for spine detection will be described in Chapter 6.

5.1. Concept

The goal of our models is to predict the spine probability or likelihood per voxel given a fluorescence intensity image. In Chapter 4 we introduced the backbone-orthogonal slices (see Figure 4.3). The models use the same concept of backbone-orthogonal slices. The prediction of spine probability will be per pixel in 2-Dimensional (2D) slices and not directly in the 3-Dimensional (3D) image. However, the prediction can be projected back into the 3D image. We use the concept of Digitally Reconstructed Fluorescence Images (DRFIs) to generate intensity images and probability images which are used as training data. The concept of backbone-orthogonal slices and DRFIs enable the statistical learning with 2D images. Due to the corresponding data in fluorescence intensity images and spine probability maps we can train and combine both models together. This enables us to transform coefficients and representations in one model space to the other model space and their corresponding coefficients.

5.2. Data Generation

Statistical models require training data. Often data is available in large databases and no data generation is required. If no database is available or depending on the requirements the generation of new data is necessary. A lot of fluorescence data is available because biologists image dendrite and spines since years. However, the required labels (segmentations) are not available. Therefore, we use the idea of synthetic data and compute DRFIs of dendrites and spines. This data enables us to generate the final training data. The training data are 2D images which are based on the DRFIs.

5.2.1. Computation of Digitally Reconstructed Fluorescence Images

In Chapter 3 we introduced the concept of DRFIs and discussed the use of synthetic data in general. Furthermore, we introduced the information transfer from the Serial Block-Face Scanning Electron Microscopy (SBFS-EM) domain to the fluorescence image domain. Related to the information transfer we introduced the computation of spine probability maps. These steps together enable us to compute 2D training data which are dendrite intensity images and spine probability maps. The application has some prerequisites for the models. One of these prerequisites is that the dendrites to analyze are mostly horizontal oriented. This knowledge can be used already during the data generation. Therefore, we generate training data from mainly horizontal oriented pieces of dendrite. We use horizontal rotated version of the dendrite pieces to generalize more and to have more training data. This is feasible because the same spine has different intensity distributions in DRFIs and fluorescence images if the orientation in space is changed. We rotate the pieces of dendrite around an axis parallel to the main orientation of the piece of dendrite in 10° steps. This gives us 36 times more unique data. A horizontal mirroring doubles the amount of training data again.

5.2.2. Backbone-Orthogonal Slices

The use of backbone-orthogonal slices has some advantages for our statistical model. It is feasible to get much more training data for 2D backbone-orthogonal slices than 3D images. It would be difficult to get enough, representative training data in 3D. The alignment and registration of the samples as 2D slices is straightforward and can be semi-automatic. The computation of the models and processing in general is much easier and less computational expensive than it would be for 3D data. Moreover, the handling of the corresponding models is more intuitive and visual interpretations are simpler. These advantages motivate to use the approach of backbone-orthogonal slices in the statistical learning approach.

We introduced the backbone-orthogonal slices in Section 4.4.4. For details about the extraction of slices we refer the reader to that section. The main advantages to use backbone-orthogonal slices for statistical model computation are: smaller data amount, high number of training samples, feasible alignment and normalization. In difference to the description in Section 4.4 we use a perfectly located, manually specified backbone. The alignment cannot be done by the backbone optimization because the optimization itself requires the statistical model.

5.2.3. 2D Slice Intensity Normalization

Intensities in fluorescence images can vary very strong. Changes of imaging parameters, fluorescent protein and the individual neuron in general can result in different intensity values from image to image. Moreover, in addition to the changes from image to image also within one image the intensity range for the same structure, e.g. intensity values along the backbone of the dendrite, can be very large. Therefore, the statistical model computation requires to normalize the 2D slices and not only the 3D fluorescence image. The normalization is only required for the dendrite intensity images but not for the spine probability maps. This is because the spine probability maps are real probabilities. The normalization used is a rescaling of the intensity values to the range

$[0, 1]$ and is computed by:

$$I_n = \frac{I - \min(I)}{\max(I) - \min(I)} \tag{5.1}$$

where I is the input image. In general the minimum intensity corresponds to the background (no fluorescence). The maximum intensity is in most cases located at the center of the backbone. At backbone positions with large enclosed organelles (like mitochondria) the intensity can be much lower. Large spines can also have the maximum intensity of the image.

The intensity normalization is applied before computing the registration (scaling). The registration is depending on the intensity values which should be equal over the different samples.

5.2.4. 2D Slice Alignment and Scaling

The goal of the 2D slice registration for statistical models is to ensure that every sample point (pixel) represents the same semantic information. This means to have all samples in correspondence. Alignment and scaling are related to registration approaches. There exist many registration methods which tackle different requirements. The main division of registration methods is into rigid and non-rigid registration. What kind of registration is required is given by the data and its semantic interpretation. In our case we define the required registration as following:

- The direction of z-axis or projection of z-axis in 2D should always direct into the same direction (defined as y-axis in 2D). This is required because the Point Spread Function (PSF) is elongated along the optical axis (z-axis) and can be asymmetric too. Therefore, we keep the orientation of the z-axis including the direction of it (up and down).
- The center of object (cut through the dendrite) is defined by the backbone and corresponds to the center of the 2D image.
- The edge of the object is defined by a threshold value for the intensity normalized 2D image. The edge line of a round object would correspond to a circle.
- Spines can occur in all directions. However, we will not rotate them to a given direction because of the asymmetry and elongation issues of the PSF. Moreover, for multiple spines it would be difficult to compute a unique, reasonable rotation.

Given these definitions and requirements we register the 2D images to a template as following:

- No translation of the object. The extraction center, located at the backbone, is the image center.
- No rotation to register spines. Spines can exist in any direction from the center.
- Independently scaling of the dendrite extension (edge line) in x- and y-direction to a given template (distance).

Figure 5.1 shows the template definition. Point C marks the center which is kept unchanged. We compute a horizontal and a vertical scaling value. The scaling is done with the origin equal to C. The scaling values are computed independent for horizontal

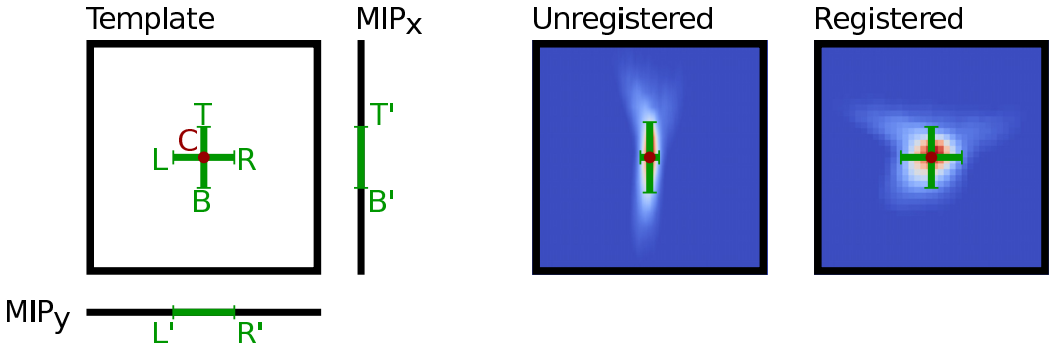


Figure 5.1.: *Left:* The registration template for 2D slice images. The center C is fix. L, L', R, R' and T, T', B, B' mark the measurement points at given threshold values for the horizontal and vertical distance respectively. *Middle:* Unaligned sample with schematically showed points T, B, L, R. *Right:* Same as *middle* but in registered version.

and vertical orientation. We explain it for horizontal scaling which uses L, L', R and R'. The vertical scaling parameter is computed analogously using T, T', B and B'. For a given 2D image the horizontal distances from the center to left and right where the intensity drops under a given threshold is computed. This results in L and R. The same is done for the Maximum Intensity Projection (MIP) projected along the y-axis. This gives the values L' and R'. The horizontal scaling factor s_h is then computed by:

$$s_\mathrm{h} = \frac{s_\mathrm{f}}{\min\left(\frac{L+L'}{2}, \frac{R+R'}{2}\right)} \tag{5.2}$$

where s_f is the defined distance for the alignment points. The averaging of the two measurement methods makes the alignment more robust. Further robustness of the scaling can be reached by smoothing the computed scaling values along the backbone. The smoothing ensures a slow change of scaling along the backbone. This corresponds to the natural shape of the dendrite which has no discrete diameter changes. Only spines or wrong located backbone points can introduce large changes of the dendrite diameter. The issue of too large distances caused by spines is omitted by taking the minimum value from left/right and top/bottom.

In Figure 5.2 we show examples of registered and unregistered slices. The slices showed are from DRFIs 2D images that are also used for the model computation. The backbone was manually specified. It is well visible that a registration of the training data is necessary. The *first column* shows the original dendrite intensity image. In the *second column* the registered version is visible. The *third* and *fourth column* show the corresponding images for the corresponding spine probability map. In the *second column* the registered images clearly show that the part of the registered dendrite corresponds to a circular or 2D Gaussian distribution. This corresponds to the expectation that in general is given for the dendrite and the PSF if the z-elongation is compensated.

To compare how our DRFI data looks like we show in Figure 5.3 some comparison between the data used for the model and the corresponding real data extracted from the 2-Photon Microscopy data available for these pieces of dendrite. The *first column* shows the unregistered DRFI data. The *second column* shows the registered version. In the *third* and *fourth column* we show the corresponding data from 2-Photon Microscopy

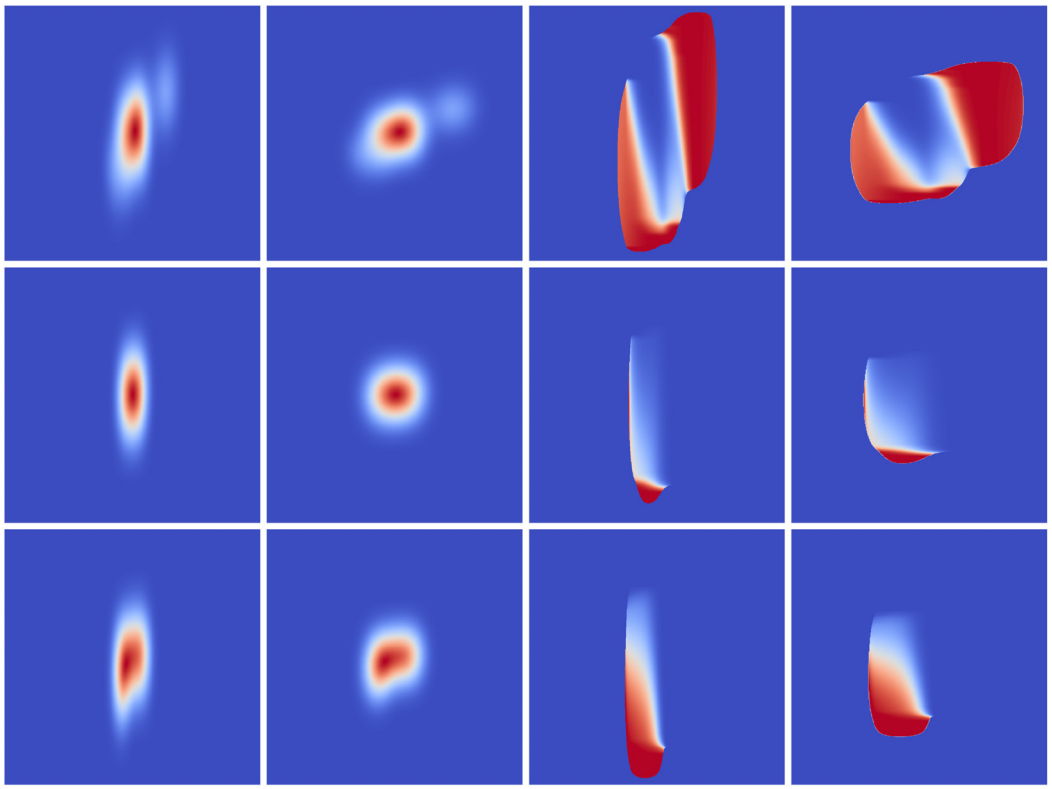


Figure 5.2.: Each row is an example. *First column:* Unregistered dendrite intensity image. *Second column:* Registered dendrite intensity image. *Third and fourth column:* Corresponding images for the spine probability map.

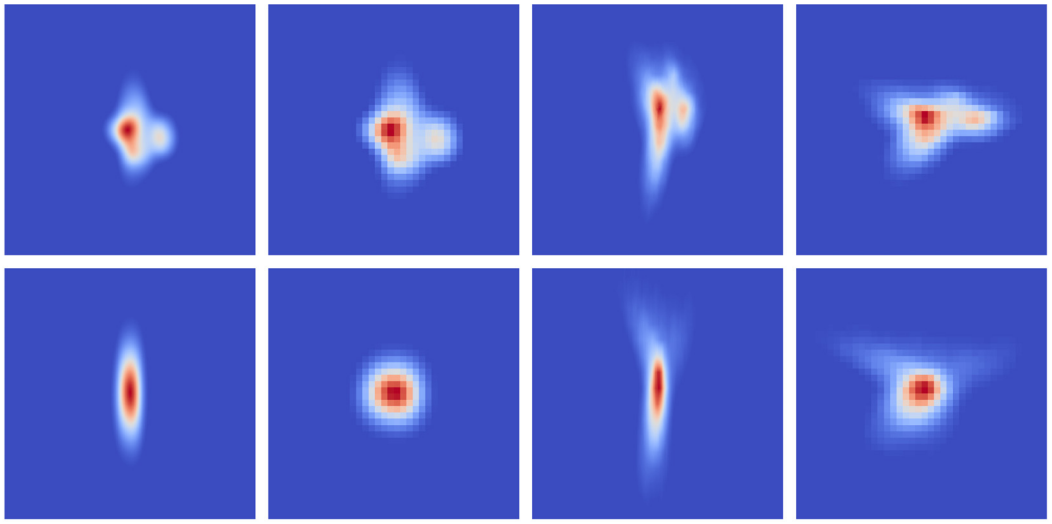


Figure 5.3.: Each row is an example. *First column:* Unregistered dendrite intensity image. *Second column:* Registered dendrite intensity image. *Third and fourth column:* Corresponding images for the 2-Photon Microscopy data.

(2PM). The real data is extracted from the same location as accurate as possible. However, only a manual overlay and mapping of the positions is possible because SBFS-EM data was independently imaged from the 2PM data. Nevertheless, Figure 5.3 enables a good understanding how the quality of the registration is similar for synthetic and real fluorescence data.

5.2.5. Spine Probability Maps

The spine probability maps are essential to predict spines in fluorescence images. Thanks to DRFIs we are able to compute corresponding images for the geometry of the dendrite including spines and the spines alone. In Section 3.6 we introduced the information transfer in general and especially for the probability of a voxel to be spine or not. The resulting probability maps are used to compute the statistical model. The spine probability maps are extracted along the backbone in the same way like the corresponding dendrite intensity 2D images. The correspondence of the probability maps and dendrite intensity images is important. Each spine probability map corresponds to a 2D dendrite intensity image. They have the same parameters for alignment and scaling. This enables to compute related models. In difference to the dendrite intensity images the spine probability maps must not be normalized (see Section 5.2.3).

We overcome the issue to compute a model from expert labeled data by the spine probability maps. The model from the spine probability maps computed from DRFI data introduces a new concept and possibilities. It is the first approach to learn in the fluorescence domain with data that is originally from the Electron Microscopy (EM) domain. The spine probability maps are also not only a classifier of the slices into spine and not spine but a probability of each voxel or pixel to have spine contribution. This enables us to make a more precise decision what to classify as spine and what not.

Examples of registered and unregistered spine probability maps can be found in Figure 5.2.

5.3. Model Computation and Coupling

In this section we introduce the computation and coupling of the two statistical models. Both models are independently computed Principal Component Analysis (PCA) models. PCA models define an orthogonal system that describes optimal the variance in the dataset. Further details about properties of PCA are described in Section 2.5.3. As introduced we use backbone-orthogonal, registered (aligned and scaled) 2D images as training data. We call the dendrite intensity model M_d and the spine probability model M_s. The training data for model M_d is named $s_{d,i}$ and for model M_s it is $s_{s,i}$. We compute for both datasets an independent PCA model:

For the dendrite intensity model we have by using the Singular Value Decomposition (SVD)

$$X_d = U_d D_d V_d^T \tag{5.3}$$

where

$$X_d = \left[s_{d,1} - \mu_d \dots s_{d,n} - \mu_d\right] \tag{5.4}$$

and for the spine probability model

$$X_s = U_s D_s V_s^T \tag{5.5}$$

where

$$X_s = \left[s_{s,1} - \mu_S \dots s_{s,n} - \mu_s\right] \tag{5.6}$$

Both matrices X_d and X_s are of size $\mathbb{R}^{m \times n}$ and the training images are vectorized in column form. The mean of the training data is subtracted to have mean free data. The mean is named μ_d and μ_s for the corresponding models and data and is computed by (μ_s analogously):

$$\mu_d = \frac{1}{n} \sum_{i=1}^{n} s_{d,i} \tag{5.7}$$

A fluorescence image sample s_d can now be approximated by the dendrite intensity model M_d using:

$$s_d \approx M_d(\alpha_d) = \mu_d + U_d \alpha_d \tag{5.8}$$

This results in the PCA coefficients α_d:

$$\alpha_d = U_d^T (s_d - \mu_d) \tag{5.9}$$

These coefficients can be transformed to the linear combination of examples by first right-multiplying Equation (5.3) with $V_d D_d^{-1}$ which gives us:

$$X_d V_d D_d^{-1} = U_d D_d V_d^T V_d D_d^{-1} = U_d \tag{5.10}$$

$\alpha_d = \left[\alpha_{d,1}, \dots \alpha_{d,\tilde{n}}\right]^T$ and $\beta_d = \left[\beta_{d,1}, \dots \beta_{d,\tilde{n}}\right]^T$ are coefficient vectors so that:

$$\sum_{i=1}^{\tilde{n}} \alpha_{d,i} u_{d,i} = U_d \alpha_d = X_d V_d D_d^{-1} \alpha_d = X_d \beta_d = \sum_{i=1}^{n} \beta_{d,i} x_{d,i} \tag{5.11}$$

We can get from the PCA coefficients α_d to the linear combination of examples β_d by:

$$\beta_d = V_d D_d^{-1} \alpha_d \tag{5.12}$$

In the opposite direction we have:

$$\alpha_d = D_d V_d^T \beta_d \tag{5.13}$$

The same is valid for the model M_s and we can compute from α_s to β_s and vice versa. The parameters β_d and β_s describe the linear combination of examples with which the models are computed. As a precondition for the model computation we used corresponding example data in both models. Therefore, any linear combination of examples from one model can be used with the other model and results in the corresponding slice. This means that a linear combination β_d which describes a dendrite intensity example can also be used for describing the corresponding spine probability map. This means that $\beta_d = \beta_s$. This enables us to compute from PCA coefficients of model M_d directly the PCA coefficients of model M_s:

$$\alpha_s = D_s V_s^T \beta_s = D_s V_s^T \beta_d = D_s V_s^T V_d D_d^{-1} \alpha_d \tag{5.14}$$

Also the computation in the other direction is feasible. However, we are interested in getting from dendrite intensities to the probability of spines.

For a given (test) slice $s_{d,\hat{i}}$ we can then find the prediction map $s_{s,\hat{i}}$ by first project $s_{d,\hat{i}}$ into model M_d which gives the coefficients $\alpha_{d,\hat{i}}$:

$$\alpha_{d,\hat{i}} = U_d^T \left(s_{d,\hat{i}} - \mu_d \right) \tag{5.15}$$

From the PCA coefficients $\alpha_{d,\hat{i}}$ for the slice $s_{d,\hat{i}}$ we can then compute the reconstruction of the spine probability map $s_{s,\hat{i}}$ using Equation (5.14) and the reconstruction in the spine probability model:

$$s_{s,\hat{i}} = \mu_s + U_s \alpha_{s,\hat{i}} = \mu_s + U_s D_s V_s^T V_d D_d^{-1} \alpha_{d,\hat{i}} \tag{5.16}$$

An important property of the statistical model is that we can calculate how probable it is to observe a slice given a model. This is the so called posterior probability. The probability to observe a slice $s_{d,\hat{i}}$ given model M_d is calculated by:

$$P\left(s_{d,\hat{i}} | M_d\right) = \frac{1}{\sqrt{(2\pi)^k} \frac{|D_d|}{\sqrt{n}}} \exp\left(-\frac{1}{2} \| \sqrt{n} * D_d^{-1} \alpha_{d,\hat{i}} \|^2 \right) \tag{5.17}$$

where k is the dimension and number of PCA coefficients. Depending on the application the normalization term is constant and can be skipped if only relative values are required and not the real probabilities. We refer the reader also to the general definition of the multivariate Gaussian distribution:

$$f_x(x_1, \ldots, x_k) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right) \tag{5.18}$$

where $|\Sigma|$ is the determinant of Σ. Σ is the covariance matrix of X and corresponds to

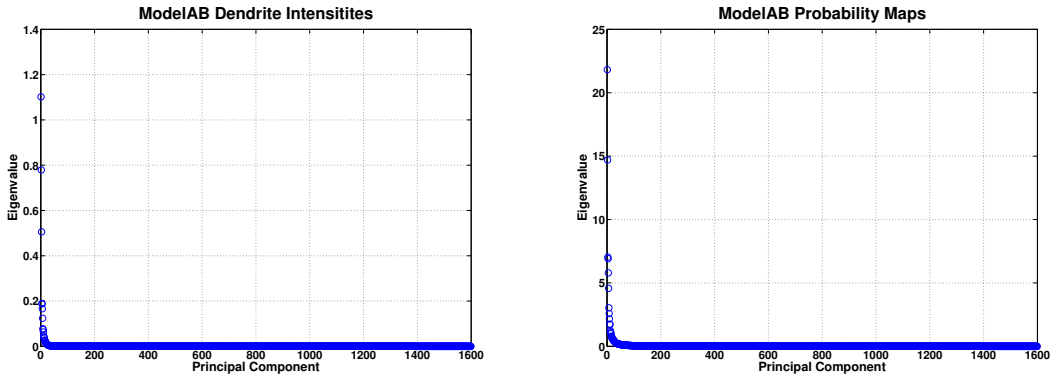


Figure 5.4.: *Left:* Fast decreasing eigenvalues from dendrite intensity model. *Right:* Eigenvalues from spine probability model.

$$\Sigma = \frac{1}{n} X X^T = \frac{1}{n} U D V^T V D U^T = \frac{1}{n} U D^2 U^T \tag{5.19}$$

using the representation of SVD. For further information about multivariate Gaussian distributions we refer the reader to the corresponding literature, e.g. [55].

The joint probability $P(s_{d,\hat{i}}, M_d)$ describes how probable it is to observe a slice $s_{d,\hat{i}}$ and to have model M_d. This joint probability can be computed by the conditional probability from Equation (5.17) and the probability $P(M_d)$ of the model:

$$P(s_{d,\hat{i}}, M_d) = P\left(s_{d,\hat{i}} | M_d\right) P(M_d) \tag{5.20}$$

We introduced and used the joint probability to optimize the backbone (see Section 4.4).

5.4. Results

We reconstructed in total four pieces of dendrite of one cell. Two of them are part of the correlative dataset and are named Dendrite4 and Dendrite6. For the other two pieces no 2PM data exists. They are named DendriteA and DendriteB. In Appendix A.4, Figure A.4, we show the reconstruction of the four pieces. Here we show results for a model computed with the non correlative data (DendriteA and DendriteB) that we name ModelAB. As earlier introduced, we rotate the dendrites in 10° steps to increase the amount of training data. The correlative pieces of dendrite (Dendrite4 and Dendrite6) are used as test dataset. The model is computed from 121 752 2D slices that are extracted using a manual clicked backbone and registered as described in Section 5.2.4. Figure 5.4 shows a plot of the eigenvalues of the PCA models. It is well recognizable that they decrease very fast. This enables us to cut off some components. We show results for using all (1599) Principal Components (PCs) and a reduced version with 50 PCs.

In Figure 5.5 we show the plots of the proportion of total variance. The proportion of total variance was introduced in Section 2.5.3. In these plots the fast increasing percentage is well recognizable and a stop criteria for the number of components like

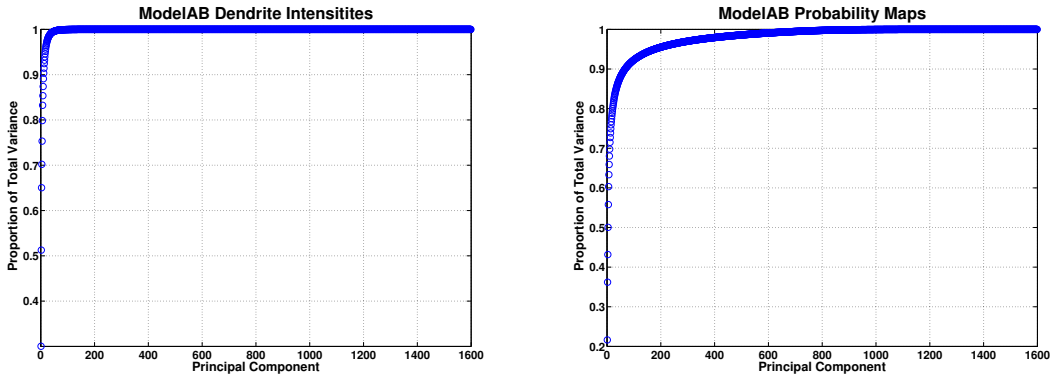


Figure 5.5.: *Left:* The proportion of total variance for the eigenvalues from the dendrite intensity model. *Right:* Same plot for the spine probability model.

the 95 % proportion of total variance can be picked. The function for plotting the proportion of total variance is given in Section 2.5.3, Equation (2.19).

The mean images of both models are presented in Figure 5.6. *Left* is the mean of the dendrite intensity model and *right* the mean of the spine probability model. It is well visible that in general always a contribution of the spine to the intensity is expected around the center of the dendrite. Especially, on the horizontal expansion of the dendrite the contribution from spines is well visible in the mean image. This shows that even for our computation with a synthetic PSF the intensities of spines are distributed over larger distances. Therefore, the detection and segmentation of spines is such a difficult task. The maximum intensity in the mean image of the spine probability model is 0.37. This means that we expect for some pixels (and positions) that on average 37 % of the intensity is from spines. The maximum intensity for the mean image of the dendrite intensity model is 0.93. This number shows that the training data is quite well aligned as the normalized intensity is between 0 and 1. We expected the highest intensity for all dendrite intensity slices to be located at the same position. Furthermore, the highest intensities are found as expected in the center of the image what corresponds to the backbone location. The dendrite intensity distribution of the mean image is similar to a Gaussian distribution.

In Figure 5.7 and Figure 5.8 we show the effect of the first three PCs of the dendrite intensity model and the spine probability model. The *first to third row* correspond to the first to third PC of the corresponding model. In the *first column* the mean image and the line through which the plot is are shown. The *second and fourth column* are 3σ and -3σ of the n^{th} PC. The *third and fifth column* are the mean plus 3σ and -3σ of the n^{th} PC. The *sixth column* shows a plot through the pink line shown in the first column for the *first, third and fifth column*. In this plot also negative values which we clip for further processing are visible. For the *first, third and fifth column* the displayed intensity range is $[0, 1]$. For the *second and fourth column* the displayed intensity range is $[-1, 1]$. A semantic interpretation of the PCs is possible. However, no real reasons for the interpretation can be given. Therefore, we leave any semantic interpretation of the first three PCs up to the reader.

In Figure 5.9 we show examples of prediction results for DRFI data which enables comparison to ground truth data. The used slices are not part of the model. Each row corresponds to one example. The *first column* shows the dendrite intensity slice. The

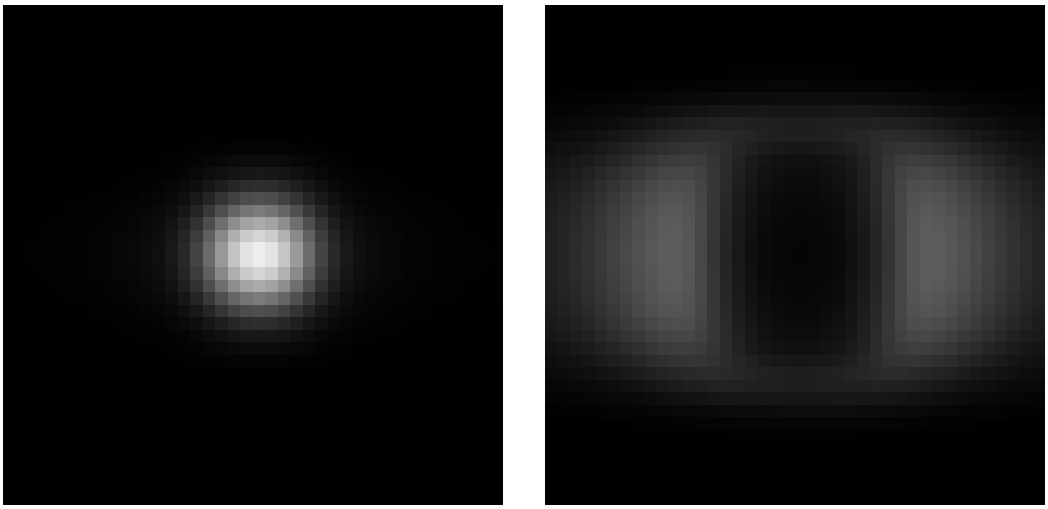


Figure 5.6.: The mean images of the two PCA models. *Left:* Dendrite intensity model ModelAB. *Right:* Spine probability model ModelAB.

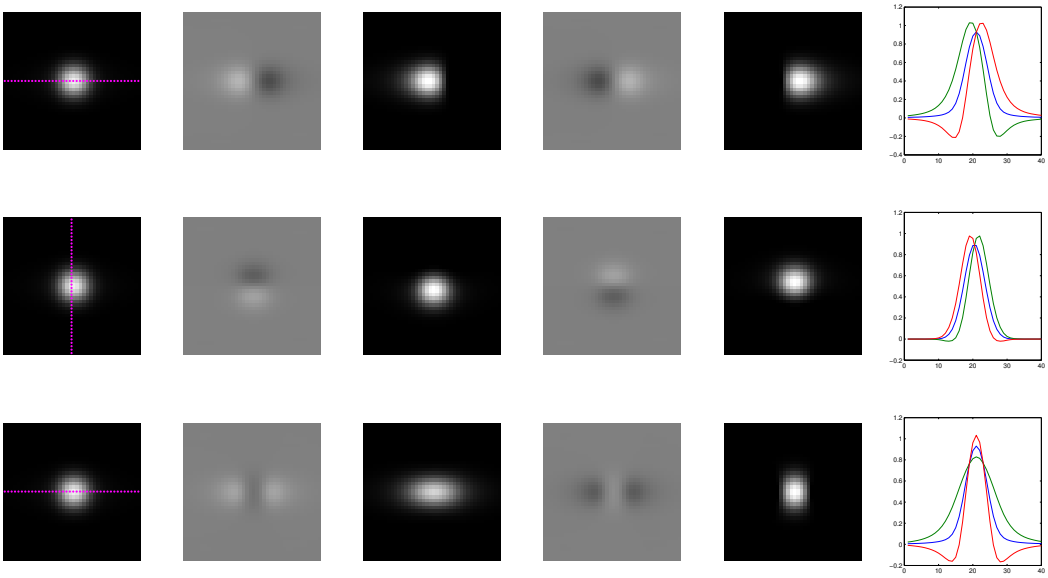


Figure 5.7.: Effect of the first three PCs for the dendrite intensity model. *First column:* Mean image. *Second column:* 3σ of PC. *Third column:* $\mu + 3\sigma$ of PC. *Fourth column:* -3σ of PC *Fifth column:* $\mu - 3\sigma$ of PC. *Sixth column:* Plot for 1^{st} (*blue*), 3^{rd} (*green*) and 5^{th} (*red*) column at *pink* line.

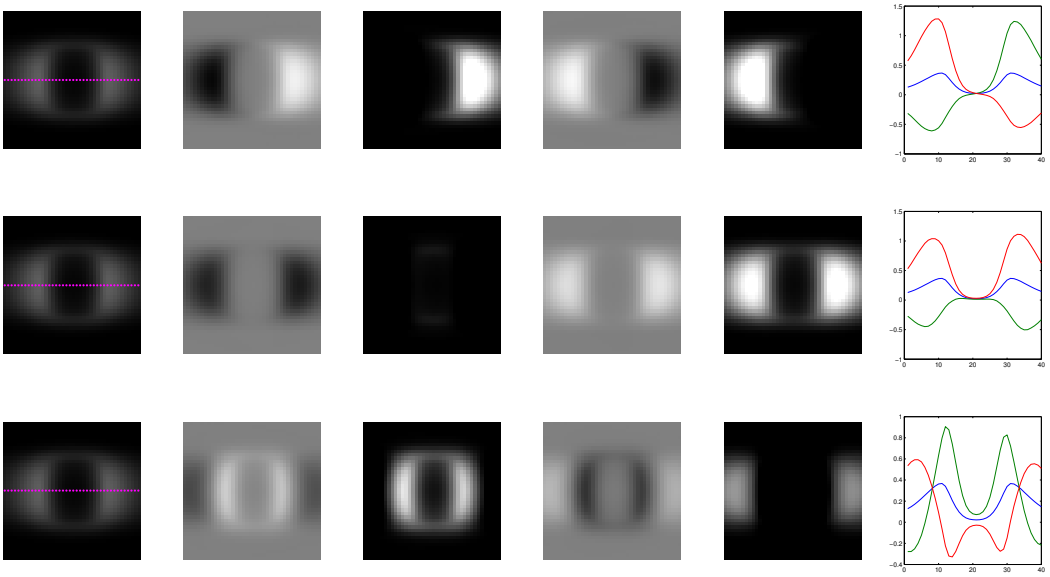


Figure 5.8.: Effect of the first three coefficients for the spine probability model. *First column:* Mean image. *Second column:* 3σ of PC. *Third column:* $\mu + 3\sigma$ of PC. *Fourth column:* -3σ of PC *Fifth column:* $\mu - 3\sigma$ of PC. *Sixth column:* Plot for 1^{st} (*blue*), 3^{rd} (*green*) and 5^{th} (*red*) column at *pink* line.

second and *third column* show the reconstruction received by the dendrite intensity model using 1599 (dimension minus one) and 50 PCs respectively. The *fourth column* shows the ground truth spine probability maps. The *fifth* and *sixth column* show the corresponding prediction of spine probability maps. The figure shows that the feedback for the spine probability maps is weaker in the prediction as in the ground truth. However, there exist correct prediction values in the correct regions (located at spines). Furthermore, in the *bottom row* it is visible that for regions without spine still a strong feedback is given because of the mean in the spine probability model. We can compensate for that problem by not adding the mean or by tuning the system for this base intensity. Another option is to use other features (e.g. the dendrite intensity) to compensate for feedback at locations where the dendrite intensity is (too) low for spines. However, from the statistical point of view it makes sense to have this average spine probability at some locations. Along the backbone already in SBFS-EM data (the real geometry) in about 20 % of the slices a spine is visible. This percentage gets even higher in fluorescence data because of the large PSF. About 83 % of all fluorescence dendrite intensity slices have spine intensity values. The maximal value in the spine probability mean is corresponding to the spine distribution in the training dataset. The more dense spines are distributed along the dendrite the higher are also the values in the spine probability mean. If the spines are distributed more dense than the size of the PSF and oriented in the same direction, then a spine prediction mean value of 1.0 in this locations exists. In that case the prediction of spines would merge to a single large spine. In the prediction map a robust splitting of these touching spines would be required.

In Figure 5.10 we compare the prediction quality for 2PM data and the corresponding ground truth DRFI data. Due to Correlative Light Electron Microscopy (CLEM) we

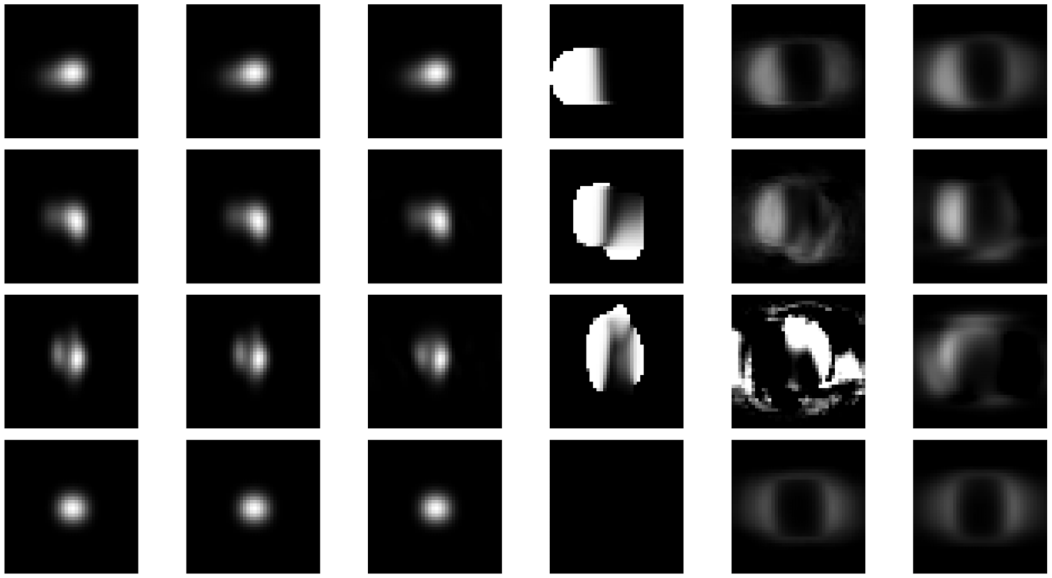


Figure 5.9.: Each row is an example. *First column:* DRFI dendrite intensity slice. *Second and third column:* Dendrite intensity reconstruction from model using 1599 and 50 PCs respectively. *Fourth column:* Ground truth of spine probability map. *Fifth and sixth column:* Spine probability prediction from model using 1599 and 50 PCs respectively.

have of the same piece of dendrite 2PM and SBFS-EM data and thus also DRFI data. By a manual alignment of the correlative dataset we are able to compare similar slices. However, it seems that the SBFS-EM data is rotate along the dendrite main axis about 20°. Therefore, to find the corresponding slices is quite difficult. It is not feasible to find the identical slices in both dataset because they were generated each independent of the other one. Nevertheless, by visual interpretation it is feasible to find the similar slices. We present some of them in Figure 5.10. The first example has a quite good result which is similar to the ground truth. In the *second row* the spine to the right is not predicted. However, in this example slice the dendrite looks quite small and we expect a wrong scaling. In the *third row* the result is not so bad. Some higher prediction values in z-direction (top left) are recognizable and correspond to the ground truth data. These results show that it is feasible to compare synthetic and real data. However, because of the independent imaging it can be quite difficult to get the corresponding 2D images.

In Appendix A, Section A.4, we show further results of prediction for DRFI and 2-Photon Microscopy data.

5.5. Spine Orientation Dependent Models

In general the training data of statistical models should be registered. The (semantic) properties of the objects are in correspondence. Furthermore, all the training data represent the same structures. For the dendrite intensity slices we have two differences. First, an alignment or rotation of the spines to locate all the spines at the same position is difficult. Moreover, this would have some impact on the model because of the PSF. The rotation would have the effect of different oriented z-axis (optical axis) and because

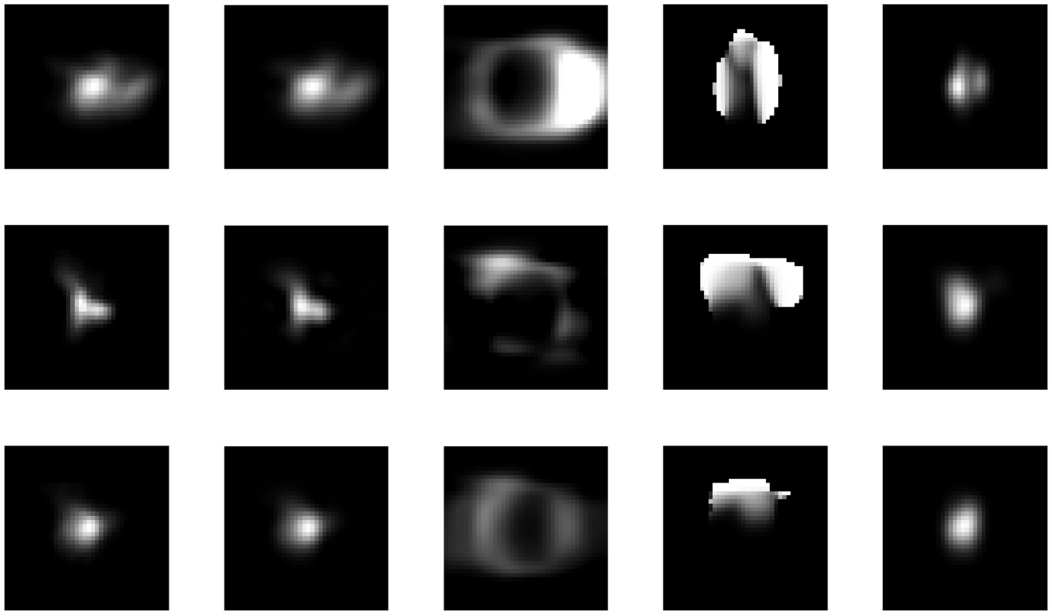


Figure 5.10.: Each row is an example. *First column:* 2PM dendrite intensity slice. *Second column:* Dendrite intensity reconstruction from model using 50 PCs. *Third column:* Spine probability prediction from model using 50 PCs. *Fourth column:* Spine probability to corresponding DRFI slice. *Fifth column:* Dendrite intensity slice of corresponding DRFI slice.

of that the elongated PSF would be mixed. This could have negative effects if the PSF is not compensated. A fully correct compensation would mean to deconvolve the images completely and correctly. However, the deconvolution of fluorescence images is difficult and not completely solved. Moreover, in 2PM the correct PSF is depending on the position within the image (depth). This means that the PSF is not constant over the whole image (see also Section 2.3.3). Even if the rotation is found and the PSF compensated, there is still the problem that the structure of the dendrite slices is not unique. A unknown number of spines can be distributed in arbitrary angles. Therefore, it is not feasible to align the spines correctly in the training data. The effect of not correctly registered training data for statistical models and PCA can be seen in [53]. In this early approach not yet a registration of the images was done. In later approaches all training data is registered and each pixel (sample point or variable) represents the semantic same information. This is a common opinion that by the registration the statistical model becomes more representative and robust.

We introduce spine orientation dependent models of the dendrite intensity and spine probability model. The goal of these spine orientation dependent models is to overcome the issues with not registered spines. Furthermore, they should improve the optimization of the backbone. The backbone optimization often tries to center the backbone in the distribution of dendrite intensity of the slice if only one model is used. This is correct if no spine is there, but for cases with spines it is required that the backbone is optimized more to the opposite side of the spine. The reason for centering the backbone in all cases is that a wrong aligned dendrite intensity slice with spine looks similar to a slice without spine. Furthermore, in the model this centering is then too probable

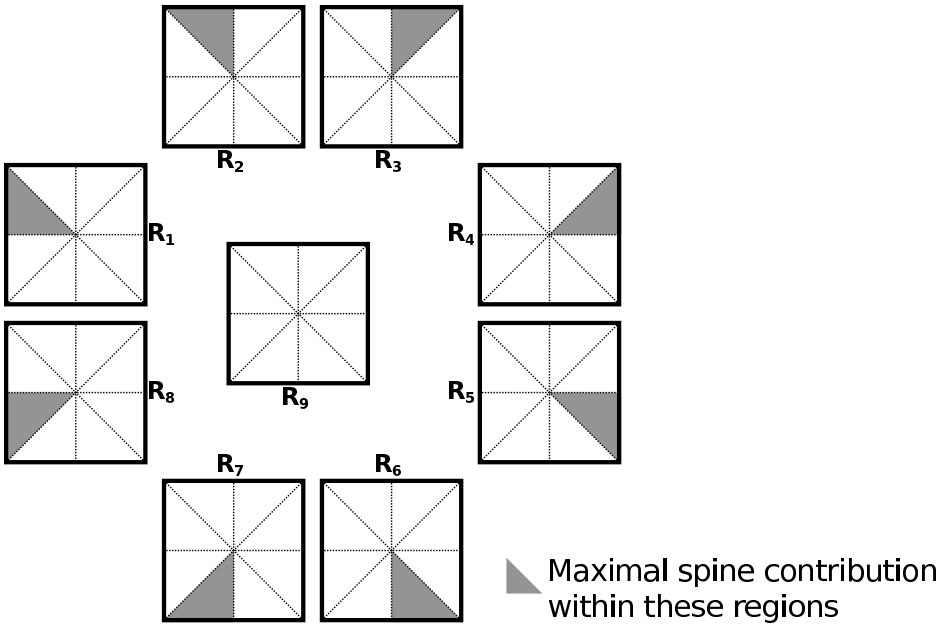


Figure 5.11.: The training data is divided into one of the eight directions or into the group with no or almost no spine contribution.

because it looks similar to the mean image of the model. Therefore, if only one model with all oriented spines is used, the backbone optimization can fail at locations with spines and would then operate similar to thinning approaches (see Section 4.3).

We distribute the training data into 8 spine orientation dependent groups and one without or almost no spine voxels. Figure 5.11 shows the template how the different oriented spines regions are defined. To decide which slice corresponds to which group we compute for every slice how much spine contribution in each region is present. The corresponding region r for each slice is found by:

$$r = \underset{k\in[0,K-1]}{\operatorname{argmax}} \sum_{n=1}^{N} I_s(n) * \theta_k(n) \tag{5.21}$$

where I_s is the spine probability slice ($n \in [1, N]$ the number of pixels) and θ_k are the 8 region templates. The spine region is only computed if the contribution of spines is larger than a given threshold. Otherwise the slice corresponds to the group without spine and belongs to region $K = 9$. The total spine contribution ξ is computed by:

$$\xi = \frac{\sum_{n=1}^{N} \begin{cases} 1, & \text{if} \quad I_s(n) > 0.5 \\ 0, & \text{else} \end{cases}}{N} \tag{5.22}$$

In our approach we used a threshold of 0.025 for ξ. This means that at least 2.5 % of the pixels in the image must have a spine contribution of more than 50 %.

Finally, we have instead of the dataset X_d and X_s the new datasets with spine X_d^1 to X_d^8 and spine probability datasets X_s^1 to X_s^8. Without spine contribution we have the dataset X_d^9 and spine probability dataset X_s^9. From these new spine orientation

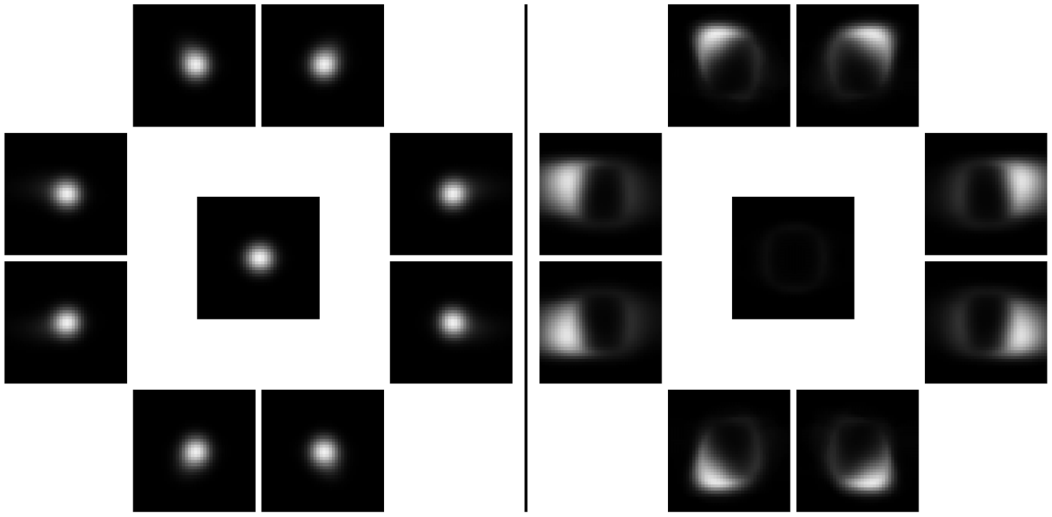


Figure 5.12.: *Left:* The 9 images show the mean of dendrite intensity models (M_d^1 to M_d^9). The images are arranged in the way of the orientation of the largest spine contribution. *Right:* The 9 images show the mean of the spine probability models (M_s^1 to M_s^9). For each model the preference of the picked orientation is in the spine probability well visible.

dependent datasets we can compute the corresponding dendrite intensity models ($M_d^k \in \{M_d^1, \ldots, M_d^9\}$) and the spine probability models ($M_s^k \in \{M_s^1, \ldots, M_s^9\}$).

In Figure 5.12 we show the mean image of the spine orientation dependent models M_d^k and M_s^k. For better visibility of the orientation differences we only display an intensity distribution from 0 to 0.5 and clipped values above for the dendrite intensity model. In the mean images it is well visible how the preferred orientation of the model to the given direction is encoded within the models. This is valid for the dendrite intensity models and the spine probability models in the same extent. The statistical conditions, Equation (5.21) and Equation (5.22), fulfill our expectations to spine orientation dependent models. The conditions generated the orientation distributed models in the way wanted. Nevertheless, the data is not manually distributed into the different groups but by the properties of the data itself. This makes the process expert independent and the idea of the statistical models is also fulfilled in the meaning that the distribution of the training data is depending on the data. The statistical conditions to distribute the data ensure a deterministic approach.

Instead of a single prediction map $s_{s,i}$ per test slice $s_{d,i}$ we compute the spine prediction maps $s_{s,i}^k$ for all model pairs M_d^k and M_s^k. The spine prediction map $s_{s,i}^K$ is the prediction map for the model without or almost no spine contribution. Therefore, the spine probability map $s_{s,i}^K$ is not very useful and can be skipped in practical applications.

The probability of observing a slice $s_{d,i}$ and having model M_d^k can now be computed by:

$$P(s_{d,i}, M_d^{k'}) = \frac{P\left(s_{d,i}|M_d^{k'}\right) P(M_d^{k'})}{\sum_{k=1}^{K} P\left(s_{d,i}|M_d^{k}\right) P(M_d^{k})} \tag{5.23}$$

In the backbone optimization process the probability of Equation (5.23) is used to

optimize the location of the backbone. We expect that the best location of the backbone is where any model has the best representation of the slice. See also Section 4.4 for further details on how this probability was used in the optimization process.

In Appendix A.4 we show some prediction results using the spine orientation dependent models in Figure A.6.

6. Spine Segmentation in Single Time Points

In principle spine segmentation is identical for single time points and time series. However, time series are requiring some more constraints and adoptions which we will discuss in Chapter 7. In this chapter we discuss the segmentation of spines for single time points. As main novelty and core of our spine segmentation we use statistical models from Digitally Reconstructed Fluorescence Images (DRFIs) (see Chapter 5). The decision if a voxel is spine or not is mainly made by the statistical models. The additionally introduced steps are used to improve the decision process. Furthermore, the system should be more robust in practical applications. The combination of prediction results from the models and the additional, backbone-parallel features is in a certain way a multi feature approach.

In this chapter, in Section 6.1, we introduce how the previously introduced dendrite intensity and spine probability models are used as main feature. Then, in Section 6.2, we introduce some backbone-parallel features which overcome the issue that the model does not have any information on changes over multiple slices. In Section 6.3, the possibilities to combine the different features are introduced and the advantages and disadvantages of them discussed. Then, in Section 6.4, we explain the backprojection of the prediction maps into the 3-Dimensional (3D) space. Finally, a binarization of the 3D spine prediction maps is required to get the real spine objects. We will discuss the difference between global and local adaptive approaches in Section 6.5. Results of practical applications are showed in Section 6.6 for single- and multi-channel images.

6.1. Feature from Models

The dendrite intensity and spine probability model encode the dendrite intensity distribution of dendrite slices and which pixels are expected to be from spine. We introduced the idea, concept, data generation and computation of these models in Chapter 5.

6.1.1. Orientation Dependent Dendrite Intensity and Spine Probability Models

We introduced in Section 5.5 the concept of spine orientation dependent models. Depending on the spine distribution the data is distributed into 9 different groups and corresponding models are computed. In the spine segmentation we use the feedback from all models except the one without spine. The feedback is composed to one response. The orientation of the spines to be segmented is unknown. Therefore, we must use a combination of all models and it is not feasible to apply only the corresponding model. If the spine orientation would be known and only the corresponding model is applied, the spine prediction quality would be better. The idea of using only the model which represents the dendrite intensity distribution with the highest probability

has a drawback. If multiple spines are within one slice the result gets very bad. Furthermore, from slice to slice a flickering of the prediction is possible. This is another reason to use the feedback from all models. The feedback can be composed together using the probability a model represents the slice (see Equation (5.17) in Section 5.3). This means that we use the orientation information represented in the different models not only for the backbone optimization (see Section 4.4) but also for spine prediction itself. Another approach would be to use a model covering spines orientated in any direction. However, as motivated in Section 5.5, we expect and have better results with the orientation dependent models.

6.1.2. Spine Prediction by Model

The basic usage of dendrite intensity and spine probability models to predict spines was already introduced in Section 5.3. There we introduced how for a given slice a spine prediction map can be computed. Here, we summarize which steps are done to get the prediction map from the models for a fluorescence image. We also discuss the differences to the steps introduced in Chapter 5. Especially, how additional classifications are used to overcome false predictions of non-spine slices. The following steps are conducted to get the spine prediction from the models:

- Extract 2-Dimensional (2D) slices at all backbone locations.
- Compute the spine predictions using all models.
- Calculate normal weight w_γ.
- Calculate slice spine probability classification $w_s(s_{d,i})$.
- Calculate local adapted classification weight $w_s^K(s_{d,i})$.
- Weight the spine prediction by all weights.

2D Slice Extraction

At all backbone points 2D slices are extracted the same way as introduced for the backbone optimization and described for the models in Section 5.2.4. Also a normalization of the intensity range to $[0, 1]$ is done. This is identical to the step described in Section 5.2.3. The resulting slices are used as input data for the prediction.

Spine Prediction for 2D Slices

The slices are registered as described in Section 5.2.4. Then we predict the spine probability using the dendrite intensity models M_d^k and spine probability models M_s^k. This is done as introduced in Section 5.3, Equation (5.16). We get the predictions $s_{s,i}^k$ where $k = [1, \ldots, 9]$, as the orientation dependent models are used. To improve the prediction quality the prediction maps $s_{s,i}^k$ are weighted by the weights w_γ, w_s and w_s^K.

Weight w_γ

The model is mainly trained with data from horizontal oriented pieces of dendrite. Therefore, we weight the prediction map by the angle of the normal $\vec{n_i}$ relative to the z-axis. We use directly the cosinus value which is given by the scalar product of the normal $\vec{n_i}$ and the optical axis $\vec{z}$. The weighting factor w_γ is then given by

$$w_\gamma(s_{d,i}) = 1 - |\langle \vec{n_i}, \vec{z} \rangle| = 1 - |n_{z,i}| \tag{6.1}$$

where $n_{z,i}$ is the z-element of the normal. The weight w_γ is cosinus shaped and not linear. Slices with a horizontal normal are more reliable than slices with a vertical normal.

Weight w_s

The second weighting factor is the probability of having a slice with spine. Except one model all have a spine contribution. We use the probabilities $P(s_{d,i}, M_d^k)$ to compute if the slice has a spine or not. This is computed by the probability of the models having spine contribution versus the total probability for representing the spine:

$$w_s(s_{d,i}) = 1 - \frac{P(s_{d,i}|M_d^K)P(M_d^K)}{\sum_{k=1}^{K} P(s_{d,i}|M_d^k)P(M_d^k)} \tag{6.2}$$

We remind the reader that the model M_d^K is the model without spine contribution. Theoretically this factor should state if a spine is located in the slice or not. However, experiments showed that in most cases the models M_d^1 to M_d^{K-1} explain all slices much better than model M_d^K, even for cases without a spine. This means that w_s is mostly 1. The models models are using a synthetic, global valid Point Spread Function (PSF) and no noise model is integrated. It is probable that these issues are a reason why all slices are better explained by the models with spine contribution.

Weight w_s^K

The locally adapted weight w_s^K is based on the probability $P(s_{d,i}, M_d^k)$. However, this weight computes the probability of having a spine by only comparing with the model M_d^K which is without spine contribution. This weight exists for all spine orientation dependent models M_d^k where $k \neq K$. The probability represents how probable it is to use the k^{th} model instead of the model without spine. The non-adapted probabilities are computed by:

$$\tilde{w}_s^K(k) = \frac{P(s_{d,i}|M_d^k)P(M_d^k)}{P(s_{d,i}|M_d^k)P(M_d^k) + P(s_{d,i}|M_d^K)P(M_d^K)} \tag{6.3}$$

The weight $\tilde{w}_s^K(k)$ is then adapted to become the weight function $w_s^K(k)$. We rescale the weight between local minimum and maximum to the range $[0, 1]$. The distance between local minimum and maximum must be larger than a given threshold. This suppresses rescaling of to small changes. Figure 6.1 shows a comparison between the unchanged original probabilities and the local adapted ones. Additionally the locations of spines are marked.

Application of Weights

Finally, the spine prediction map $s_{s,i}^k$ is weighted by the products of the weights which is $w(s_{s,i}^k) = w_\gamma w_s w_s^K(k)$. The applied weights try to decide if the computed prediction should be used or not. The weights represent the probability if it is a spine or non-spine slice and how good the models can explain the slice. For the K-th model (without spine) no weights exist and the prediction map is discarded.

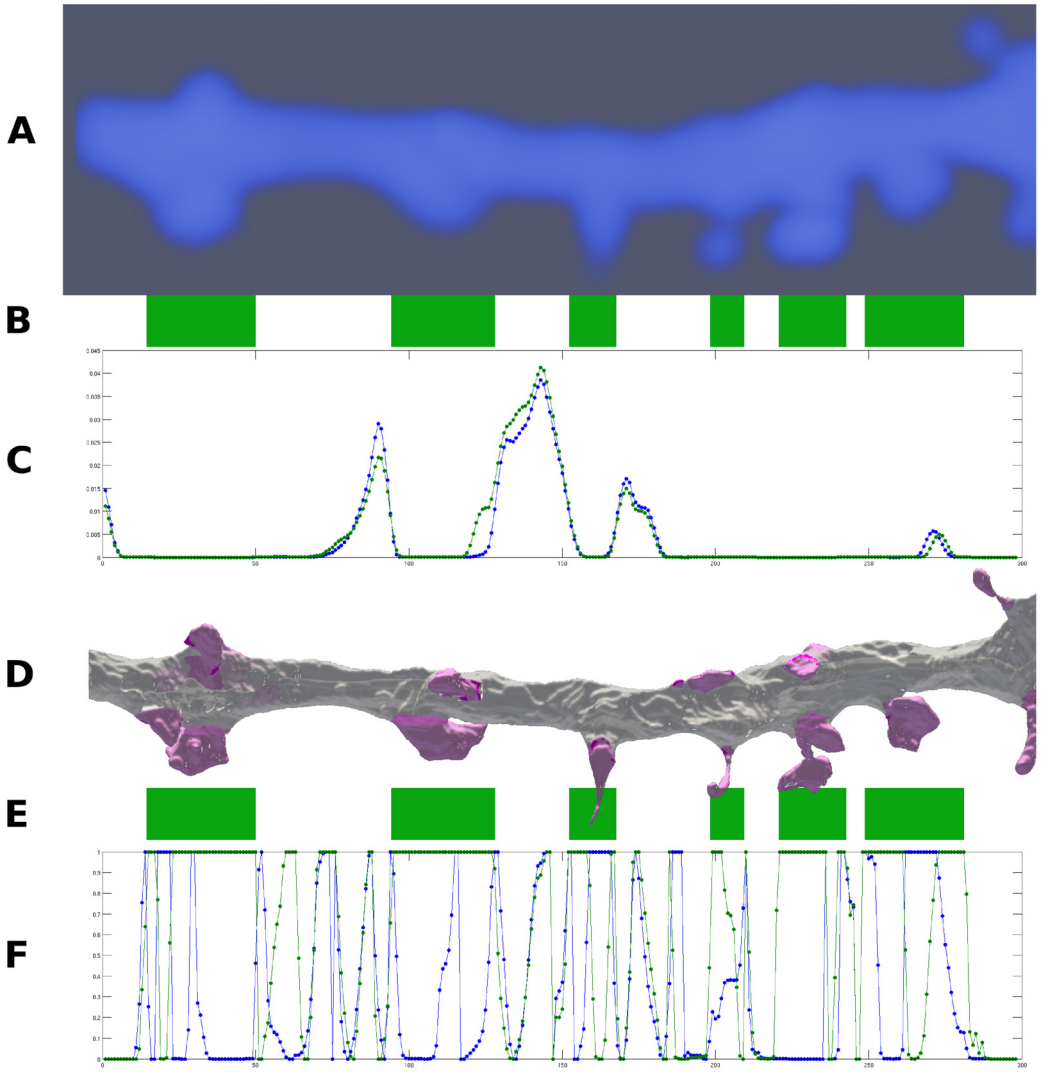


Figure 6.1.: *A:* DRFI of Dendrite6 as test data. *B:* Marker of spines for which models M_d^4 and/or M_d^5 should have positive weights. *C:* Plot of original probabilities for $P(s_{d,i}, M_d^4)$, *blue*, and $P(s_{d,i}, M_d^5)$, *green*. *D:* SBFS-EM reconstruction with spines (*pink*). *E:* See *B*. *F:* Adapted weight w_s^K which is 1 at locations of spines and corresponds to the labels in *E*.

6.2. Backbone-Parallel Features

The robustness of the spine prediction is heavily depending on the alignment and registration of the 2D slices. This is because the used dendrite intensity and spine probability models are requiring good correspondence of the test data. In the practical application with real data it clearly shows that this requirement is not always fulfilled. Already a small misalignment can change the quality of the prediction dramatically. This issue is tried to minimize by the backbone optimization. However, also the optimization can fail. This motivates to use other features too. The use of a single feature would not give a enough robust system for the use with real data in biological research.

No information about neighboring slices nor any backbone-parallel properties are part of the used models. The models have not integrated how dendrite intensity changes along the backbone, neither how the probability maps change in that direction. The models are based on independent slices because of the difficulty of registering 3D volumes. Furthermore, the coupling of multiple slices would become computational expensive. However, we are aware that this information is important. Backbone-parallel properties describe dendrites and spines well. We will introduce why it is so obvious to use backbone-parallel features and what kind of them we use.

Imagine a dendrite without spines. This object could be modeled in a simplified shape by a tube. We can model the dendrite as tube that has curves and changes slowly its radius. The backbone would correspond to the centerline of the tube. A virtual fly through an object like this along the backbone is like driving through a tunnel. Imagine now adding spines to the tube. The change of radius for spines is quite fast compared to the slow change of radius of the dendrite. Furthermore, the radius changes only in one direction and not in all directions. Spines are somehow high-frequency, disturbing objects along the dendrite. In Electron Microscopy (EM) reconstructions spines appear within 2 or 3 slices in the virtual tunnel fly. This corresponds to a very high frequency. In rays parallel to the backbone the gradient change for spines can be easily detected. For this the geometrical correct shape and the backbone have to be known. The backbone is known due to the initialization and the backbone optimization. However, we do not have the geometrical correct shape, except for the training data. We have to process fluorescence data. In this data the high frequency changes introduced by spines are very smoothed. The difference between spine changes and dendrite changes become much smaller. Nevertheless, we transfer the idea of detecting high-frequency changes parallel to the backbone to the 2-Photon Microscopy (2PM) data.

In fluorescence images all edges are smoothed strongly by the PSF. Instead of high-frequency changes we have much lower gradients of the intensity change. Furthermore, also in the core of the dendrite intensity changes are feasible. These changes are mainly caused by enclosed organelles. The situation changed such that instead of no changes in the dendrite core and high-frequency changes at spine locations, we have now possible, medium changes in both regions. This means that the difference between the dendrite core and appearing spines is much smaller in fluorescence images. The use of the frequency changes is more challenging than in EM reconstructions.

We will use similar features in the registered and unregistered slices to detect the high-frequency changes. The computation of features in both spaces (registered and unregistered slices) is done because all slices are independently registered. In both spaces different results can be achieved. We have different challenges in both spaces.

In the unregistered space the change of the dendrite itself can be misinterpreted as spine. This is because the surface of the dendrite must not be parallel to the backbone. However, we try to compute backbone-parallel features regardless of that fact. The registration of the slices compensates this issue. The surface of the dendrite is parallel to the backbone. However, we compute all the registrations independent of each other. A smoothing of the registration parameters tries to ensure a continuity of the registration along the backbone. If this is not successful artifacts and wrong high-frequency changes can be introduced. Therefore, we use backbone-parallel features of both spaces to overcome the issues.

6.2.1. Features in Registered Slice Space

In principle we use some kind of gradient. In difference to classical gradient computation we use a version normalized by the intensity. Furthermore, the distance at which we measure the intensity values is variable. The gradient is independently computed in both backbone directions. The intensity from spine pixels away always decreases in both directions. The computed gradients are normalized by the local intensity. This normalizes the feature value with respect to the local intensity. Hence, also weak spines could generate high feature values. The backbone-parallel feature bpf_r for a single pixel in slice $s_{d,i}$ at pixel (x, y) of the slice is computed by:

$$\text{bpf}_r(x, y, s_{d,i}) = \frac{2 * s_{d,i}(x, y) - \min_{l=1}^{L} \{s_{d,i-l}(x, y)\} - min_{1=1...L}\{s_{d,i+l}(x, y)\}}{s_{d,i}(x, y)} \quad (6.4)$$

where $s_{d,i\pm l}$ are the neighboring slices. We use the range $l = [1, 15]$. The distance between the slices is given by the backbone sampling distance which is $0.02\,\mu\text{m}$.

6.2.2. Features in Unregistered Slice Space

In the unregistered slice space we use in principle also Equation (6.4). However, in difference to the slices $s_{d,i}$, which are aligned, we use the unregistered slices in this approach. We name the unregistered slices $\tilde{s}_{d,i}$ and get the following equation for the backbone-parallel features bpf_u in the unregistered space:

$$\text{bpf}_u(x, y, \tilde{s}_{d,i}) = \frac{2 * \tilde{s}_{d,i}(x, y) - \min_{l=1...L} \{\tilde{s}_{d,i-l}(x, y)\} - min_{1=1...L}\{\tilde{s}_{d,i+l}(x, y)\}}{\tilde{s}_{d,i}(x, y)} \quad (6.5)$$

In the unregistered space the sampling along the neighboring slices corresponds to sample along a curve which is parallel to the backbone. Therefore, this curve is described by the normals $\vec{n_i}$. In contrast, in the registered space the parallelism of the curve to the backbone is not guaranteed.

6.3. Combination of Different Features

The combination of different features aims for a more powerful and robust system. The overall goal is that one feature compensates with its advantages the disadvantages of a different feature. All the features are computed independently of each other.

We combine the different features to a single feature map or prediction map. Most obvious approaches are to add or multiply the feature values. Also a combination of both approaches is feasible by using a hierarchical structure. However, we will not discuss hierarchical approaches. The additive approach corresponds to average the feature values. The extreme values are smoothed out. In contrast, the multiplicative approach is more sensitive to the variance of all feature values. We will introduce both approaches. Furthermore, we will explain the decision to use a multiplicative approach. We also discuss the issue of feature value range and normalization.

Additive Combination

Summation or computing the mean value of features corresponds to additive combination. The combined feature value ζ_Σ computes by summation:

$$\zeta_\Sigma = \sum_i \mathrm{fv}_i \tag{6.6}$$

The different feature values are denoted by fv_i. The additive combination of feature values does not care about the variance of all feature values. This means that we neither penalize small feature values nor reward high feature values. The variance of the distribution is completely ignored.

If the features have different ranges they must be normalized to have the same impact on the result. A normalization is described in Section 6.3.

Multiplicative Combination

The use of a multiplicative combination of feature values is well known for probability trees. The overall probability along a path in the tree is calculated by multiplication. For features with a value range ≥ 0 a multiplication can be applied. The multiplicative overall feature value ζ_Π is calculated by:

$$\zeta_\Pi = \prod_i \mathrm{fv}_i \tag{6.7}$$

However, if the ranges are not identical the interpretation can become quite complex. In difference to the additive approach the variance is very important in the multiplicative approach.

The multiplicative approach decreases the False Positive (FP) rate because already one negative voting feature (≈ 0) can suppress all others values. However, the False Negative (FN) rate can increase too. A single high feature value (≈ 1) is not able to get a final True Positive (TP). Depending on the wanted goal (high recall or precision) the multiplicative approach has clear advantages. High precision can be achieved by the the multiplicative combination because the FP rate is decreased by the effect of the variance. At the same time the recall can decrease because of the higher FN rate.

The terms recall and precision are mainly used in pattern recognition and information retrieval. We introduce them using the terms defined in Table 6.1. Recall is defined as following:

$$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6.8}$$

Term	Description
True Positive (TP)	A True Positive is a classification result that labels something as positive (belonging to the class) which is also part of the class.
True Negative (TN)	A True Negative is a classification result that labels something as negative (not part of the class) which is also not part of the class.
False Positive (FP)	False Positive is a classification that labels something as part of a class. However, the object is not part of the class. This is a wrong classification.
False Negative (FN)	False Negative is a classification as not part of the class. However, the correct classification would be part of the class.

Table 6.1.: Classification results can be divided into four categories.

A high recall is received by returning everything as positive and then surely no FN would be present. In difference to that precision cares if a high FP rate is present. Precision is defined by:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.9}$$

To get a high precision no FPs should be present. This is achieved by using only strongly positive feedback of the classifier (≈ 1). This ensures a low FP rate and the precision converges to the maximum.

To achieve a high precision and recall at the same time is more difficult. However, exactly this is wanted. For further explanation about precision and recall we refer the reader to the literature or the Wikipedia entry about precision and recall [57]. Also further measurements like Accuracy or True Negative Rate are explained there.

Feature Value Range and Probabilities

Feature values can have different value ranges. Depending on the computation any feature range is imaginable (also negative values). This makes it difficult to combine different features. Combining features with different value ranges can lead to unwanted effects on the final classification. Therefore, a normalization of the intensity ranges can be useful.

Probabilities have an range of $[0, 1]$. This range is very intuitive and supports to combine features. Furthermore, the possibilities to combine probabilities are well studied. The importance of graphical models in modern approaches emphasizes the advantages of using probabilities. By normalization likelihoods or feature values can be transformed to probabilities. Then, the concepts of probability theory can be used. This motivates to use probabilities or normalized feature values instead of raw feature values.

Normalization of Features Values

Normalization is the approach of mapping a value distribution to a given value range. This can be the range of probabilities, $[0, 1]$, or to any other range. If multiple features have the same (normalized) value range the interpretation of weights is simpler.

The normalization to the range $[0, 1]$ is heavily used. It is calculated for a value v_i by:

$$\text{normalized}(v_i) = \frac{v_i - \max(V)}{\max(V) - \min(V)} \tag{6.10}$$

where v_i is a value and V is the set of all values. This normalization is a linear transformation.

Weighting of Features

It is possible to weight features. If the feature ranges are normalized the weights are directly intuitive. Otherwise the weights are not very intuitive. To understand the effects of the weight the different value ranges must be considered. In the additive combination of features weights can strength or weak the importance of each feature. The additive combination changes to:

$$\zeta_\Sigma = \sum_i \alpha_i \text{fv}_i \tag{6.11}$$

where α_i are the weighting values. Normally $\alpha_i \geq 0$ is used. The same application of a weighting in the multiplicative approach has a different effect. The weights merge to a constant C. The weighting is just a linear, constant scaling:

$$\zeta_\Pi = \prod_i \alpha_i \text{fv}_i = \prod_i \alpha_i \prod_i f v_i = C \prod_i f v_i \tag{6.12}$$

Therefore, weighting is not meaningful in all approaches. Often weighting corresponds to a tuning of results and is very subjective.

6.4. Backprojection to 3D Image

We extracted 2D slices from the original 3D fluorescence image. We compute our features in registered and unregistered slices. However, the prediction and segmentation of spines is needed for the 3D fluorescence image. Therefore, all information is transformed back to the 3D space. We call this process backprojection of the prediction.

The following steps compute the final 3D prediction map:

- Invert registration steps.
- Transform pixel information from 2D to 3D using plane information.
- Merge overlapping information form multiple slices.
- Rasterization.

Inversion of Registration

It can be a very difficult task to invert a registration. To invert a registration means to invert the vector field or in other words to invert the transformation. For most registrations this is not straightforward. Our task is much simpler because we applied as transformation a scaling into the direction of the x- and y-axis of the 2D image. We can compute the inverted transformation matrix directly from the original transformation matrix:

$$T^{-1} = \begin{bmatrix} s_{\text{horiz}} & 0 \\ 0 & s_{\text{vert}} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{s_{\text{horiz}}} & 0 \\ 0 & \frac{1}{s_{\text{vert}}} \end{bmatrix} \tag{6.13}$$

where s_{horiz} and s_{vert} are the horizontal and vertical forward scaling respectively. The transformation is applied to the image space of the 2D images. The origin is at the center of the image.

Transform Pixel

We want to find the original position in the 3D space of each pixel in an unaligned slice. We sampled from the 3D image using planes described by normals $\vec{n_i}$, the plane axes $v\vec{1}_i$ and $v\vec{2}_i$ and the backbone position p_i. The x and y pixel positions of any 2D slice describes the 3D point $p_{i,x,y}$ by:

$$p_{i,x,y} = p_i + \alpha(x)v\vec{1}_i + \beta(y)v\vec{2}_i \tag{6.14}$$

The parameters α and β move within the plane at point p_i that has the normal $\vec{n_i}$. α and β can be used to describe the pixel position within the 2D slice because they scale each one plane vector. The relation between a given α and the corresponding x is given by:

$$x(\alpha) = \frac{\alpha}{\text{spacing}_x} + \frac{r}{2 \cdot \text{spacing}_x} \tag{6.15}$$

where r is the radius (half side length) of the extracted 2D slice. The parameter spacing_x describes the image spacing in x-direction. In the opposite direction we have the following equation:

$$\alpha(x) = \left(x - \frac{r}{2 \cdot \text{spacing}_x} \right) \text{spacing}_x \tag{6.16}$$

The relation of α and x as well as of β and y leads directly to the position in the 3D space which is described by Equation (6.14).

Combination of Overlapping Slices

All slices along the backbone are independently extracted from the 3D image and projected back. It is possible that in the 3D space multiple slices intersect and describe the same voxel. This means that information from the different slices should be written to the same voxel in 3D. Ideally the classification results of intersecting slices have the same prediction value for the corresponding voxel. However, in practice this is not true. Main reasons are the independent predictions and variable registration parameters used. A combination of the different predictions is required. The simplest approach is to use

the minimal, maximal or averaging value. We use the maximum projected back from any 2D prediction map.

Rasterization

Backprojection of points from prediction to the 3D image corresponds to a forward warping approach. This means that each projected point is a vertex in the 3D space. Forward warping requires to raster in the 3D space using the projected vertices. It is possible that not for every voxel a prediction value exists and gaps exist if no rasterization is done. Gaps in the background are no problem. However, gaps in the foreground can mean to have spines with holes or separated spines. A rasterization of the triangles built by the vertices is required if no additional assumptions are made. The rasterization can be skipped if the points are projected from a more dense sampled space.

In our application we sample the 2D slices more dense than the original 3D fluorescence images. Also along the backbone the sampling is much denser than the resolution of common fluorescence images. Therefore, we skip the rasterization. A smoothing of the 3D prediction map will partially overcome issues if in spite of the preconditions some gaps are present. However, we like to emphasize that depending on the application and data a rasterization is required for forward warping and cannot be skipped.

6.5. Binarization of Prediction Maps

The prediction maps are not binary images. They have high intensities (probabilities) at locations where it is likely to have a spine. However, the classification into spine and non-spine is not yet done. The binarization step generates the real spine objects and spine candidate objects. Statistical values like size and average intensity can be computed for these objects.

The binarization can be done globally (one condition for the whole image) or locally (local adaptive conditions). Both approaches have specific advantages and disadvantages. The goal of the binarization can influence the choice of a global or local approach. We introduce a global and local approach to binarize the prediction map. However, we clearly favor the local adaptive approach because of the properties of the prediction map.

6.5.1. Global Binarization

A global binarization uses the same decision function over the whole image. The decision is not depending on the local situation of the prediction map. If information is used from the prediction map then this is in general a global statistical value (e.g. the average intensity of the prediction map). In Table 6.2 we list some of the most used global binarization approaches with a short description.

Global binarization approaches are faster than local adaptive approaches. Depending on the application, it can be very important that over the whole image the same condition is used. In cases like this a global binarization must be used. Global binarization is less powerful as local adaptive approaches. Global approaches are not able to handle images where the semantic meaning of intensities is not identical over the whole image. It would not be possible to use a lower threshold for weakly detected spines in spine

Method	Description
Fix Threshold	A fix threshold divides the pixels into two classes. The same threshold value is used to classify all pixels. For real probabilities this is often used. The threshold can be defined previously or depending on other results of the algorithm.
Fix Lower and Upper Threshold	Similar to fix threshold approach. However, here two thresholds are used. All pixels with a value between the lower and upper threshold belong to one category and all other pixels to the other group. This approach enables to define classification ranges with two defined limits. In the approach with one threshold the image maximum or minimum can be seen as the second limit.
Otsu Thresholding	Otsu Thresholding [39], also Otus's Method, solves the problem of finding a threshold fulfilling the condition that the mean of both classes are as far away from each other as possible and the variance of the group is as small as possible.
Multiple Otsu Threshold	Uses the same approach like Otsu Thresholding but tries to find multiple thresholding values to divide the intensity range into more than two classes.

Table 6.2.: Overview of standard and often used global binarization approaches.

prediction maps. Using a global threshold can imply to lose weak spines. If a very low threshold is used to keep weak spines, then it is possible to segment too much volume of spines with high detection values. In situations like this a local adaptive approach is much more powerful. A local adaptive approach fulfills our requirements that we have for the automatic spine segmentation.

The computational cost of global binarization approaches is linear depending on the number of pixels or voxels and is $\theta(n)$.

6.5.2. Local Adaptive Binarization

Local adaptive binarization approaches adapt to semantic changes of intensities over the space of the image. Prediction maps have regions with strong and weak detections results. Local adaptive approaches make a decision for a local window and not for the whole image at once. The computational cost increases dramatically because of the local window. The larger the local window is, the higher the cost become. Given a window of size m the complexity changes from the global binarization with $\theta(n)$ to $\theta(nm)$ for local adaptive approaches. The global binarization approaches can be used as local adaptive versions. It is only required to compute their conditions within the local window instead of the whole window. Therefore, we will not introduce local adaptive approaches in more detail. However, we will introduce our concept of local adaptive binarization that we designed for the spine prediction maps.

Our approach is based on the following steps:

- Detection of seed points.
- Removing of too weak seed points.
- Local adaptive Otsu thresholding.
- Splitting of touching spines.

Detection of Seed Points

The prediction map has intensity peaks located at spines. However, these peaks have a large intensity range. We locate first seed points. The seed points are local maxima in the prediction map. The spine prediction map is preprocessed by a Gaussian filter. Then, all local maxima are used as seed points. The spine prediction maps have no feedback in the background because we predict spines along the backbone for the extracted slices. Furthermore, due to the backbone-parallel feature we can ensure not to have FPs signals in the background. Weak seed points can be removed if required. However, in time series analysis wrong weak spine feedback is removed automatically because not over all time points a wrong feedback is present at the same location. Therefore, the wrong feedback cannot be tracked and is discarded. In Figure 6.2, *bottom*, we show a prediction map and the corresponding local maxima that are used as seed points. Figure 6.2, *top*, shows the corresponding fluorescence image input.

Removing of Weak Seed Points

If the number of FPs should be as small as possible, then, it is useful to reduce already the number of seed points. Seed points with weak intensities are most likely FPs. If the seed points are not filtered a huge range for the intensity of seed points is possible. A simple thresholding depending on the mean or maximum seed point value removes

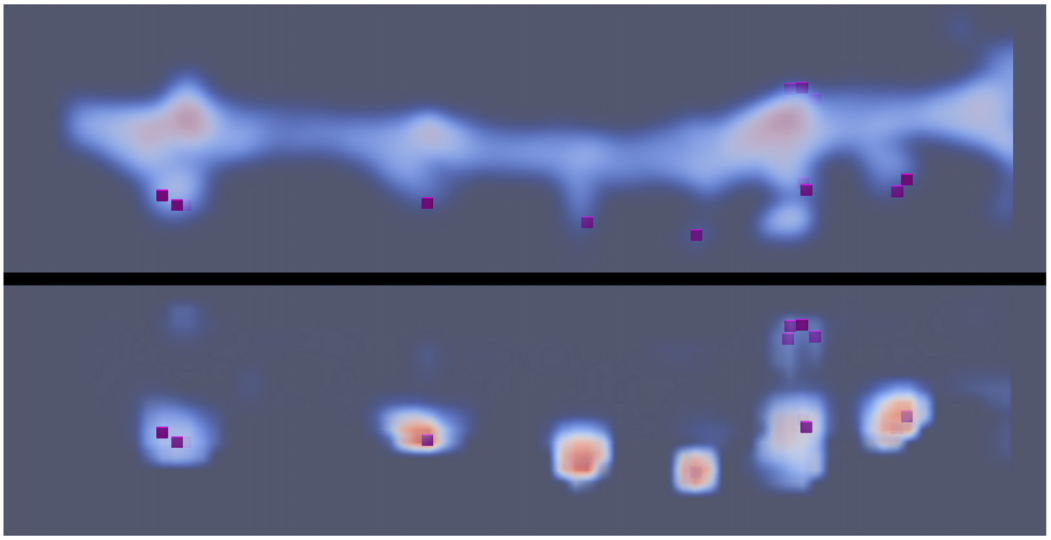


Figure 6.2.: *Top:* Volume rendering of DRFI of Dendrite6 with the automatically detected seed points (*pink*) at local maxima of the prediction map. *Bottom*: Volume rendering of spine prediction map and seed points.

too weak seed points. In practical approaches we use the criteria of a threshold relative to the maximal intensity of the seed points.

Local Adaptive Thresholding

For each seed point a threshold value is computed in a local window (e.g. 2 µm radius). Because of the elongated PSF in the optical axis the local window should be elongated in this axis or be enough large in general. The threshold value computed corresponds to a good binarization of the local spine and its surrounding. Experience showed that a relative threshold value works quite well. We use a threshold value relative to the seed point intensity. The use of 70 % of the seed point intensity gives good results. Moreover, only the connected object that contains the seed point is kept. Other objects within the local search window are discarded. More complex thresholding conditions, e.g. Otsu thresholding, are also applicable. However, the computational complexity is unnecessarily increased. It can happen that the local search window overlaps with other local search windows. In this case it must be decided for every voxel of the binarized object to which spine it belongs.

Splitting of Touching Spines

Spines can be distributed very dense along the dendrite. It is also possible that FPs exist at local maxima. Therefore, seed points can be located near to each other and the search windows of the binarization overlap. In cases like this it is important to decide to which object (spine) the voxels belongs. We apply a distance condition for each voxel that is occupied from multiple objects. The voxel belongs to the nearer seed point. This is only a geometrical separation. However, the exact separation plane is not influencing that value because biologists are mainly interested in the top percentage intensity of the spine.

Dataset	# Spines	TP	FP	FN	Precision	Recall
Dendrite4 - DRFI	15	10 (11[7])	3	5	0.77	0.67
Dendrite6 - DRFI	16	11	0	5	1.0	0.69
Cumulative - DRFI	31	21 (22[7])	3	10	0.88	0.68
Dendrite4 - 2PM	15	13	3	2	0.81	0.87
Dendrite6 - 2PM	16	12	4	4	0.75	0.75
Cumulative - 2PM	31	25	7	6	0.78	0.81

Table 6.3.: Quantitative results of testing the CLEM dataset with Dendrite4 and Dendrite6.

More complex splitting conditions could use information from the prediction map and the original intensity image. In both images the separation plane could be optimized to be located at the local minimum between the seed points. These minima are found by using gradient images. However, a significant intensity minimum does not exist for dense located spines because of the PSF. Therefore, even by using the spine prediction map and the fluorescence intensity the geometrical distance still would be an important separation criteria.

6.6. Results

In this section we show results for test data with a single time point. First we will evaluate the performance of our system on the Correlative Light Electron Microscopy (CLEM) data. Then we show more practical test sets which demonstrate the use for biological experiments.

6.6.1. Correlative Data

Here we present results for the two pieces of dendrite for which CLEM data exists (Dendrite4 and Dendrite6). We use the reduced model ModelAB which is composed of DendriteA and DendriteB. The model is reduced to the 10 first Principal Components (PCs). The used PSF of the DRFIs has as parameters: Numerical Aperture NA = 0.8, wavelength $\lambda = 810$nm and refractive index $\eta = 1.42$. In the 2PM data the volume is fluorescent. Quantitative results of the spine detection and segmentation are presented in Table 6.3. Additionally, we present a quantitative evaluation of the spine detection and segmentation for the horizontal oriented spines in Table 6.4. In both settings (all spines or only horizontal spines) the precision rate is higher for the DRFI data. This was expectable because the intensity distribution in 2PM is much more irregular and the Signal Noise Ratio (SNR) worse. Moreover, in DRFIs no noise is added what simplifies the task.

In Figure 6.3 we show the result for the DRFI of Dendrite4. In the prediction map the five larger spines have a very strong feedback. All other spines have a much weaker feedback. There exists also feedback at wrong locations. This feedback is the challenge of the binarization and gives FPs. However, in the prediction map an intensity difference between the weak TPs and FPs is recognizable. This means that the main

[7] one spine is represented by two detections

Dataset	# Spines (horiz.)	TP	FP	FN	Precision	Recall
Dendrite4 - DRFI	7	7 (8^7)	3	0	0.7	1.0
Dendrite6 - DRFI	11	8	0	3	1.0	0.73
Cumulative - DRFI	18	15 (16^7)	3	3	0.83	0.83
Dendrite4 - 2PM	7	7	3	0	0.7	1.0
Dendrite6 - 2PM	11	10	4	1	0.71	0.91
Cumulative - 2PM	18	17	7	1	0.71	0.94

Table 6.4.: Quantitative results of testing the CLEM dataset with Dendrite4 and Dendrite6. However, only the horizontal oriented spines are taken into account.

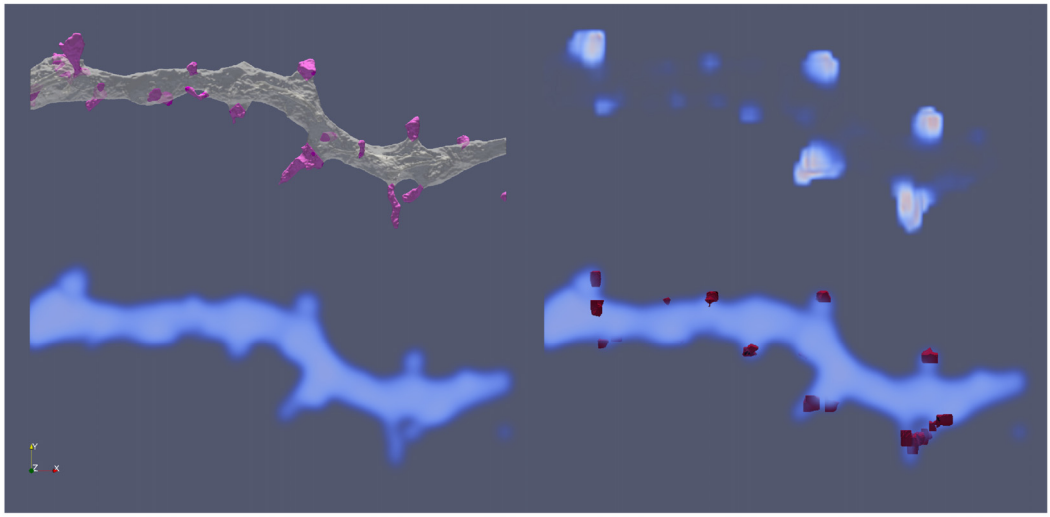


Figure 6.3.: *Top, left:* Reconstruction of Dendrite4. *Bottom, left:* Input data as DRFI. *Top, right:* 3D prediction map. *Bottom, right:* Spine candidates (*red*) and input image.

issue is the tuning of the parameter which picks the seed points for the binarization. One TP is not visible in the volume rendering because it is oriented in negative z-direction and completely occluded by the dendrite volume rendered.

In Figure 6.4 we show the result for the 2PM data of Dendrite4. The result is quite similar to the result of the DRFI analysis. However, for more spines a strong feedback is given. Main reason seems to be a more vertical orientation of the spines. We have the impression that in the SBFS-EM data the spines in the left part are rotated to a more vertical orientation. In principle this should not be the case. However, transformations like this can happen during fixation and further preparation. This dataset has high signals in the background which cause some FPs.

Figure 6.5 shows the results for the DRFI data of Dendrite6. For this dataset the results are very well. There are no FPs at all. Furthermore, some of the FNs are vertical oriented spines. The FNs of horizontal spines have a low feedback in the prediction map. The threshold value for the seed points of the binarization is too high. An even

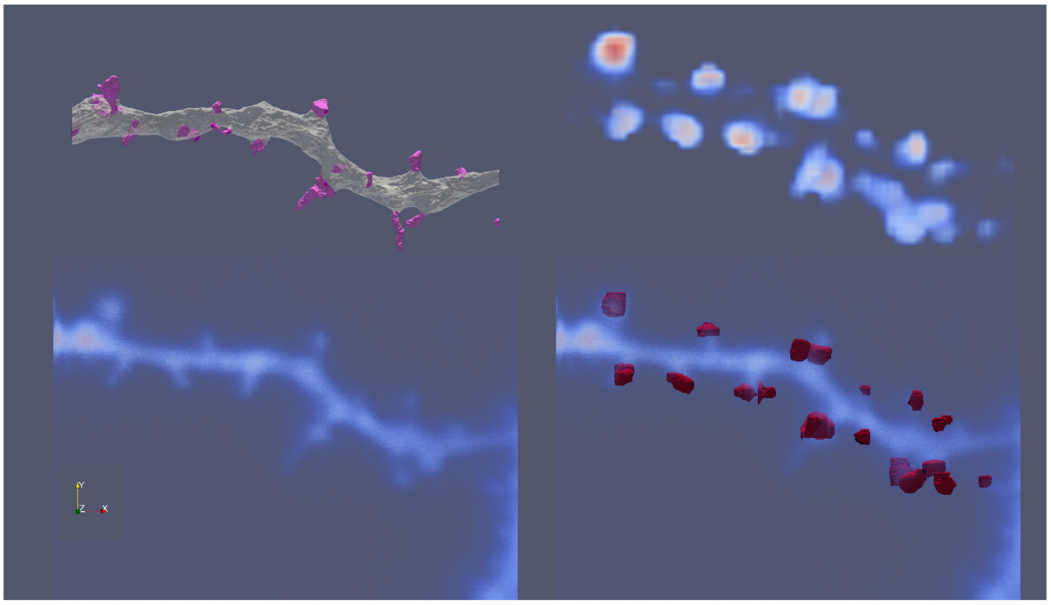


Figure 6.4.: *Top, left:* Reconstruction of Dendrite4. *Bottom, left:* Corresponding input data from 2PM. *Top, right:* 3D prediction map. *Bottom, right:* Spine candidates (*red*) and input image.

better result could be achieved by parameter tuning.

In Figure 6.6 we show the results for the 2PM data of Dendrite6. The results are comparable to those of the DRFI data. Compared to Dendrite4 2PM data, also in this result the FPs have a strong signal but weaker than for most TPs. Also for Dendrite6 the tuning of the binarization can improve the result.

6.6.2. Multiple Channels

In this section we demonstrate the performance of the system on multi-channel 2PM data. The goal is to show the application of our method to data from biological experiments. Moreover, it demonstrates the advantages of using an automatic method

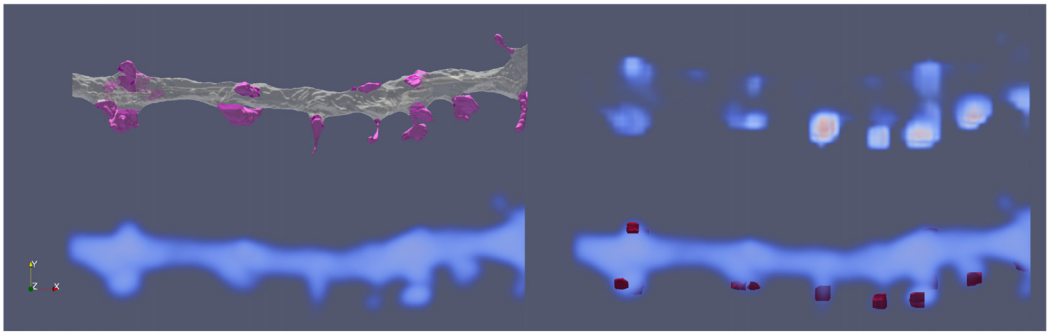


Figure 6.5.: *Top, left:* Reconstruction of Dendrite6. *Bottom, left:* Input data as DRFI. *Top, right:* 3D prediction map. *Bottom, right:* Spine candidates (*red*) and input image.

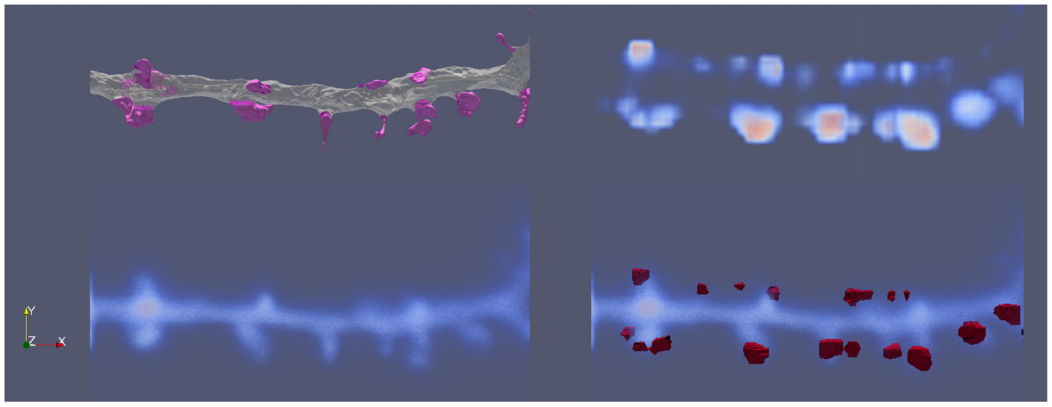


Figure 6.6.: *Top, left:* Reconstruction of Dendrite6. *Bottom, left:* Corresponding input data from 2PM. *Top, right:* 3D prediction map. *Bottom, right:* Spine candidates (*red*) and input image.

to analyze the data. These datasets are completely independent from the CLEM data. Therefore, we use the model ModelAB46 which is trained by all pieces of dendrite (Dendrite4, Dendrite6, DendriteA and DendriteB).

Endoplasmic Reticulum Classification

The Endoplasmic Reticulum (ER) is of importance for the generation of synaptic plasticity. The spine apparatus is part of the ER. Therefore, biologists are interested to classify spines into spines with and without ER. In Figure 6.7 we show a Maximum Intensity Projection (MIP) of the input data. The *red* channel marks the volume (volume filling dye). The *green* channel has a marker for the ER which is visible in spines having ER. At locations with signal from both channels the color is *yellow*. This means that yellow spines should be classified as ER spines and the other ones as non-ER spines.

In Figure 6.8 we show the results of the automatic approach. *Left* a MIP with overlaid spine detection locations is shown. The volume- versus ER-intensity plot (Top5Percentage mean intensity, see Section 7.2.5, Table 7.2) is shown in Figure 6.8, *right*. Spines *1, 2, 3, 8, 12, 13, 18, 21, 33* are classified as spines with ER if the threshold is at 0.1 for the ER channel. Spine *1/2* and *12/13* are double detections. Spine *8* is a FP which is located on the dendrite. However, all spines that have an ER are correctly detected and we have only spine *8* as FP. No spine with ER is missed (no FN).

Bouton Colocalization Detection

In this task the goal is to detect and segment spines in the volume channel of a fluorescence image. In the second channel presynaptic boutons are labeled. Boutons are located along axons. It is interesting to know which spines have a presynaptic bouton. We try to detect the spines with a presynaptic bouton automatically.

In Figure 6.9 the spine with a presynaptic bouton (*white circle*) is manually labeled by a biologist. He verified that this spine has a presynaptic bouton and is a functional spine. The verification that the spine is responding was done by calcium imaging

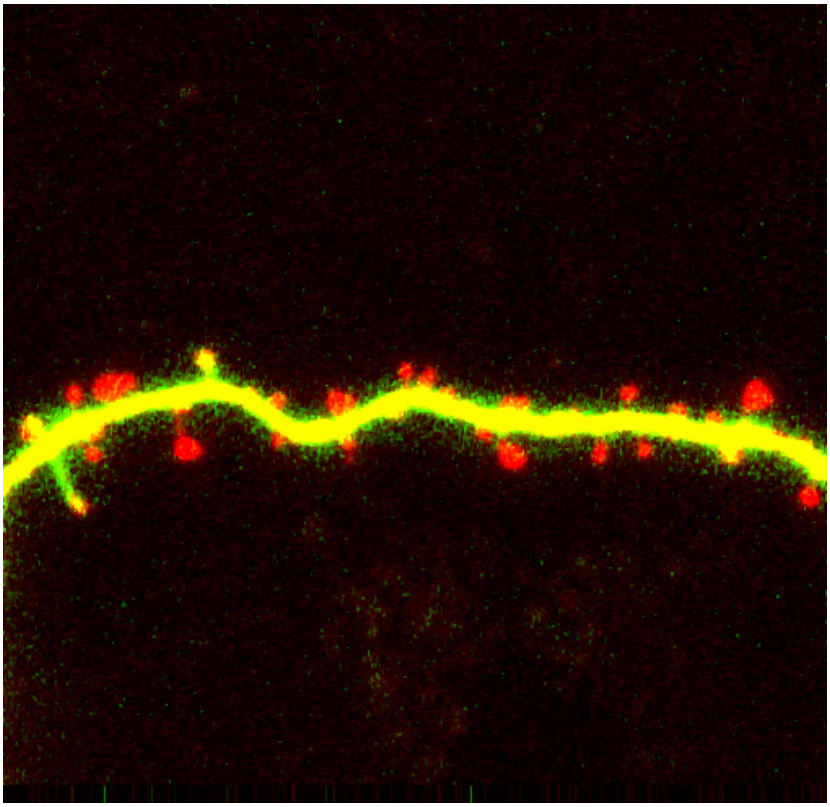


Figure 6.7.: MIP of input data for detection of spines with ER. The volume channel is in *red* and the non-volume channel (ER) is in *green*. Pixels with both signals are *yellow*.

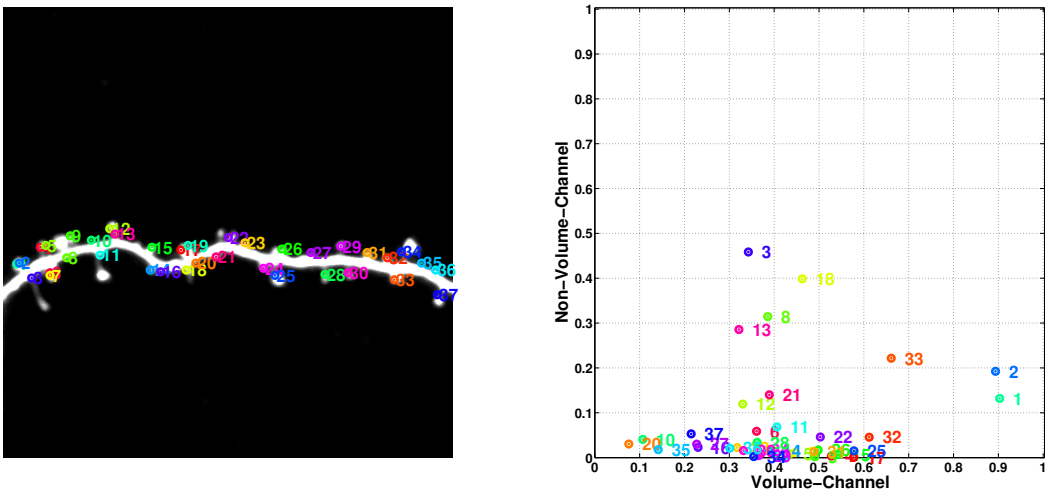


Figure 6.8.: *Left:* MIP of input data with overlaid location of spine detections. *Right:* For each spine the volume versus non-volume intensity is plotted. Spines with ER have in the non-volume channel an intensity > 0.1.

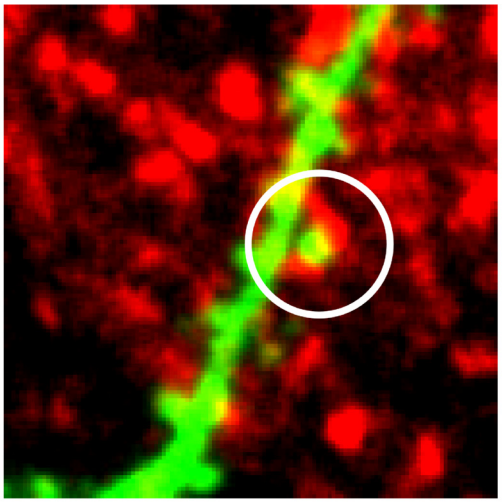


Figure 6.9.: A biologists manually evaluated the dataset. He was aware which axon was stimulated that crosses the dendrite. He manually verified that at the location marked (*white circle*) the spine has a presynaptic bouton. The image showed is a MIP. A signal in both channels gives *yellow* in the MIP.

which we will not discuss in more detail. However, the colocalization of spine and bouton without further controls do not certify a presynaptic bouton.

The goal is to get the same result using automatic analysis. We use the automatic spine detection and segmentation software to detect spines. For each spine in both image channels statistical values are computed. We use the normalized Top5Percentage mean intensity (see Table 7.2 in Section 7.2.5) of each channel as the spine values. In Figure 6.10, *left*, the locations of spine detections are marked. The *right* panel shows the plot of the volume channel intensity versus the non-volume channel intensity (presynaptic bouton marker) for each spine. It is well visible that spine number *5* has a high intensity in both channels (volume and bouton marker). Therefore, we can identify this spine as the active spine. The task is successfully solved. However, the colocalization of high intensity in both channels is only possible because of the large PSF. In principle the spine and the presynaptic bouton are near together but not overlapping. Due to the large PSF it is feasible to do a colocalization in 2PM data. A further issue for the practical use exists. The colocalization of a spine and bouton in 2PM data does not guarantee that the spine is functional. We only labeled one dendrite and many other dendrites could also be nearby the bouton and being functional spines. Therefore, this application shows a possible application, but in real experiments this evaluation is not robust.

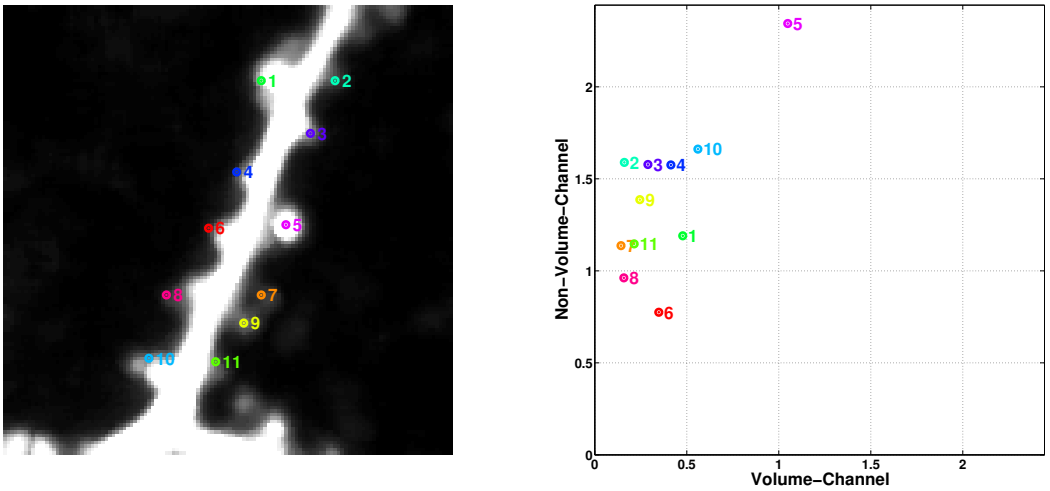


Figure 6.10.: Automatic spine detection, segmentation and identification of spines with a bouton. *Left:* MIP of the dendrite and location of automatically detected spines. *Right:* Plot of volume vs. non-volume channel intensities for all spines. Spine *5* is the functional spine. This spine has a high intensity in both channels.

7. Automatic Time Series Analysis

Time series analysis has a further level of difficulty for spine detection and segmentation. In addition to the detection in every single time point also the tracking over all time points is required. This induces new challenges like registration of input data, automatization of backbone initialization and tracking of spines over time. Therefore, adaptions of the time series data and the processing pipeline are required. In Section 7.1 we introduce the general situation and environment given for time series analysis. In Section 7.2 we introduce the adaption of the single time point analysis to the time series analysis. Finally, in Section 7.3 we present results of time series analysis using data from biological experiments.

7.1. Analysis Situation

Biological experiments with time series imaging require an automatic analysis of time series. The practical use of a spine segmentation software is depending on its ability to analyze time series. Many experiments are based on time series imaging and require to analyze the complete data automatically and not just every time point independently. Otherwise the advantage of automatic spine segmentation for single time points would not overcome the disadvantages of grouping together the results of multiple time points. Furthermore, the data amount in time series imaging can become quite large because often more than 10 time points are imaged. All these different issues show the importance of an automatic time series analysis of fluorescence images.

7.1.1. Time Series Experiments

A common setup for fluorescence imaging in biological experiments consists of tissue samples on glass plates which are living and are stored in a cell incubator. These samples can be imaged multiple times at typically regular intervals, e.g. every hour. This enables to study the impact of medicines and different stimulations of neurons and spines over time. Time series imaging generates a lot of image data. The manual analysis is very time consuming and becomes impractical for a larger data amount. By imaging over one working day (8 hour) at every hour 2 times (test and control group) 10 pieces of dendrite with each piece having about 10 spines already about 1600 spines must be manually segmented and measured. This demonstrates that already for small experiments an automatic analysis is crucial.

Studies of genetic diseases often require time series imaging. These diseases are frequent and accumulate high costs for the society. Today it is possible to identify people with such diseases early. However, mostly no cures are available. Therefore, it is important to study such diseases. Computer assisted image analysis enables to overcome the bottleneck in experiments conducted to study genetic diseases.

7.1.2. Imaging

The imaging of time series is done by fluorescence microscopy like Laser-Scanning Confocal Microscopy (LSCM) or 2-Photon Microscopy (2PM). Technically there is no difference between imaging a single time point or multiple time points. However, the skills the biologist needs to produce usable time series are diverse. It is crucial that the tissue samples survive multiple hours or even days under different situations (imaging, storing, moving around) and experimental influence (stimulation). In addition to the health state of the cell, the navigation with the microscopy in the tissue sample becomes important. The shape of the sample can change from time point to point. Nevertheless, the biologist must locate the same region and even the same piece of dendrite to image it multiple times. Navigation in the tissue probe and health state of the sample are crucial for time series imaging. Furthermore, issues like bleaching the cell must be considered. The biologist must find a balance between all these requirements such that over longer time images of living neurons in good quality are feasible.

7.1.3. Data Amount

Time series analysis produce within a short time a lot of data. From one sample typically 10 or more time points are imaged. The parameters that influence the amount of data are image sizes (width, height and number of slices), imaging time delta (how often to image), and duration of the experiment (number of images).

Typical image sizes are about $256 \times 256 \times 20$ voxels or $512 \times 512 \times 25$ voxels. The z-size can change much dependent on how much depth of the tissue should be imaged. The z-step size can also vary heavily (e.g. 0.5 µm or less up to 1.0 µm). However, it is also feasible to generate images of about $1024 \times 1024 \times 200$ which are about 800 MB of data (per time point).

It is feasible to image neurons a day or even longer such that spines are recognizable over the whole duration. If images are taken once an hour that gives 24 images. For shorter experiments with a smaller imaging time delta (e.g. every 10 minutes) similar number of images over time are generated. Even shorter imaging time deltas are feasible. However, the danger of harming the neuron increases and shorter time delta is often not required. The more time points are to be analyzed the more complex the tracking of spines over time becomes because the number of possible paths increases rapidly (exponential). Therefore, the time step delta should not be selected smaller as required by the experimental setting. Also the duration of imaging should not be extended too much because this makes the analysis more complex. Especially, if the shape of the dendrite or spines changes too much over time. Practical experience showed that after 12 hours shape changes are numerous and after 24 hours the fully automatic analysis becomes almost infeasible.

In Table 7.1 for a typical setting the data amount is shown. The used parameters are $256 \times 256 \times 20$ image size and about 15 time points. Furthermore per voxel two image channels (e.g. red and green fluorescent dye) are used and using unsigned int as data point.

The memory requirement during computation is much higher because additionally output and temporary images for the computation are required. However we were able to process even the largest dataset on a machine with 16 GB.

Size per	Calc. (bit)	Total in MB
Voxel	$2 * 32$	0.8×10^{-5}
Slice	$2 * 32 * 256 * 256$	0.5
Image	$2 * 32 * 256 * 256 * 20$	10.5
Time Series	$2 * 32 * 256 * 256 * 20 * 15$	157.3

Table 7.1.: Overview of typical data amount for input images of time series.

7.2. Pipeline of Time Series Analysis

The time series analysis is mainly based on the single time point analysis. However, the degree of automatization is higher. During the processing no interaction is possible. All time points are independently imaged. It is not guaranteed that the imaged regions are always the same. The number of transformation parameters is restricted because the tissue is on Petri dishes. We expect that a rotation along the vertical axis (optical axis) is possible. Additionally a horizontal shift (x-,y-axis) is possible. The rotation around x- and y-axis are infeasible because of the solid glass plate which is always oriented horizontal under the microscope. A shift in z-direction can be introduced by imaging. Further transformation are only possible because of tissue or shape changes of the dendrite. In summary, the pipeline for time series analysis must care about the automatic processing over all time points without intermediate interaction and the registration of the different time points. We introduce the steps to successfully detect, segment and analyze spines in time series.

7.2.1. Initialization and Optimization of Backbone in Time Series

The initialization of the backbone is an important issue. Thanks to the backbone optimization (see Section 4.4) we are able to deal with a rough initialized backbone. We demonstrated the use and success of this concept for single time points in Chapter 6. In time series an optimized backbone is required for every time point. However, the user cannot initialize the backbone for every time point. Therefore, an automatic transfer of the backbone from time point to time point is required. The backbone of the first time point can be manually initialized by the user by clicking seed points. This is identical to the single time point. For the later time points an automatic approach is required. We do a rigid alignment of all time points to the first one. Therefore, we can take the backbone from the previous time point as initialization. We sample seed points from the previous backbone at locations where we have high intensities in the aligned fluorescence image of the current time point. The backbone gets adapted to the current time point by using the backbone optimization. We demonstrate this effect in Figure 7.1.

Figure 7.1, *first row*, shows the alignment of a second time point to the first time point. *First row, left,* is the first time point with its backbone. In the *first row, middle,* a second time point with the backbone from the first time point is shown. It is visible that the dendrite is shifted and an alignment of the fluorescence image is required. *First row, right,* the aligned second time point with the backbone from the first time point is shown. The better overlap is now visible. In Figure 7.1, *second row*, the optimization of the backbone is presented. *Second row, left,* we see the first time point with its

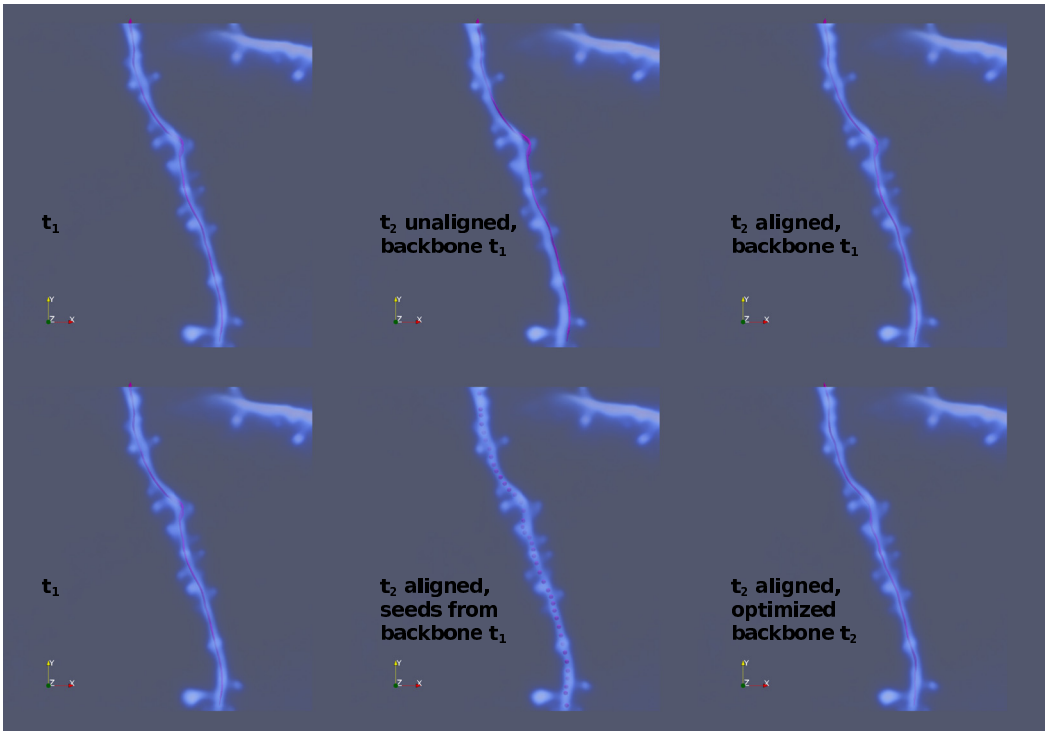


Figure 7.1.: *First row, left:* Fluorescence image of 1st time point with its backbone (*pink*). *First row, middle:* Backbone from 1st time point and a fluorescence image from the 2nd time point. No full overlap is given. *First row, right:* Aligned fluorescence image. The image is now overlapping what is visible by the backbone. *Second row, left:* Same as first row, left. *Second row, middle:* Backbone from 1st time point sampled for 2nd time point (*pink spheres*). *Second row, right:* Optimized backbone of 2nd time point and its fluorescence image.

backbone again. In the *second row, middle,* we see in *pink* the sampled points from the first backbone. At some locations it is still a little bit off. In the *second row, right,* the optimized version of the initialized backbone is shown. Finally, the backbone is located at the center of the dendrite and a successful spine prediction is possible.

7.2.2. Prediction on Time Series

The alignment and backbone optimization make it feasible to track spines by a search algorithm over time and spine candidates. No real tracking, based on the input (raw) data, is required. Therefore, an independent spine prediction for every time point is feasible. In our approach the prediction of spines is done identically for time series and single time points. Only the alignment of the fluorescence images to the first time point and the initialization of the backbone are different. In Figure 7.2 we show prediction results over multiple time points. Most spines have a very high response. Only some difficult ones have low responses. However, in the final time series analysis the low responses make it challenging to compute spine series over all time points. Therefore, we cannot discard low spine responses. In combination with multiple time points it showed that better results are generated if also the low spine prediction results are kept. However, to keep low intensities can increase the False Positive (FP) rate.

7.2.3. Segmentation of Prediction of Time Series

The segmentation of the prediction results is the last step of the detection and segmentation process. This step creates the binary spine candidates. This step is simple if the spine prediction is very strong and no responses are present at wrong locations. If the spine prediction signal within one time point is very heterogeneous, then it is important to apply a local thresholding approach (see Section 6.5). Otherwise it can happen that spines with a low feedback are discarded because of the strong spine feedback. The tracking of spines is quite robust even though irregular wrong spine candidates exist. Therefore, we segment also weak spine responses. The decision if such a spine candidate fits into one series of spines over time or not is up to the tracking over time. The segmentation of the spine prediction is done independently per time point. Details about the binarization can be found in Section 6.5. In Figure 7.3 we show segmentations which correspond to the prediction results visible in Figure 7.2. Each segmented object is a spine candidate.

7.2.4. Tracking of Spines in Time Series

The tracking of the spines in time series enables to make statements about spine changes over time in biological experiments. The detection, segmentation and measurement of spines in single time points is in many experiments not helpful. In the above described steps we introduced the required processing pipeline to have all the spine candidates of each time point. The final step is to find in this large group of spine candidates the same spine in each time point. This is achieved by finding all possible paths through the space of spine candidates and time. A path cost combined of distance and probability makes it possible to reduce the number of found paths. Each spine candidate is not used in more than one path (uniqueness of spine). A local search region reduces the number of paths heavily because generally from time point to time point spines move

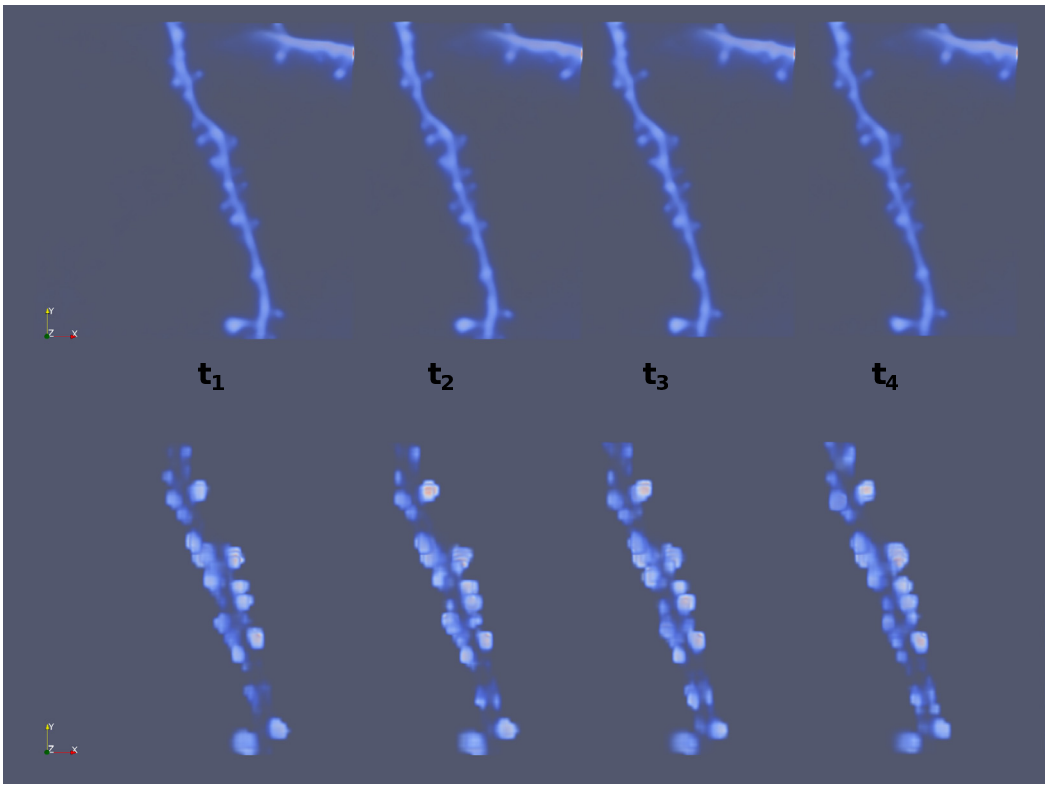


Figure 7.2.: *First row:* Raw data from t_1 to t_4. *Second row:* Spine prediction of t_1 to t_4. The prediction intensity can change from time point to time point. Also at wrong locations spine prediction feedback is possible. These wrong responses (FPs) are filtered out by the final spine tracking over time of the spine candidates.

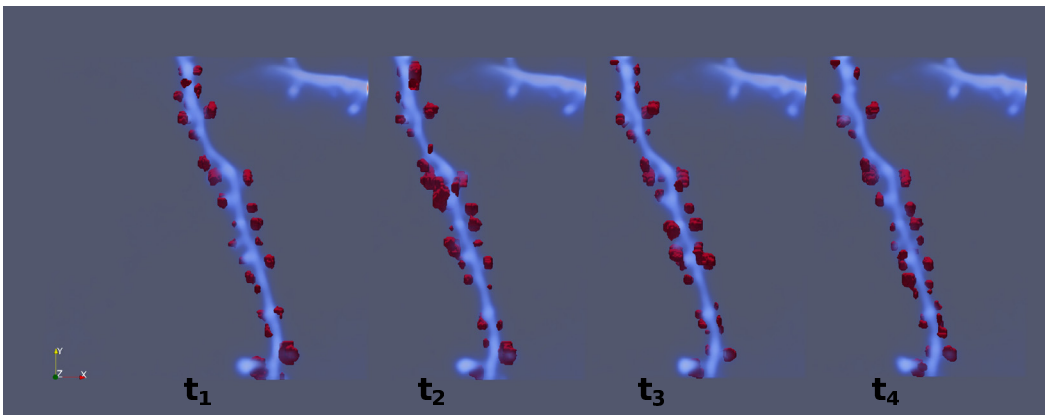


Figure 7.3.: Spine segmentation of t_1 to t_4. The segmentation shows spine candidates for strong and weak spine prediction results. However, the size is depending on the local thresholding condition.

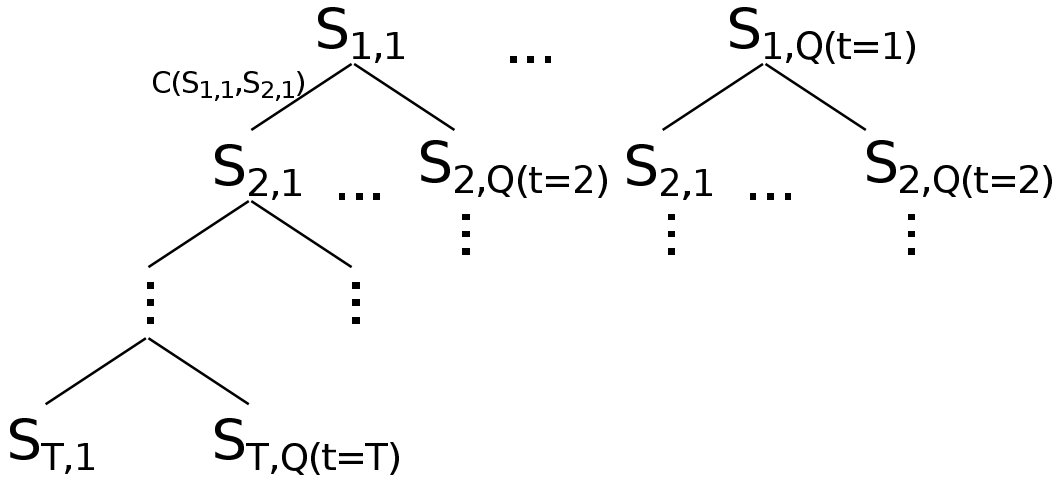


Figure 7.4.: Spine candidates path tree. For all spine candidates a tree with edges for all spine candidates from time point $t = 1, \ldots, T$ can be built. This tree consists of all possible paths and is fully expanded.

at maximum some µm. Equation (7.1) shows how dramatically fast the number of possible paths increases (exponential). Therefore, a local search is very useful reduce the computational cost. In Equation (7.1) $\overline{Q}$ is the average number of spine candidates per time point. The number of time points is T. The grow of paths and also the complexity is directly depending on the number of time points to be analyzed.

$$\#_{paths} = \prod_{t=0}^{T} Q(t) \approx \overline{Q}^{(T)} \tag{7.1}$$

In Figure 7.4 we show the tree of all possible spine paths. The number of paths is strongly reduced by cropping of all branches where the distance between the spines of time point t and $t+1$ is larger than a threshold (we use a maximal distance of 0.75 µm). Without the cropping a huge tree with the size approximated in Equation (7.1) has to be processed. This size grows exponential depending on the number of time points. To find the final paths the following steps are repeated:

- Move the path with lowest cost $C(S_{1,q_1}, \ldots, S_{T,q_T})$ to the result set, see Equation (7.2).
- Remove all paths from the tree which have at least one spine from the result set.
- Restart with the first step until the tree is empty.
- The resulting spine paths (spine traces over time) are in the result set.

The spine costs are defined by distances to the previous spine and to the first spine in time series. Additionally, also the spine prediction value is used (how probable a spine is). Equation (7.2) describes the cumulative cost of a spine path. The spines are described as S_{t,q_t} where $t = \{1, \ldots, T\}$ is the time point. A spine of any time point is named by S_{t,q_t}. The position of the spines are written as $p(S_{t,q_t})$. $L(S_{t,q_t})$ is the average spine likelihood of a spine (see also Table 7.2).

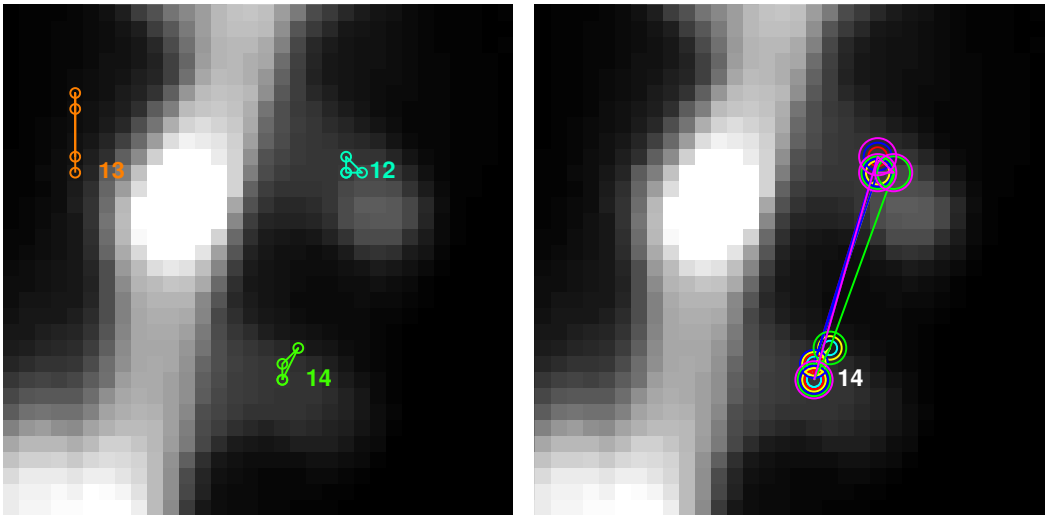


Figure 7.5.: *Left:* Tracking result over 4 time points and 3 visible spines. *Right:* Tracking possibilities for spine *14*. The finally picked path (see *left* in *green*) and all possible, rejected paths are displayed.

$$C(S_{1,q_1},\ldots,S_{T,q_T}) = \sum_{i=2}^{T} \frac{\|p(S_{i,q_i}) - p(S_{1,q_1})\|}{L(S_{i,q_i})L(S_{1,q_1})} + \frac{\|p(S_{i,q_i}) - p(S_{i-1,q_{i-1}})\|}{L(S_{i,q_i})L(S_{i-1,q_{i-1}})} \tag{7.2}$$

The spine path cost tries to represent the following properties that are motivated from the expectation we have to a meaningful spine path:

- Minimize movement from time point to time point.
- Minimize movement with respect to first time point. This is motivated as the dendrite pieces are rigid registered and in the optimal case there is no movement or only from the shape changes but not from imaging issues.
- Prefer spines with a high spine prediction likelihood over weak ones.

In Figure 7.5 we show a result of the tracking. The *left column* shows the result over the different time points for all detected spines. We will focus on spine *14*. In the *right column* we show for spine *14* all possible paths. There exist 6 possible paths. However, some possible paths have also spines which are a in possible paths of spine *12*. Therefore, the tree cost is taken into account and successfully finds the most probable path of spine *14*.

The concept of tracking the spines and visualization of the results is implemented in Matlab scripts. For further details about the implementation we refer the reader to Appendix B.3.

7.2.5. Statistical Analysis of Time Series Segmentation

We calculate the same statistical values for single time points and time series. Traces of spine values over time are possible. This means that the changes of a spine over time can be evaluated. All measured values can be exported as comma separated

Value	Description
Dendrite Intensity	Average intensity over all backbone voxels. This value is calculated for all available channels.
Mean Intensity	Average intensity over all voxels of a spine. This value is calculated for all available channels. In addition to the direct mean intensity value also a normalized version is computed. The normalization uses the average dendrite intensity which enables a comparison over different images.
Median Intensity	Median intensity over all voxels of a spine. This value is calculated in the same version like the mean intensity.
TopNPercentage Mean Intensity	Mean intensity calculated for the N percent highest voxel intensities of the spine. This value is calculated in the same version like the mean intensity.
Voxel Count	Number of voxels the spine has.
Spine Size	Size of the spine in μm^3.
Average Spine Probability	Mean value of all spine prediction values (voxels) of the spine.

Table 7.2.: List of all statistical values computed per spine and time point.

values file. The user can use these values directly within any environment (individual post-processing).

In the process of tracking spines a visualization of the most important spine values over time, e.g. top percentage intensity, is shown. These plots enable the user to see how the different spines change over time. Also he gets a first impression if all spines change similar or not. Additionally required statistical values must be calculated by the biologist himself.

Table 7.2 shows a list of all statistical values calculated per time point and spine. Additionally, the dendrite intensity is calculated.

7.3. Results

The quality of time series results depends on different factors of the experimental setup and the imaging system. Also the imaging experience of the biologist can give better or worse results in analysis. The most important parameters influencing the time series analysis are:

- Image resolution
- Number of imaged time points
- Laser power
- Fluorescent dye
- Health of neuron
- Geometrical shape change over time

Some of these parameters can be influenced. However, all the parameters driven by the neuron itself are most likely not influenceable (e.g. shape changes). Experienced biologists are able to image neurons with as less changes as possible over multiple time points. Therefore, it is feasible to successfully do time series analysis. We present here results of time series analysis of data imaged for biological experiments. These data are not specially imaged for our purpose but for research tasks of biologists. This means that we present real experimental data which we are able to analyze fully automatic (except the initialization of the first time point).

7.3.1. Single Channel

We present two results in which we discuss single channel, time series spine detection and segmentation. First we introduce a successful example of tracking spines over time. Then we show an example that demonstrates the additional challenges time series have compared to single time points segmentation.

Tracking

In Figure 7.6 we show the Maximum Intensity Projection (MIP) of the input data. 10 time points were processed, spines detected and segmented. The spine candidates were tracked and the final spine intensity traces calculated. The 21 found spines are overlaid in color. We can see that most of them do not have a large movement. In many cases, e.g. spine *2*, it is so small that the single time point detection positions are not recognizable. There exist also very weak spines, e.g. spine *16*, for which almost no intensity is visible. However, in the different time points the spine can be visually verified if the brightness and contrast is adjusted. This shows that manual spine detection can be very error-prone, time consuming and depending on the experience.

Biologists are interested in intensity changes over time. In Figure 7.7 we show the trace of spine intensity changes of the volume channel. In the same channel the detection and segmentation of spines was conducted. It shows that the spines have a broad range of Top5Percentage mean intensity values (normalized by the dendrite intensity). There exist very weak spines (spines *1*, *6* and *7*) which have no huge changes over time. However there exist also very strong spines (spine *2*) and spines which clearly increase the intensity over time (spine *8*). The automatic spine detection, segmentation and statistical spine value computation enables to get these facts in a simple, fast and robust manner.

Long Time Series Issues

It is feasible to image dendrites over multiple hours. We present here a result of imaging a dendrite every hour. After 10 hours we have a larger time delta because of the night. We successfully processed the huge dataset which has 20 time points in total. Moreover, each input image has a size of about $1024 \times 1024 \times 170$ voxels and two channels which are about 800 MB per image. In this example the issues of dendrite shape changes over time and the related challenges for the registration are recognizable. In Figure 7.8 we show the registration result for time point $t = 1$, $t = 6$,$t = 11$ and $t = 16$ (*left to right*). For $t = 6$ to $t = 16$ we also show the reference $t = 1$. In the *first row* the view is from top, xy-view. In the *second row* the view is from side, xz-view. In the view from top

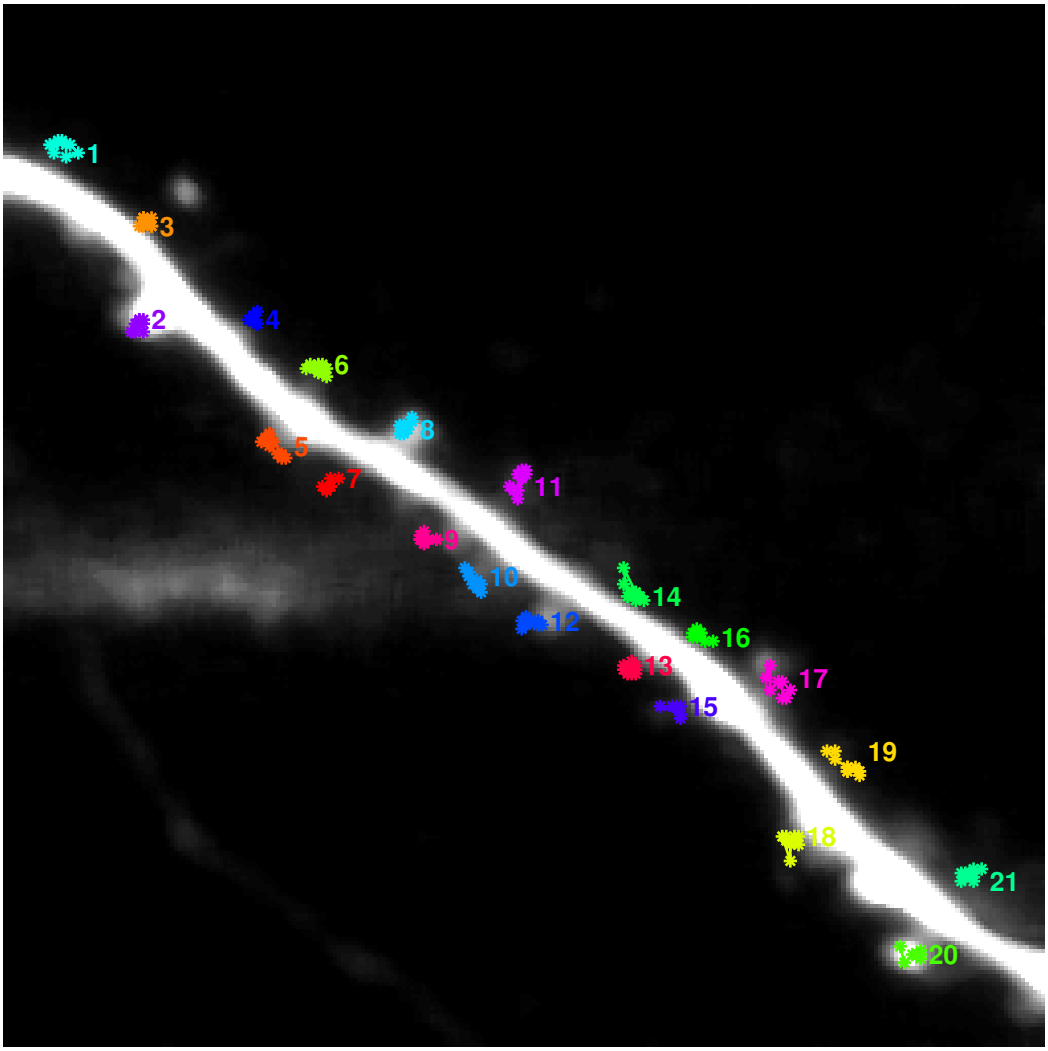


Figure 7.6.: The MIP of the input data is visualized. The intensities are clipped at about 30 % of the maximum for better visibility of the spines. The 21 found spines are tracked over 10 time points.

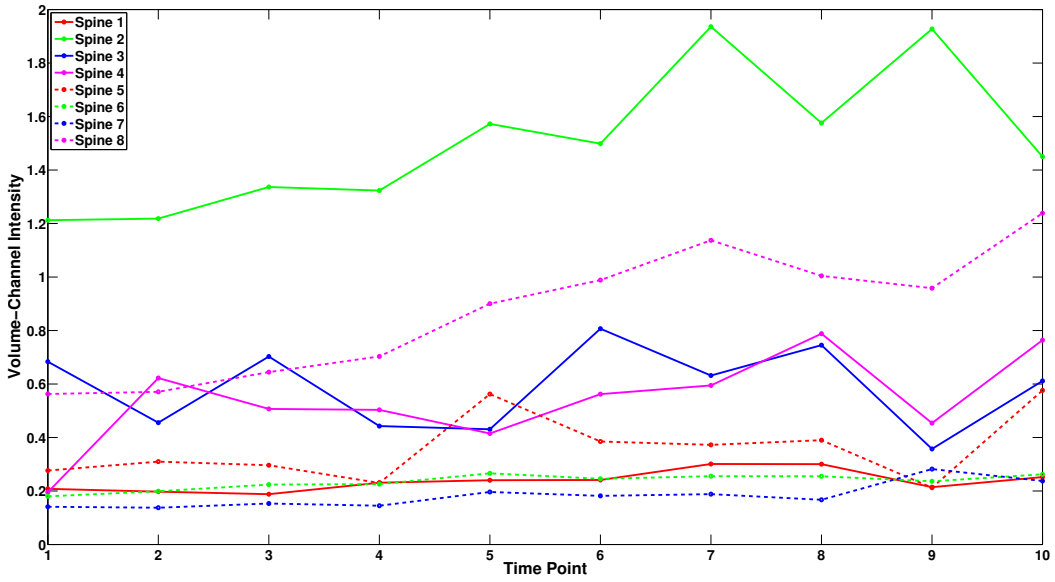


Figure 7.7.: For spine *1* to *8* from Figure 7.6 the Top5Percentage mean intensity from the volume channel is shown. Spine *2* has an intensity slightly higher than the average dendrite intensity and is hence > 1.

only slight misalignments are visible. However, in the view from the side the larger shape changes the lower dendrite has are well visible. This dendrite has a large change in z-depth with respect to the other branch. Therefore, a registration using translation and rotation cannot successfully register the images to the reference image ($t = 1$).

In Figure 7.9 we show the corresponding tracking results. From the *second* to *fourth column* we increased the number over which the tracking goes using $t = \{1, \ldots, 6\}$, $t = \{1, \ldots, 11\}$ and, $t = \{1, \ldots, 16\}$ respectively. These time points correspond to the time points showed in Figure 7.8. It is well recognizable that already after $t = 6$ almost no spines are successfully tracked at the branch which decreases in z-direction (left branch). The number of spines tracked decreases until time point $t = 11$ for this region. However, in the other regions the number of spines does not decrease very much, even until time point $t = 16$. In this example we do not discuss the detection quality because the registration fails over time. However, the focus is on the registration and tracking and the corresponding challenges. Furthermore, this huge datasets demonstrates the successful processing of large data amount.

This example shows how important a successful registration is. Furthermore, it demonstrates that in regions with good registration a tracking over multiple time points is feasible. Even though we do the prediction independent for each time point, the difficulties of processing large datasets with multiple time points increases exponential. This is because the following challenges are given. The data amount grows rapidly. The likelihood that a registration fails increases exponential with each additional time point. The tree to be solved in the tracking task also increases exponential (see Equation (7.1)). Furthermore, the detection rate can decrease exponentially. However, most likely the same spines are not detected in different time points. Otherwise by an average detection rate of 0.8 after three time points the detection rate would decrease to $0.8^3 = 0.51$ and after ten time points it would be only $0.8^{10} = 0.11$. Even with the incorrect registration,

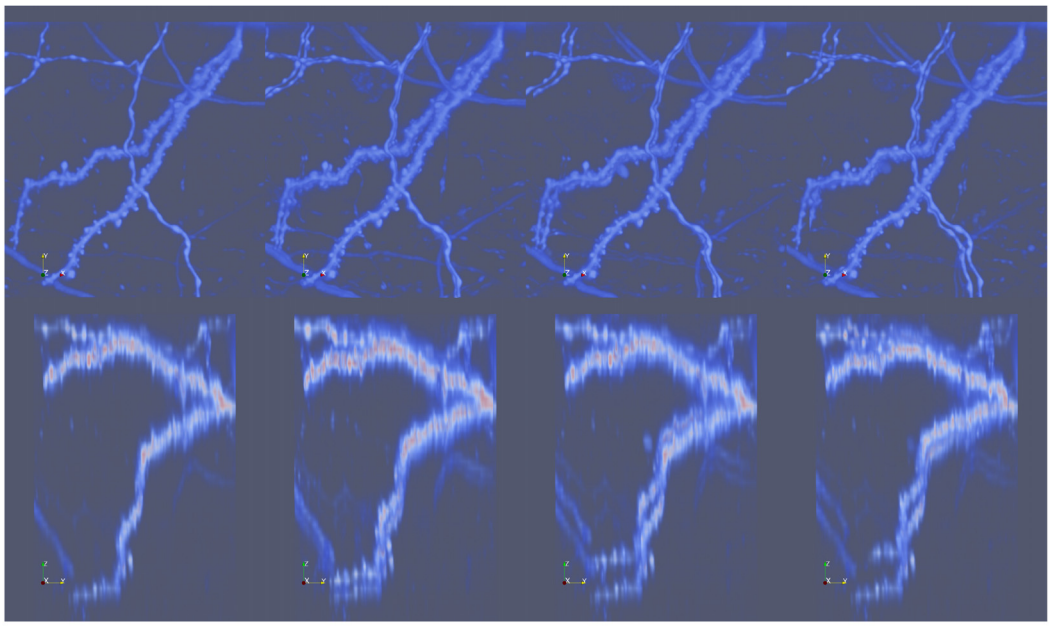


Figure 7.8.: *First row, first column:* Volume rendering of volume channel for $t = 1$. View from top, xy-view. *First row, second to fourth column:* Volume rendering of volume channel for $t = 1$ (registration reference) and additionally $t = 6$, $t = 11$ and $t = 16$, respectively. View from top, xy-view. The lower dendrite has shape changes which are mainly the change in z-depth. Therefore, a rigid registration is not fully successful. *Second row:* Same as *first row* but viewed from side, xz-view.

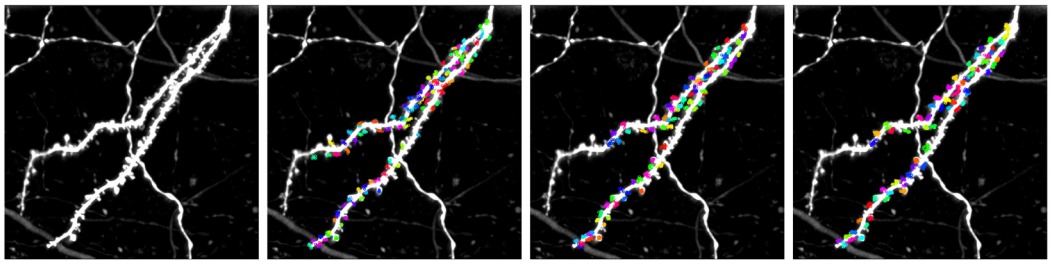


Figure 7.9.: *First column:* Input data $t = 1$ without tracking. *Second to fourth column:* Input data $t = 1$ and overlaid tracking results for $t = \{1, \ldots, 6\}$, $t = \{1, \ldots, 11\}$ and $t = \{1, \ldots, 16\}$ respectively.

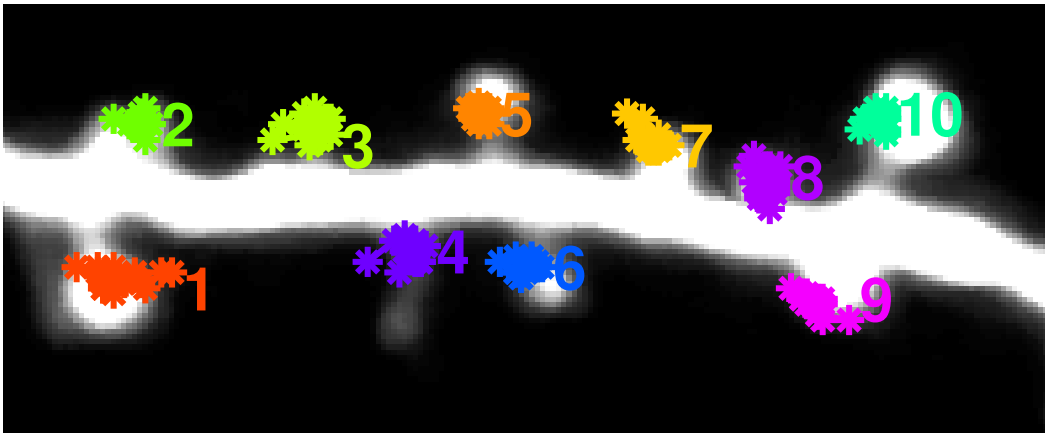


Figure 7.10.: MIP of volume channel of first time point of ER detection experiment. Overlaid are the spine locations over all time points.

which failed because of the dendrite shape changes, we have clearly a higher rate. This shows that our approach has for similar input data (different time points) a similar output. This demonstrates the robustness of the approach.

7.3.2. Multiple Channels

We present an application of tracking spines in multiple channels over time. Basically, the goal is the same like for the Endoplasmic Reticulum (ER) classification experiment in Section 6.6.2. However, here we have a time series. The same piece of dendrite is imaged every 10 minutes. In total 18 time points were imaged. In this task the goal is to classify for each spine over time if it has an ER or not. Therefore, this task requires to detect, segment and track the spines during all time points. Furthermore, the identified values in the second channel (ER marker) must be distinct. Tracking the ER in spines will help unveil whether manipulating the ER presence and activity would alter synaptic plasticity and potentially, the stability of certain memories. For a deeper understanding about ER we refer the interested reader to Table 2.1 in Section 2.1 and to the corresponding literature.

In Figure 7.10 we show the MIP of the first time point with overlaid spine positions over time. The intensity is clipped at 30 % to visualize the spines better.

We show for two selected spines (*5* and *6*) a trace of both channel intensities over all 18 time points in Figure 7.11. The *red* curve is the ER channel and the *blue* one is the volume channel. Spine *5*, *dashed line*, has an ER at time points *4*, *8* and *16*. In contrast, spine *6*, *solid line*, never has a signal in the ER channel. The same results were retrieved by an expert manually analyzing the data set.

With this example the whole complexity of spine detection, segmentation and tracking in multi-channel fluorescence images is demonstrated. We are able to identify in time series which spines have an ER at each time point independently. Furthermore, we can follow the changes of each spine over time. The results of True Positives (TPs) correspond in most cases to the results that we have from manual expert labeling. However, depending on the detection quality spines can be missed or the classification can have wrong results. In the case of weak ER signals this is also an issue of the binariza-

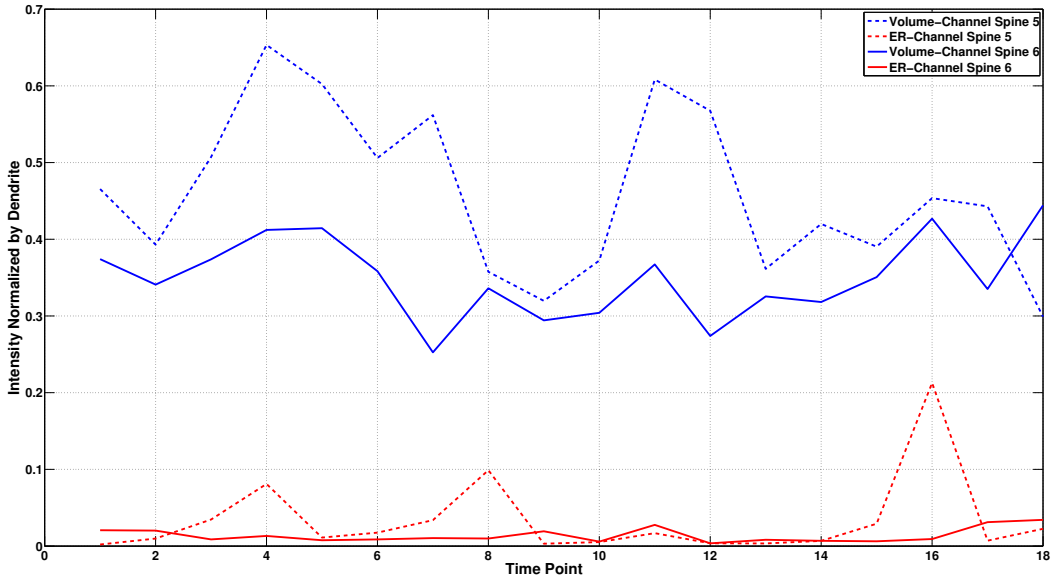


Figure 7.11.: Intensity traces for spines *5* and *6* of Figure 7.10. Spine *5* has an ER (*red* curve) at time points *4*, *8* and *16* while spine *6* never has an ER.

tion of the spines. Though, the basic task is solved and the intended application was demonstrated. In practical use the method and its advantages and disadvantages have to be studied in future. This will require the regular use in a biological laboratory and further comparison to expert labeling which will be time consuming. Furthermore, it has to be evaluated if the approach is enough robust compared to the manual analysis.

8. Orientation of Spines in Space

Dendritic spines are protrusions along the dendrite. In literature there does not exist a discussion about the preferred orientation of spines in space. They are expected to be oriented in all directions or in other word homogeneous distributed in 360° around the dendrite. Older automatic analysis of fluorescence images is mostly done using Maximum Intensity Projection (MIP) data. In approaches like this only horizontal oriented spines (with respect to the laser direction) can be detected. Also experts label almost exclusively horizontal oriented spines. In difference to the automatic MIP approaches experts are able to label huge vertical oriented spines. However, even for experts it is very difficult to see and label vertical oriented spines correctly because of the elongated Point Spread Function (PSF). Most full 3-Dimensional (3D) spine detection and segmentation approaches try to overcome this issue and detect spines which are oriented in any direction. With the detection of spines in all orientations it must be discussed if really vertical oriented spines are detected. Moreover, we query how important the detection of all spines is and if the homogeneous distribution is a fact. Thanks to the Serial Block-Face Scanning Electron Microscopy (SBFS-EM) data it is feasible to compute the spine orientation within a tissue sample. In Section 8.1 we discuss the prerequisites of spine distribution. The computation of the orientation of spines in space is introduced in Section 8.2. The results and the impact on spine detection and segmentation are discussed in Section 8.3.

8.1. Prerequisites

In fluorescence microscopy the imaged neurons are living and the sample has some thickness (tissue). This simulates a similar situation like *in-vivo*. However, the tissue is only some micrometers thick and placed in a Petri dish. Furthermore, the tissue is cropped what kills some of the cells. The distribution of neurons is more horizontal than vertical because of the preparation in the Petri dish. All these prerequisites motivate us to expect that the spines have a preferred horizontal orientation (with respect to the Petri dish). Figure 8.1 shows by a schematic diagram how the cultures and imaging situation look like. The tissue is horizontally oriented in the Petri dish. The laser and the optical axis are vertical to the Petri dish from top to bottom. We will verify if spines have a homogeneous distribution in space or if a preferred orientation exists.

8.2. Used Serial Block-Face Electron Microscopy Data and Analysis

The motivation that spines do not grow in a uniform distribution is given by fluorescence images and their analysis. However, it is not possible to study the orientation of spines in space in fluorescence images. Therefore, we use SBFS-EM data in which all the spines can be labeled and analyzed in all directions.

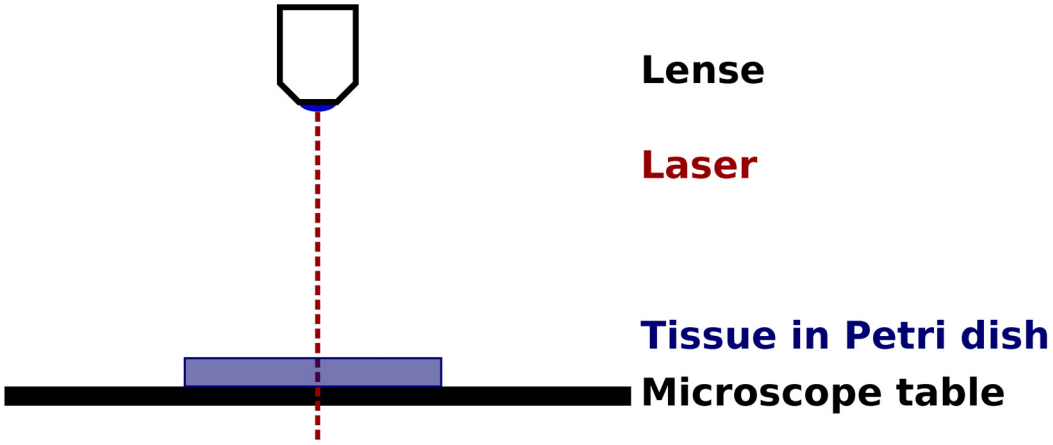


Figure 8.1.: Schematic diagram of slice culture and imaging axis. At bottom the Petri dish with the tissue in it. From top the imaging laser. Tissue in the Petri dish has a larger horizontal than vertical width.

Data

Manual reconstructions from SBFS-EM data for a labeled neuron are feasible. We described how the dendrite shape can be reconstructed in Section 3.3. In SBFS-EM data the correct shape and orientation of spines is visible. The tissue is fixed but not changed or influenced different as for fluorescence microscopy.

Analysis

Two simple approaches to compute the spine orientation are feasible. The simpler approach uses the image moments of the segmented spines. The first principal axis describes the orientation of the spine in space. This approach has the disadvantage that for almost spherical spines the direction can become arbitrary and does not correspond to the vector from the spine neck to the tip. Furthermore, also for spines with a very short neck it can happen that the direction of the principal axis is parallel to the backbone. This approach is not as robust as the second one. The second approach uses as direction of spine the vector from the nearest backbone point to the center of mass of the spine. The center of mass is robust and always computable. Furthermore, the vector from the nearest backbone point to the center of mass is a good approximation of the orientation of the spine. Therefore, only small errors are expected with this approach. The center of mass for a segmented spine in a binary image is computed by:

$$c_x(S_i) = \frac{\sum_{x,y,z \in S_i} (x I_{x,y,z})}{\sum_{x,y,z \in S_i} I_{x,y,z}} = \frac{\sum_{x,y,z \in S_i} (x)}{\sum_{x,y,z \in S_i} I_{x,y,z}} \tag{8.1}$$

where I is the binary image and x, y, z are pixel positions in the image. The spine is named S_i. The 3D point of the center of mass of the spine is given by $c(S_i) = \begin{bmatrix} c_x(S_i) & c_y(S_i) & c_z(S_i) \end{bmatrix}^T$. The orientation vector $\vec{o}(S_i)$ is then given by:

$$\vec{o}(S_i) = c(S_i) - \hat{b}(S_i) \tag{8.2}$$

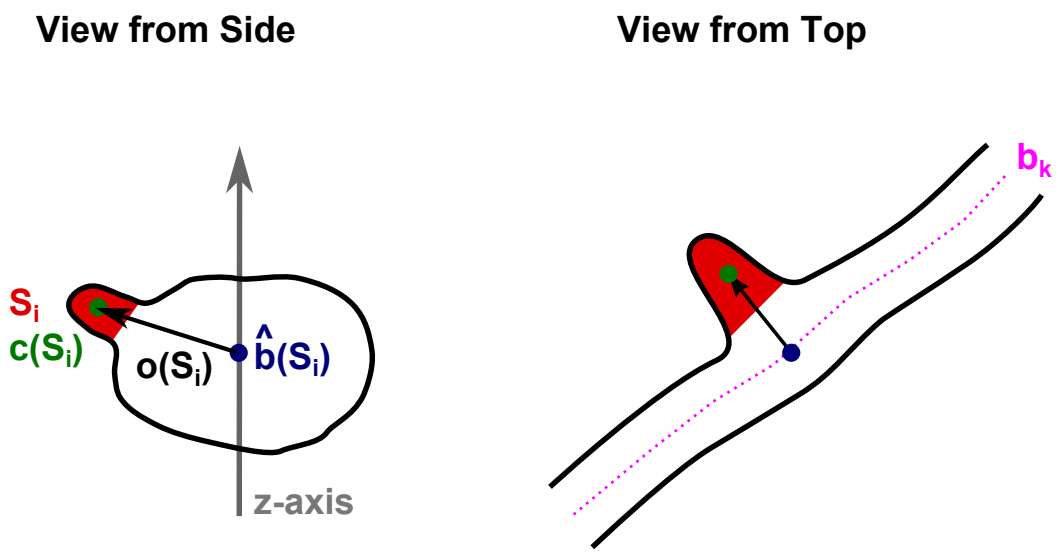


Figure 8.2.: *Left:* Schematic view from side. In *red* spine S_i is marked. Its nearest backbone point $\hat{b}(S_i)$ (*blue*) and center of mass $c(S_i)$ (*green*) are the end points of the orientation vector $\vec{o}(S_i)$. In *gray* the z-axis is visualized. The orientation is then given between the vector $\vec{o}(S_i)$ and the z-axis. *Right:* Same situation viewed from top. Additionally in *pink* all the backbone points b_k are visualized.

where $\hat{b}(S_i)$ describes the nearest backbone point. The nearest backbone point is given by:

$$\hat{b}(S_i) = \underset{b_k}{\operatorname{argmin}} \|b_k - c(S_i)\| \tag{8.3}$$

where b_k describes all the points of the backbone. The vector $\vec{v}$ enables us to compute the orientation with respect to the z-axis. Then we can compute a histogram of the orientation distribution. In Figure 8.2 we show a schematic drawing which simplifies the understanding of the situation.

8.3. Results and Impact on Spine Detection

We will present here the distribution of spines which we computed with the above introduced methods. Moreover, we will discuss the impact on manual and automatic fluorescence image analysis.

8.3.1. Results

We were able to compute the orientation of spines for a larger dataset. We computed in total the spine orientation of over 200 spines. These spines are from two cells. The orientation of the spines is computed by the method using the center of mass described in Section 8.2. No further manual correction or selection of the results was done. The spine orientation analysis was proceeded fully automatic. Therefore, partial errors of

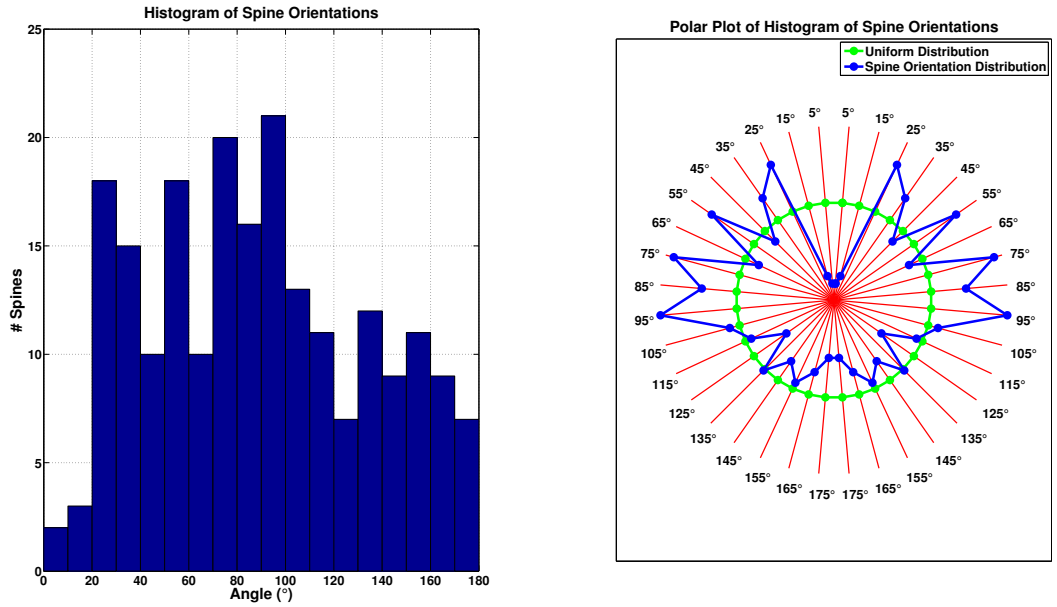


Figure 8.3.: Distribution of dendritic spines in space. A clear preference to the horizontal direction (Petri dish) is visible. *Left:* Histogram of spine orientations. 0° are spines oriented to the top, 90° are horizontal and 180° to the bottom oriented spines. *Right:* Polar plot with horizontal mirrored data.

spine orientation are possible. However, the high number of spine orientations ensures a significant meaning of the data.

Figure 8.3 shows the result of the analysis. In Figure 8.3, *left*, we show the histogram of spine orientations. The spines are divided into 10° bins. 0° corresponds to upwards, 90° to horizontal and 180° to downwards oriented spines. The data was projected on one side (horizontal) and then the distribution computed. In Figure 8.3, *right*, the results were mirrored for visual purpose in the polar plot. The distance from points to the center describes the number of spines relative to the maximum. In *blue* the histogram distribution is visualized. In *green* the corresponding homogeneous distribution is displayed. The polar plot demonstrates well the higher probability of horizontal oriented spines.

The expected situation of more horizontal than vertical oriented spines is confirmed. However, there is also a slight preference to the bottom compared to the top visible. Different reasons can be listed for these effects. We can imagine that gravity lets spines fall to the side. However, this should then also be the case *in-vivo*. Another issue could be that just more connectable neurons are available on side and bottom than top. This because imaging takes place more at the surface of the sample. However, this cannot be verified by the SBFS-EM data and reconstruction from the used tissue samples. Furthermore, the lower probability of downwards oriented spines would not be explained. The analysis in *in-vivo* data and larger volumes is beyond the interest of the authors and would also require different SBFS-EM data and its reconstructions.

The mean over all measured values is 88.9° and the median is 86.1°. This means that the gravity cannot be the reason for the changed spine orientation. Otherwise we would expect that the mean and median are significant larger than 90°. The mean of

spines per 10° bin is 11.7 spines. However, like expected the standard deviation is quite high with ±5.2 spines per bin.

In Appendix A, Section A.5, a list with the detailed spine distribution can be found.

8.3.2. Impact on Spine Detection

We clearly could demonstrate that spines in slice cultures are more oriented in horizontal than vertical direction. This distribution has some impact on spine analysis. It becomes more important to detect spines in horizontal orientation because they seem to be more frequent. Fortunately, it is also simpler to detect horizontal oriented spines because the PSF of fluorescence microscopy is less elongated in this direction. This result demonstrates that the analysis of MIP data is more accurate as expected. Less spines are lost or cannot be detected (lower False Negative (FN) rate). If an approach is able to detect 100 % of spines within 45° to 135° that still results in about 60 % of all existing spines. Though, the angular range is only 50 %. Furthermore, the detection of spines in 3D between 30° and 150° corresponds to about 76 % (162 of 212 spines) of all spines. Therefore, the more difficult and error-prone detection of vertical oriented spines is less important. It has not so a large impact to only detect horizontal spines as expected before this evaluation. Of course, nevertheless a huge part of spines (about 25 %) is still not detected and analyzed if the vertical spines are neglected.

With focus on a low False Positive (FP) rate and the fact that spines are not uniformly distributed in orientation of space the detection of horizontal spines preferred over vertical spines makes sense. The higher probability of horizontal spines helps to have a high True Positive (TP) rate and low FP rate because the detection and segmentation of horizontal spines is simpler. Furthermore, for vertical spines the computation of intensities is more error-prone because the PSF has a larger influence in the vertical direction (optical axis).

It is more feasible to track horizontal oriented spines of 4-Dimensional (3-Dimensional plus time) data over all time points than tracking the vertical oriented spines. This is based on the fact that the spine concentration of horizontal oriented spines is higher and also the spine detection and segmentation rates of most algorithms is higher for horizontal oriented spines. This makes it more feasible to track horizontal oriented spines over longer time series, e.g. more than 10 time points.

9. Discussion

In this thesis we introduced a new concept for spine detection and segmentation. Moreover, we were able to present an application in multi-channel images of single time points and time series. We introduced novelties which were integrated into a full pipeline to successfully support biologists in their process of analyzing 2-Photon Microscopy (2PM) data. A discussion about these features can be found in Section 9.1. We were able to demonstrate the success of our method in theoretical and practical applications. However, we present ideas how to extend and improve the method and its application as well as corresponding future work to be carried out in Section 9.2.

9.1. Conclusion

We were able to present a correlative dataset (Correlative Light Electron Microscopy (CLEM)) which is one of the key points of our approach. The dataset shows dendrites including spines in high resolution and enclosed mitochondria. This dataset is a novelty in terms of ground truth data. We were able to validate our method on real data while having additionally high resolution ground truth data. We were independent from expert knowledge. We are not aware of any other method using CLEM to validate an approach. Furthermore, the Serial Block-Face Scanning Electron Microscopy (SBFS-EM) data was reconstructed in high detail and combined with a synthetic Point Spread Function (PSF) we computed Digitally Reconstructed Fluorescence Images (DRFIs). Due to DRFIs the amount of training data was not an issue anymore. At the same time, we overcame the problem of expert labeled training data. It is challenging to get expert labeled data in good quality and large number. In contrast, the generation of real fluorescence data is no challenge. However, the labeling of the data is very subjective to the expert and an error-prone and time consuming task. Therefore, expert labeled data is limiting the idea of training a statistical model. We demonstrated that the manual reconstruction of a dye marked neuron (or its dendrites) is straightforward. Also non-experts can do reconstructions. Furthermore, the manual detection of spines in Electron Microscopy (EM) reconstructions is simple. Combined with the new concept of DRFIs we overcame the issue of being dependent on an expert in the processing pipeline. Moreover, we were able to generate more data as we rotated the reconstructions before computing the DRFIs. This enabled us to compute new, representative training data. Because of CLEM data, we were able to validate the concept in general but in particular to verify that we overcame the differences between synthetic and real data. Correlative data was a very helpful tool for testing and validation because we did not only count spines (True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN)). We also studied why spines were not detected on the level of the 2-Dimensional (2D) slices. Furthermore, the intensity distribution from DRFIs and real 2PM data was comparable.

In the second part we introduced a statistical dendrite intensity and spine probability

model. This combined model was trained by data from SBFS-EM data using DRFIs. We demonstrated that it is feasible to predict the most probable location of spine from dendrite intensity 2D slices and segment them. Furthermore, a more stable prediction result was achieved by optimizing the backbone using the probability of a slice being represented by the model. We are not aware of any previous approach optimizing the dendrite backbone by using a statistical model. Optimizing the backbone with respect to the statistical model improves the prediction quality and enables to use an inaccurate backbone initialization from the user. Therefore, we were able to work with clicked seed points as backbone initialization and did not require high precision of the seed points.

These novelties combined together with additional image analysis concepts enabled us to develop a spine detection and segmentation software (see Appendix B). We demonstrated the success of the software for single time point fluorescence images with multiple channels to detect and segment spines (see Section 6.6.2). Furthermore, we showed that it is feasible to automatically detect properties of spines in single time points and to divide the spines into two classes. We demonstrated the classification into spines with and without Endoplasmic Reticulum (ER). In addition to single time points we also introduced the analysis of time series (see Chapter 7). We demonstrated how multiple time points can be registered and how spines can be detected and segmented by only initializing the first time point. Furthermore, we introduced concepts for the spine tracking in an additional software module. Due to the independence of the spine tracking and visualization from the detection software we enable biologists to change the tracking and visualization software to their individual requirements (see Section B.3). In Section 7.3 we demonstrated that it is feasible to track spines and their changes over multiple time points using real fluorescence images. We demonstrated this by using data which was not made for testing but which already existed from biological experiments. Additionally, we demonstrated that the detection of horizontal spines is more important because the distribution of spines in space is not uniform (see Chapter 8).

In summary, we introduced a new spine detection and segmentation software for multi-channel, time series fluorescence images in 3-Dimensional (3D) data. We used the new concept of DRFIs data from SBFS-EM reconstructions and showed a validation using CLEM. Finally, we presented the practical application in single and multiple time point, multi-channel images. All together we introduced a complete pipeline for spine detection, segmentation and analysis which shows its performance in practical application.

9.2. Outlook

We presented a complete processing pipeline and also demonstrated its practical use. Nevertheless, further improvements and extensions are possible. Further work would be feasible in the area of training the statistical model, detection as well as segmentation and post processing. For all these parts we will give some ideas what could be done in future.

Training Data

We presented how a huge amount of training data can be generated by the use of rotation. Because of the PSF this data is unique and provides a high variability.

However, the number of dendrite pieces and spines is quite limited (four dendrites from one cell). An extension to further data would probably improve the quality of the statistical model.

As the second area of possible improvements the PSF has to be mentioned. We use an approximation by a 3D Gaussian distribution. This works and generalizes well. If, however, always the same imaging parameters (microscope setting, dye and other parameters) were used, the computation of a more specific PSF could improve the results. A more accurate PSF would probably improve the analysis of weak signals. The issue of a location dependent PSF cannot be solved in an easy way.

Analysis

The spine detection could be extended using additional features. We combine the results of the statistical model with backbone-orthogonal features. Additionally, it would be feasible to learn 3D features. Due to the DRFIs, the division into the two classes would be provided. For example random forests can be learned and they introduce a further classification criterion of every voxel into spine or non-spine.

Another issue is the segmentation of the prediction results and the splitting of spines. In many situations spines are densely distributed and overlap in the fluorescence image. Touching spines require splitting. In the prediction result this separation is quite difficult because regardless of the fluorescence intensity of the spine, a high value in the prediction map is possible. Therefore, a splitting criteria based on the original fluorescence intensities would be more promising. However, besides finding the plane that splits the spines, the candidates for splitting must be found first. This is a challenging task. Depending on the application, a semi-automatic approach where the user picks the spines to be split and where an algorithm finds the separation plane would be helpful. For single time points we already integrated such functionality for test reasons in our application.

In contrast to the splitting of spines, in some situations merging of spines is required. This is mainly the case if the detection quality was bad. For test reasons we already integrated a merging method for single time points which computes a threshold for the volume between two spines. This merges the two spines but tries not to use background values as spines. Also here a detection of such spines and automatic merging would be required and helpful for time series analysis.

Tracking and Post Processing

In post processing we introduced a simple but quite stable spine tracking concept. The implemented tracking is not able to track spines which appear, disappear or in some time points are not detected (FNs). With respect to this problem an extension of raw tracking to a more complex system combining tracking and reestimating would be helpful if in intermediate time points the spines disappeared or a FN exists. If this decision is made, it would probably be feasible to compute the segmentation from neighboring time points. Like this, even the tracking process could be improved because the number of FNs decreases and, therefore, the tracking paths become more distinct.

A. Additional Material

A.1. Prior Work

In the introduction (Section 1.2) we introduced the most important prior work and their ideas. Here we present an overview of these approaches in Table A.1. We give a short summary about the dendrite and spine segmentation and the used methods. Furthermore, a list of properties is given which are the use of skeleton, different handling of attached and detached spines, classification of spines into the classical types (mushroom, stubby, thin), use of deconvolution, data type (Laser-Scanning Confocal Microscopy (LSCM) or 2-Photon Microscopy (2PM)) and the publication year. The table is sorted chronologically and then by the last author.

It is to mention that many of the publications are from the group of Stephen T.C. Wong or collaboration with his group. Therefore, many of these publications build up on each other. Furthermore, conceptually they are often similar and have also similar quality of test data. However, the evolution of spine detection and segmentation is recognizable in the publications [10, 11, 24, 32, 47, 60, 61] of this group.

Last Author/Work	Dendrite Segmentation	Spine Segmentation	3D	Skeleton/Backbone	Attached/Detached Spine Processing	Spine Classification	Deconvolution	LSCM	2PM	Year
Svoboda [34]	Thresholding to binarize dendrite.	Use of branches from skeleton to detect spines.	X[8]	X	X	X	X	-	X	2002
Obermayer [45]	Fitting skeleton and radius simultaneously.	-	X	X	-	-	X[9]	X	-	2004
Wong [10]	Adaptive Thresholding.	Local Signal Noise Ratio (SNR) and local morphology segmentation.	X[8]	X	X	-	-	-	X	2007
Wong [61]	Estimated by centerline extended to boundary.	Spine detection by centerline branches and boundary lines.	-	X	X	-	-	-	X	2007
Zhou [62]	Level set segmentation and iso-surfacing.	Using tips for attached and radius for detached spines.	X	X	X	-	-	-	X	2008
Wearne [43]	Adaptive thresholding and radius estimation.	Cluster building from tips.	X	-	-	X	X	X	X[10]	2008
Asal [3]	No dendrite segmentation is done.	Segmentation using local attributes and features. Classification of the features with Support Vector Machine.	X	-	-	-	-	X	-	2008

[8] Partially 2D
[9] Not entirely clear
[10] No examples

Last Author/Work	Dendrite Segmentation	Spine Segmentation	3D	Skeleton/Backbone	Attached/Detached Spine Processing	Spine Classification	Deconvolution	LSCM	2PM	Year
Roysam [58]	Intensity weighted Minimal Spanning Tree on gray scale image.	Spine Model for Skeleton.	X	X	-	-	X[11]	X	-	2009
Wearne [44]	Binary image with seed point and region growing.	-	X	X[12]	-	-	X	X	-	2009
Wong [32]	Surface representation followed by medial axis finding using medial geodesic function.	Growing detached spines to dendrite and find by skeleton branches the spines.	X	X	X	-	X	-	X	2009
Wong [47]	Surface reconstruction from skeleton.	Feature computation and watershed segmentation.	X	X	-	X	-	X	-	2009
Wong [60]	Gradient vector tracking and central region.	Feature points and shape model of spines.	X	X	-	-	X	X	-	2010
Wong [11]	Local Region Of Interest (ROI) for background removal and oriented Markov random field.	Estimate local dendrite diameter and detect foreground pixels with larger distance.	-	X	X[13]	-	-	-	X	2010

[11] Partially
[12] Goal of Approach
[13] For measurement

Last Author/Work	Dendrite Segmentation	Spine Segmentation	3D	Skeleton/Backbone	Attached/Detached Spine Processing	Spine Classification	Deconvolution	LSCM	2PM	Year
Kim [50]	Thresholding.	Spine detection from skeleton branches, tip detection followed by geodesic active contour segmentation.	X	X	-	X	-[14]	X	-	2011
Deng [37]	2D Rayburst algorithm to find backbone and radius estimation.	Use of 3 features: distance to backbone, mean curvature of surface and normal variance. Followed by region growing segmentation.	X	X	-	X	-	X	-	2012
Wong [24]	Unclear if segmentation of dendrite is done and if any backbone is used for spine segmentation.	Detection of tip by cross-sectional curvature followed by region growing segmentation.	X	-	-	-	X[15]	X	-	2012

Table A.1.: Overview of prior work. We give a short description and list the methods of the segmentation. Moreover, we categorize the work properties.

[14] Unsharp Mask Filtering
[15] Regularized Wiener Filter

A.2. Overview of Concept

In Figure A.1 we present an overview of the whole approach. This figure was earlier presented at the AllSystemsX.ch-Day 2013 in a poster. However, this figure demonstrates in a nice overview the main parts of the approach and thesis. *A to F* is the learning part. From Serial Block-Face Scanning Electron Microscopy (SBFS-EM) raw data (A) the reconstructions (B) and then the Digitally Reconstructed Fluorescence Image (DRFI) (C) are computed. Then 2-Dimensional (2D) backbone orthogonal slices from the dendrite and spine DRFI are extracted (D). Additionally, the spine probability maps are computed (E). From the dendrite intensity slices and spine probability maps two Principal Component Analysis (PCA) models are computed (F). The second part (*F to K*) is the prediction part. For a test data set (G) 2D slices (H) are extracted. For these slices by using the models (F) a spine prediction map I is computed. Together with additional backbone parallel features (which are not showed) the 3-Dimensional (3D) spine prediction (J) is computed. This map is then binarized and referred as spine candidates (K). As third part (*K to M*) the whole process can be done for time series. Independently the prediction part is done and then multiple, 3D spine prediction maps exist. These are then used for tracking and finding the real spines (L). Finally, it is feasible to trace the intensity changes of a each spine over time as visualized for some spines (M).

A.3. Backbone Optimization

We show here some additional results for the process of backbone optimization and optimization in time series.

In Figure A.2 we show how the backbone looks for a dataset of 31 time points. We show the backbone and its corresponding registered input image at every fourth time point plus the last one. It is recognizable that the backbone optimization and automatic seed point initialization works well over all time points. However, at some time points it can happen that the backbone is optimized towards spines. This effect is already known from single time points. It is recognizable that we hold the start and end point of the backbone fix over all time points. Therefore, at some time points the backbone is longer than the available piece of dendrite. The processing of 31 time points demonstrates the robustness of the approach and the implementation.

In single time points it is interesting to see the iterative optimization approach. In Figure 4.6 of Section 4.6 we showed the optimization over two time points. Here we demonstrate its stability over more than two time points. In Figure A.3 we show an optimization that is iterated six times. In *pink* the initialization for each iteration is shown. In *green* the optimized backbone per iteration is displayed. Furthermore, the *gray* vectors show the maximum spine probability of the shifted slices. The main optimization takes already place in the first iteration. Afterwards, mainly slight changes are recognizable which are partially also caused by the recomputed normals after each iteration. However, we can see that the backbone stays robust. That it is feasible to correct misaligned regions of the backbone in very few iterations is already showed in Section 4.6.

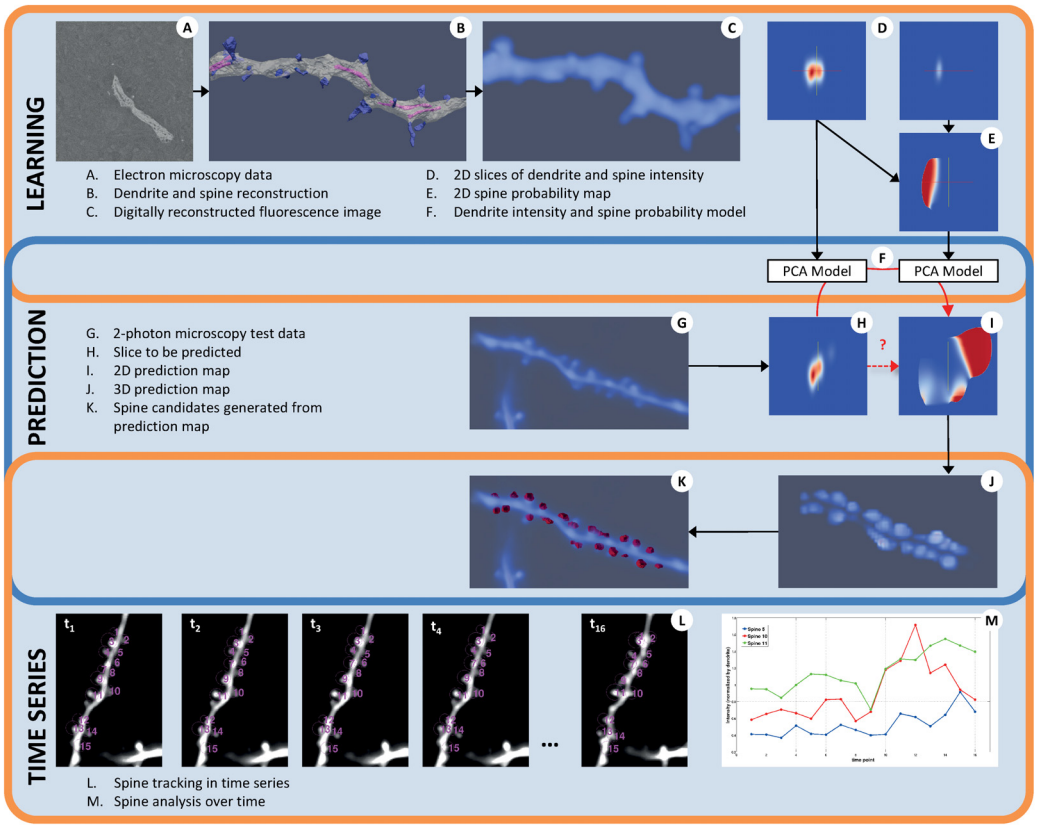


Figure A.1.: Overview of the whole concept except backbone parallel features.

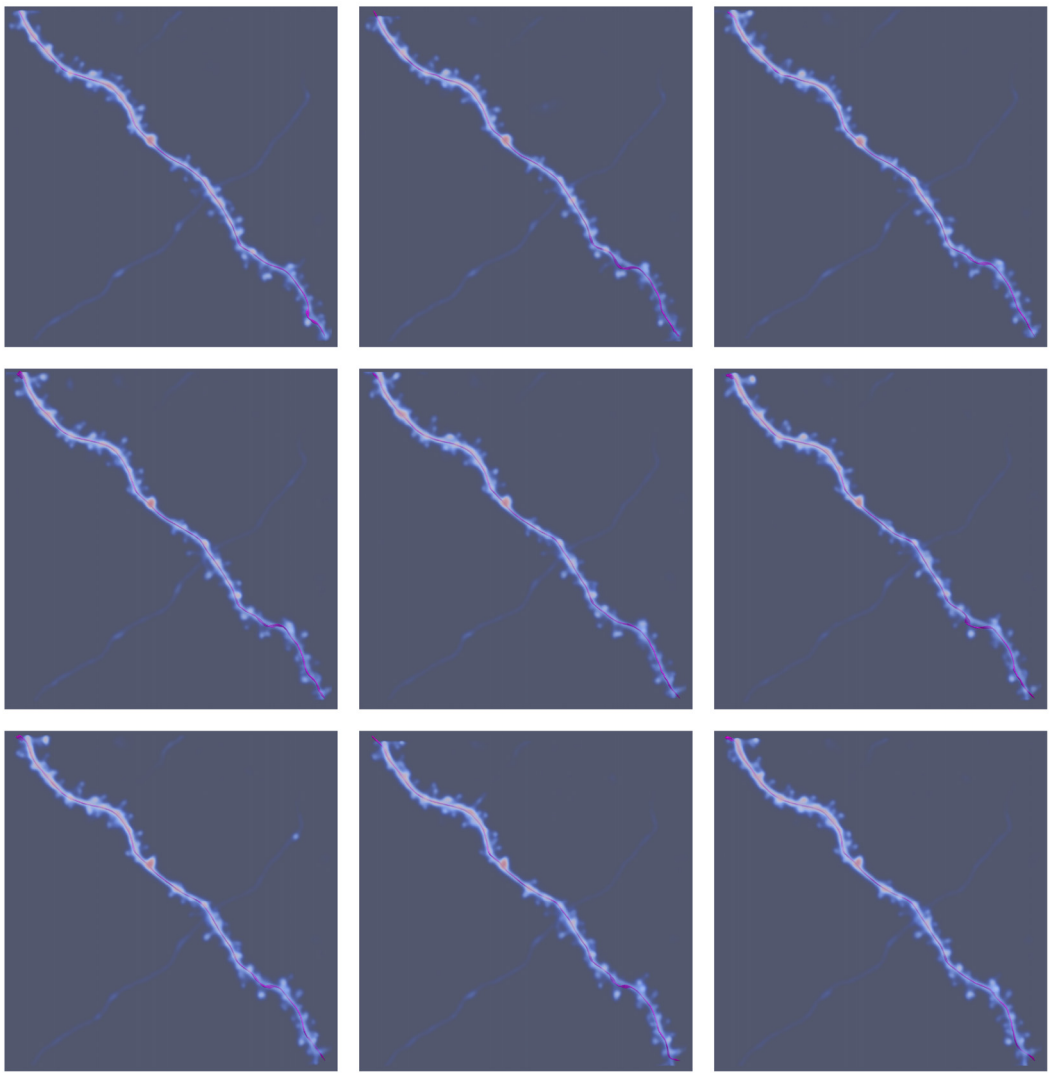


Figure A.2.: Volume rendering of input image with its backbone in *pink*. *First row:* Corresponds to $t = 0$, $t = 4$ and $t = 8$. *Second row:* Corresponds to $t = 12$, $t = 16$ and $t = 20$. *Third row:* Corresponds to $t = 24$, $t = 28$ and $t = 30$.

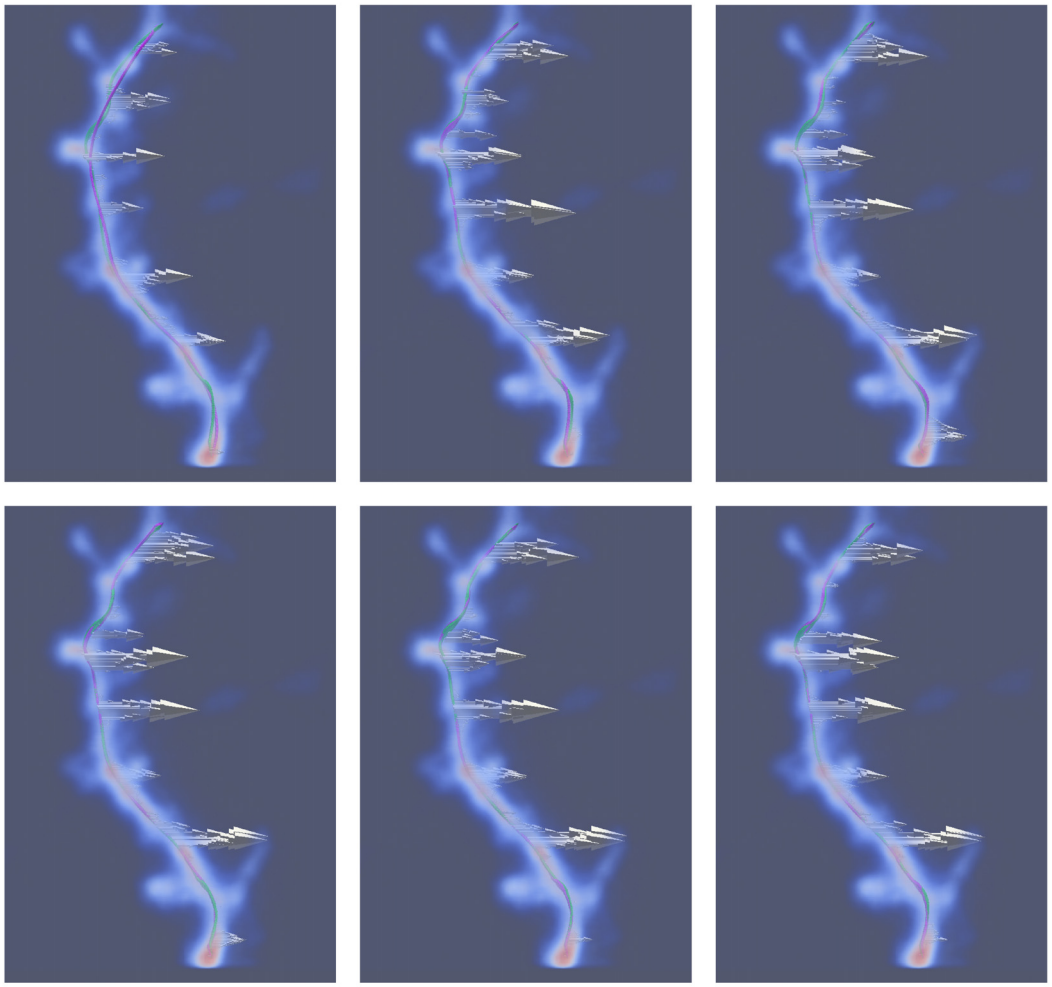


Figure A.3.: Optimization of the backbone over six iterations. In *pink* the initialization and in *green* the optimized backbone of each time point is shown. The *gray* vectors are the maximal probability of the shifted slices. *First row:* Iteration 1 to 3. *Second row:* Iteration 4 to 6.

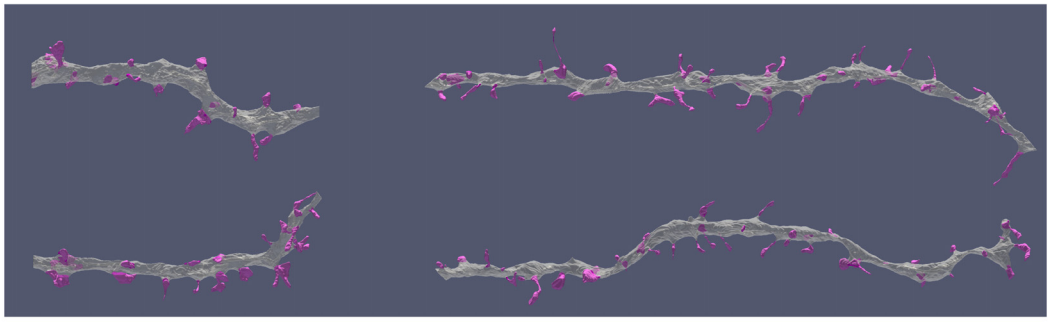


Figure A.4.: *Left, top and bottom:* Reconstruction of Dendrite4 and Dendrite6. *Right, top and bottom:* Reconstruction of DendriteA and DendriteB.

A.4. Spine Prediction by Model

In Section 5.4, Figure 5.9, we presented results of the reconstruction of dendrite intensity slices and the prediction of the spine probability maps. We show here some more examples in Figure A.5 for using one model and in Figure A.6 for the spine orientation dependent models. We use the model ModelAB which is based on DendriteA and DendriteB. The test data is coming from DRFI computed for Dendrite4 and Dendrite6. In Figure A.4 we show the reconstruction of the four pieces of dendrite. The *first column* is Dendrite4 and Dendrite6 and the *second column* DendriteA and DendriteB.

In Figure A.5 for DRFI input data dendrite intensity reconstructions and spine predictions are showed. The *first to third column* are the input data and reconstruction using 50 and 1599 Principal Component (PC) respectively. The *fourth to sixth column* are the ground truth of spine probability and the spine prediction maps by using 50 and 1599 PC respectively. *Each row* corresponds to an example. We will highlight here some of the examples. In the *second row* for the prediction using 1599PCs a lot of noise and holes in the prediction are recognizable. In the 4^{th} *row* we can see how a more vertical spine can be reconstructed and predicted. The 8^{th} *row* demonstrates the independence of the spine probability from the spine intensity in the dendrite intensity slice. We have a strong and a weak spine at the same time. However, the prediction is similar strong for both spines like in the ground truth.

In Figure A.6 we show two results using the spine orientation dependent models M_d^k and M_s^k. We use 50 PCs. The *first four* and the *second four rows* correspond each to an example. We show the dendrite intensity slice input data and the 9 reconstructions using the models M_d^1 to M_d^9. Visually there is no big difference in these images. However, the probability $P(s_{d,i}|M_d^k)$ is quite different depending on the used model. Relative to the highest probability we have for the first example for each model the probabilities $p = \{0.34, 0.05, 0.20, 1.00, 0.55, 0.02, 0.09, 0.53, 0.00\}$ respectively. This shows that in the region where the strong intensities in the probability map are also the highest probability is reached. This region is R_4 (see Figure 5.11 in Section 5.5 for the region encoding). However, also region R_8 has a high probability and in that direction a spine is oriented too. In the second example we have the probabilities $p = \{0.82, 0.61, 0.30, 0.69, 0.85, 1.00, 0.97, 0.77, 0.13\}$. They are all (except the last one) in a similar range. However, at the bottom location where also a small region with spine exists we have the highest probability (R_6). Furthermore, for the model in region

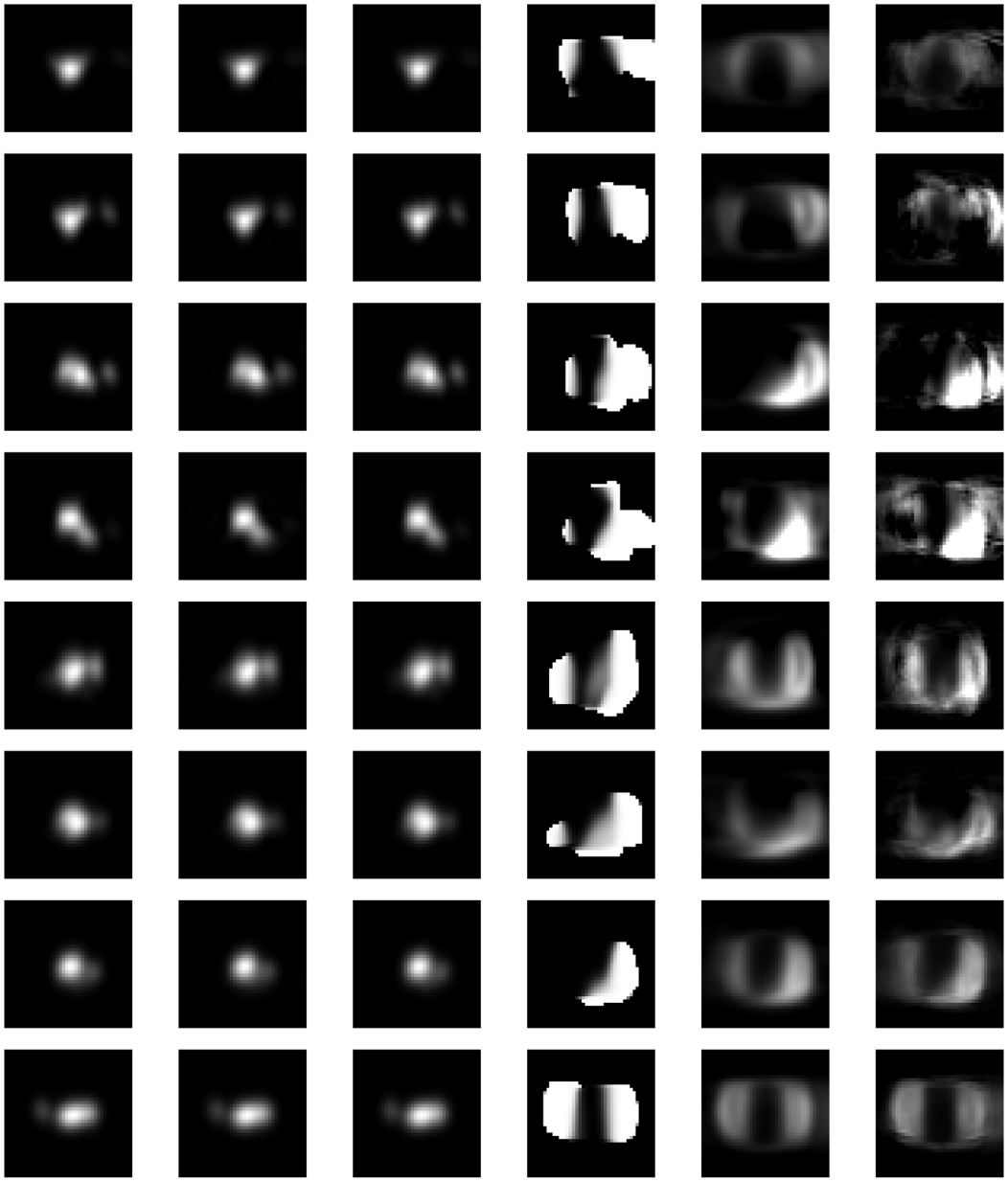


Figure A.5.: Every line corresponds to an example. *First column:* DRFI dendrite intensity slice. *Second and third column:* Reconstruction of dendrite intensity slice using 50 and 1599 PCs. *Fourth column:* Ground truth of spine probability. *Fifth and sixth column:* Prediction of spine probability using 50 and 1599 PCs.

R_9 the probability is clearly larger zero. This is remarkable as in most cases the models with spine have dramatically higher probabilities.

A.5. Spine Orientation

Table A.2 shows a detailed list about the spines reconstructed in data from SBFS-EM with the corresponding locations and computed angles.

Spine	Position			Spine Axis			Angle
	x(µm)	y(µm)	z(µm)	x	y	z	0°-180°
1	30.1	24.5	12.1	-0.6	-0.7	-0.4	112.5
2	20.0	38.8	11.3	0.9	0.2	-0.3	106.8
3	7.8	39.7	12.9	0.1	0.4	0.9	24.0
4	23.8	32.3	11.5	-0.1	-0.2	-1.0	165.5
5	31.1	46.7	0.5	0.6	0.5	-0.7	131.2
6	27.6	48.3	1.9	-0.1	0.9	-0.5	118.5
7	30.0	45.5	0.6	-0.2	-0.4	-0.9	151.7
8	22.5	45.8	6.8	-0.3	-0.1	-0.9	159.1
9	35.1	43.0	2.4	1.0	0.0	-0.1	97.7
10	23.3	32.0	13.2	-0.4	-0.5	0.8	39.5
11	29.4	47.0	1.6	0.1	1.0	-0.1	97.2
12	33.6	24.6	13.6	0.0	0.8	0.6	53.3
13	11.2	42.8	11.8	-0.4	0.9	0.1	85.9
14	26.2	47.1	6.1	0.8	0.6	0.3	71.0
15	9.0	40.6	12.7	-0.2	0.3	0.9	23.7
16	4.8	37.7	13.0	0.3	0.3	0.9	25.1
17	34.6	42.4	3.6	0.3	-0.4	0.9	29.9
18	22.2	34.7	11.4	0.2	0.4	-0.9	154.3
19	25.0	32.2	11.1	0.1	0.3	-0.9	161.2
20	8.0	39.1	10.8	0.0	-0.2	-1.0	171.1
21	1.3	35.0	12.8	-0.6	-0.8	0.2	76.7
22	20.7	36.4	12.4	0.8	0.6	0.1	81.4
23	33.5	44.8	3.1	0.7	0.7	0.3	72.1
24	28.0	48.2	2.9	0.4	0.9	0.3	71.7
25	23.5	33.4	11.5	0.3	0.4	-0.9	150.7
26	27.4	31.7	13.0	-0.2	0.5	0.8	34.4
27	9.4	41.3	11.3	-0.3	0.7	-0.7	130.9
28	18.4	36.8	12.1	-0.9	-0.3	-0.1	93.6
29	19.0	41.4	12.0	0.6	0.8	0.3	74.1
30	17.2	40.1	11.6	-0.6	-0.8	0.2	80.0
31	12.1	43.4	10.5	-0.7	0.4	-0.6	125.6
32	18.6	45.8	10.4	0.6	-0.3	0.8	39.2
33	24.4	45.7	4.7	-0.8	-0.6	-0.3	106.0
34	20.6	47.2	8.4	0.2	1.0	-0.1	96.5
35	16.5	45.8	10.7	-0.4	0.8	0.5	58.3
36	13.1	44.6	10.7	-0.8	0.4	0.3	70.2

Spine	Position			Spine Axis			Angle
37	24.9	46.1	6.8	0.8	-0.1	0.5	57.6
38	14.6	41.8	11.7	-0.3	-0.7	0.6	52.1
39	16.9	46.2	9.9	-0.5	0.7	-0.5	120.5
40	18.4	41.5	10.9	0.8	0.3	-0.5	120.0
41	19.2	38.7	11.0	-0.2	0.0	-1.0	167.3
42	24.1	46.8	7.3	0.1	0.7	0.7	46.2
43	18.7	45.6	9.1	0.0	-0.1	-1.0	174.0
44	26.0	46.1	3.6	-0.1	-1.0	-0.2	102.9
45	22.6	45.3	7.9	0.3	-0.5	0.8	39.4
46	31.3	26.7	13.4	0.7	0.6	0.5	61.1
47	15.0	45.2	11.1	-0.6	0.5	0.6	52.2
48	20.8	45.7	9.0	0.1	-0.4	0.9	27.6
49	21.0	34.6	12.8	-0.2	-0.5	0.8	32.8
50	28.7	31.2	12.4	0.7	0.7	0.1	84.4
51	30.2	45.4	2.7	-0.6	0.0	0.8	37.4
52	35.2	22.8	13.1	0.9	-0.4	0.1	86.9
53	32.3	46.1	3.4	0.3	0.8	0.5	63.1
54	34.4	23.3	12.0	0.8	-0.2	-0.6	127.4
55	25.6	46.7	3.5	-0.9	0.0	-0.5	116.8
56	27.2	46.1	3.3	0.3	-0.9	0.2	76.3
57	16.3	41.3	10.7	-0.4	-0.5	-0.8	142.5
58	21.4	35.2	12.9	0.6	0.3	0.7	43.5
59	34.3	23.6	13.4	0.7	-0.7	0.1	84.1
60	23.0	45.9	7.7	0.0	0.2	1.0	10.3
61	32.6	22.5	13.0	-0.5	-0.9	0.1	84.7
62	19.6	38.8	11.9	0.7	0.2	0.6	51.7
63	38.1	40.7	2.7	0.8	0.6	-0.1	96.9
64	39.1	17.6	9.5	-0.9	-0.5	-0.1	95.6
65	53.0	28.3	4.1	0.9	0.3	-0.4	116.0
66	51.8	7.3	4.8	-0.9	0.2	-0.2	104.4
67	42.1	36.4	5.3	-0.2	0.0	1.0	9.7
68	37.1	40.3	2.3	-0.7	0.0	-0.7	138.5
69	55.1	24.3	6.5	-0.2	-0.6	0.8	41.2
70	38.0	17.8	11.5	-0.1	-0.9	0.5	59.9
71	54.1	25.0	5.5	-0.7	-0.7	0.1	85.5
72	50.8	30.3	4.4	0.6	0.2	0.8	36.2
73	67.1	15.5	1.7	0.3	0.3	-0.9	153.9
74	64.7	17.6	1.4	-0.1	0.1	-1.0	172.8
75	65.7	17.7	0.8	0.5	0.4	-0.8	142.2
76	37.1	39.8	3.7	-0.8	-0.3	0.5	60.6
77	46.3	33.5	3.7	0.2	0.6	0.8	37.5
78	36.1	22.2	11.6	-0.5	0.9	0.1	86.9
79	46.7	31.8	4.5	-0.5	-0.5	0.7	48.1
80	51.2	9.1	6.6	-0.7	-0.7	-0.1	93.4
81	57.7	21.7	5.6	-0.6	-0.5	0.6	51.6
82	50.3	28.5	3.9	-0.7	-0.6	0.3	70.7

Spine	Position			Spine Axis			Angle
83	45.7	32.7	4.3	-0.3	-1.0	0.2	81.2
84	56.6	26.2	6.2	0.4	0.8	0.5	62.4
85	61.0	22.1	4.0	0.6	0.7	-0.5	119.3
86	56.4	25.3	6.2	-0.5	-0.8	-0.1	98.0
87	51.5	9.8	6.1	-0.4	0.4	-0.9	148.5
88	37.5	21.2	12.3	0.2	0.8	0.6	51.4
89	54.9	4.6	3.8	0.4	0.4	-0.8	144.3
90	63.4	19.4	4.9	0.5	-0.2	0.8	33.9
91	63.7	18.3	3.2	-0.3	-0.9	-0.2	104.0
92	60.3	19.4	5.0	-0.6	-0.7	0.3	74.3
93	40.8	16.9	8.6	-0.4	0.3	-0.9	152.0
94	67.1	17.6	2.1	0.9	0.5	0.1	86.0
95	41.7	35.4	4.3	-0.7	-0.7	-0.3	107.4
96	44.3	33.5	4.3	-0.8	0.5	-0.2	99.5
97	39.0	36.8	5.0	-0.5	-0.7	0.4	63.6
98	60.3	22.3	3.9	0.8	0.4	-0.4	115.4
99	52.5	27.8	3.9	0.3	-0.6	-0.8	139.6
100	57.1	3.6	3.5	0.9	0.0	-0.4	111.0
101	44.7	16.1	8.6	0.0	0.5	-0.9	152.8
102	52.3	6.8	4.5	-0.6	-0.4	-0.7	132.5
103	45.1	34.5	4.4	0.6	0.1	0.8	36.8
104	49.9	31.1	4.4	0.2	0.8	0.5	57.7
105	64.6	19.6	2.7	0.2	0.9	-0.3	106.5
106	43.6	35.6	3.7	-0.1	0.3	-1.0	164.1
107	65.1	19.5	2.8	0.6	0.8	-0.1	97.1
108	56.5	26.0	4.3	0.3	0.2	-0.9	161.1
109	67.8	13.7	2.2	0.1	-0.3	-1.0	162.2
110	56.7	23.3	4.4	-0.5	-0.8	-0.2	103.1
111	69.0	13.4	3.6	0.2	1.0	0.2	79.7
112	51.3	12.4	8.1	0.7	0.6	-0.3	105.9
113	46.9	13.2	9.3	-0.5	-0.8	0.2	75.7
114	50.6	10.2	7.0	-0.9	-0.1	-0.5	117.0
115	39.5	38.4	5.0	0.3	0.3	0.9	24.3
116	52.6	6.1	5.4	-0.5	-0.6	0.7	49.0
117	45.5	33.5	2.2	0.0	-0.5	-0.9	151.7
118	45.7	14.3	9.5	-0.3	-0.9	-0.1	93.2
119	42.7	16.3	8.9	0.3	0.6	-0.7	134.8
120	44.2	34.8	3.3	-0.3	0.1	-0.9	159.9
121	57.5	24.4	5.3	0.8	0.1	0.6	53.1
122	47.5	14.0	8.4	-0.5	-0.2	-0.8	143.6
123	68.2	11.6	4.3	-0.8	-0.5	0.3	72.6
124	56.1	2.4	4.2	0.6	-0.7	-0.1	98.2
125	48.1	12.9	8.3	-0.5	-0.5	-0.7	134.1
126	60.9	19.3	4.7	-0.2	-1.0	0.0	87.3
127	41.3	37.7	4.5	0.3	1.0	0.0	89.3
128	64.4	17.8	3.0	-0.3	-0.8	0.5	57.3

Spine	Position			Spine Axis			Angle
129	44.7	15.0	10.3	0.0	-0.6	0.8	39.4
130	42.7	15.5	9.3	0.0	-0.7	-0.7	136.7
131	64.4	17.7	3.1	-0.3	-0.8	0.5	57.9
132	45.6	33.7	2.6	0.0	-0.3	-1.0	163.7
133	51.7	12.7	8.0	0.8	0.6	-0.3	106.2
134	30.2	19.1	10.3	-0.4	0.3	0.9	29.1
135	31.6	18.8	9.3	0.3	0.5	0.8	32.4
136	11.8	12.0	3.1	-0.1	0.8	-0.6	129.4
137	22.3	6.1	5.0	0.3	0.3	0.9	27.0
138	27.1	3.6	4.0	0.2	-0.9	-0.4	111.7
139	32.9	17.4	8.3	-0.1	-1.0	0.0	92.0
140	19.2	14.2	4.6	-0.4	-0.1	0.9	26.3
141	16.9	4.8	4.6	-0.5	0.4	0.7	41.6
142	23.5	4.5	4.7	-0.4	-0.9	0.2	80.9
143	26.6	17.8	6.0	0.2	0.7	-0.7	136.4
144	13.7	13.4	5.1	-0.6	0.4	0.7	44.0
145	37.3	20.0	9.7	0.7	0.0	0.7	47.1
146	11.1	10.9	2.8	-0.1	0.1	-1.0	172.0
147	10.4	5.9	1.7	-0.3	-0.1	-0.9	158.8
148	19.6	14.7	4.1	0.1	1.0	0.0	89.2
149	20.3	5.7	4.4	-0.7	0.6	0.3	69.6
150	25.6	4.7	5.8	-0.3	-0.1	0.9	20.4
151	27.7	5.2	3.3	-0.1	0.1	-1.0	172.8
152	30.3	6.9	5.3	-0.3	0.7	0.6	50.9
153	18.0	15.3	3.7	0.0	1.0	-0.1	95.0
154	34.3	6.8	5.2	-0.1	0.1	1.0	9.1
155	13.8	5.2	2.6	-0.6	-0.5	-0.6	129.4
156	24.2	17.8	6.7	-0.7	0.7	0.3	74.2
157	13.7	5.4	4.0	0.1	-0.6	0.8	38.4
158	33.2	17.9	7.4	-0.3	-0.1	-1.0	162.6
159	32.2	5.6	5.1	0.4	-0.7	0.5	57.2
160	25.2	15.4	7.2	0.6	-0.5	0.6	54.9
161	23.4	5.1	3.1	0.4	-0.4	-0.8	146.0
162	23.8	14.3	5.1	0.9	-0.2	-0.4	114.9
163	16.2	14.8	4.8	0.1	1.0	0.2	77.5
164	27.2	17.8	7.4	-0.1	-0.4	0.9	26.6
165	36.8	20.1	8.1	0.5	0.1	-0.8	147.1
166	32.3	7.1	5.2	0.1	0.3	0.9	21.0
167	13.6	11.0	5.0	0.4	-0.8	0.4	67.8
168	35.9	6.3	4.0	0.4	-0.5	-0.7	135.6
169	13.2	12.9	3.6	-0.3	0.5	-0.8	144.8
170	34.8	19.2	8.7	-0.8	0.5	0.2	79.9
171	24.9	5.6	4.8	0.2	0.9	0.3	69.7
172	16.8	4.0	2.9	-0.5	-0.8	-0.3	110.3
173	12.3	6.9	3.8	-0.4	0.5	0.7	43.7
174	8.8	9.8	3.2	-0.8	0.5	-0.2	104.0

Spine	Position			Spine Axis			Angle
175	23.5	6.1	4.5	0.0	1.0	0.2	77.0
176	16.8	5.4	3.4	0.1	0.8	0.5	58.4
177	9.1	9.4	2.3	0.0	0.0	-1.0	178.7
178	29.8	5.0	4.5	0.7	-0.7	0.2	79.8
179	17.4	3.6	3.3	0.1	-1.0	-0.1	93.2
180	15.4	12.6	4.1	0.2	-0.9	-0.3	106.6
181	17.7	3.9	4.7	-0.2	-0.3	0.9	20.9
182	24.5	16.7	5.6	-0.5	0.2	-0.8	145.4
183	23.1	15.2	5.4	-0.4	0.9	-0.2	100.0
184	11.8	10.3	3.7	0.6	-0.8	0.0	90.0
185	23.9	14.2	6.3	0.3	-0.8	0.4	63.4
186	21.3	13.5	4.6	-0.1	-1.0	-0.1	94.0
187	11.2	10.9	4.2	-0.2	0.1	1.0	14.0
188	12.7	6.9	2.9	0.1	1.0	-0.3	105.0
189	36.4	20.0	7.3	0.0	0.0	-1.0	176.9
190	34.7	7.3	3.9	0.3	0.8	-0.6	126.6
191	22.3	13.5	4.8	0.1	-1.0	0.1	86.1
192	6.3	8.2	2.2	-0.6	0.6	0.5	58.7
193	15.9	14.3	5.1	-0.3	0.3	0.9	25.2
194	25.8	16.0	6.2	0.7	-0.7	-0.1	93.7
195	16.3	13.6	5.3	0.4	-0.6	0.7	46.3
196	35.1	6.3	5.4	0.0	-0.6	0.8	35.7
197	8.9	9.2	4.0	-0.4	0.0	0.9	22.5
198	10.3	10.6	4.4	0.0	0.5	0.9	27.0
199	28.0	18.6	6.2	0.2	0.3	-0.9	159.7
200	20.7	4.9	4.1	0.6	-0.8	-0.1	93.4
201	26.5	16.7	7.5	-0.1	-0.6	0.8	35.4
202	37.0	7.2	4.5	1.0	0.2	-0.1	96.8
203	19.8	4.6	3.8	0.1	-0.1	-1.0	169.0
204	12.5	5.8	2.2	0.4	-0.5	-0.8	139.8
205	23.0	13.7	4.4	0.6	-0.5	-0.7	133.9
206	26.1	15.1	7.2	0.8	-0.4	0.3	71.0
207	26.4	4.7	5.2	0.4	-0.2	0.9	23.9
208	17.5	3.7	4.0	-0.4	-0.8	0.4	66.0
209	23.4	4.0	4.8	-0.3	-0.9	0.2	77.9
210	18.7	4.3	4.6	-0.1	0.4	0.9	25.2
211	23.1	4.6	4.9	-0.8	-0.6	0.3	73.5
212	29.7	18.8	11.0	-0.3	0.2	0.9	19.9
Mean	32.1	22.9	6.3	0.0	0.0	0.0	88.9

Table A.2.: Location and orientation of spines in SBFS-EM data used for computation of spine orientation.

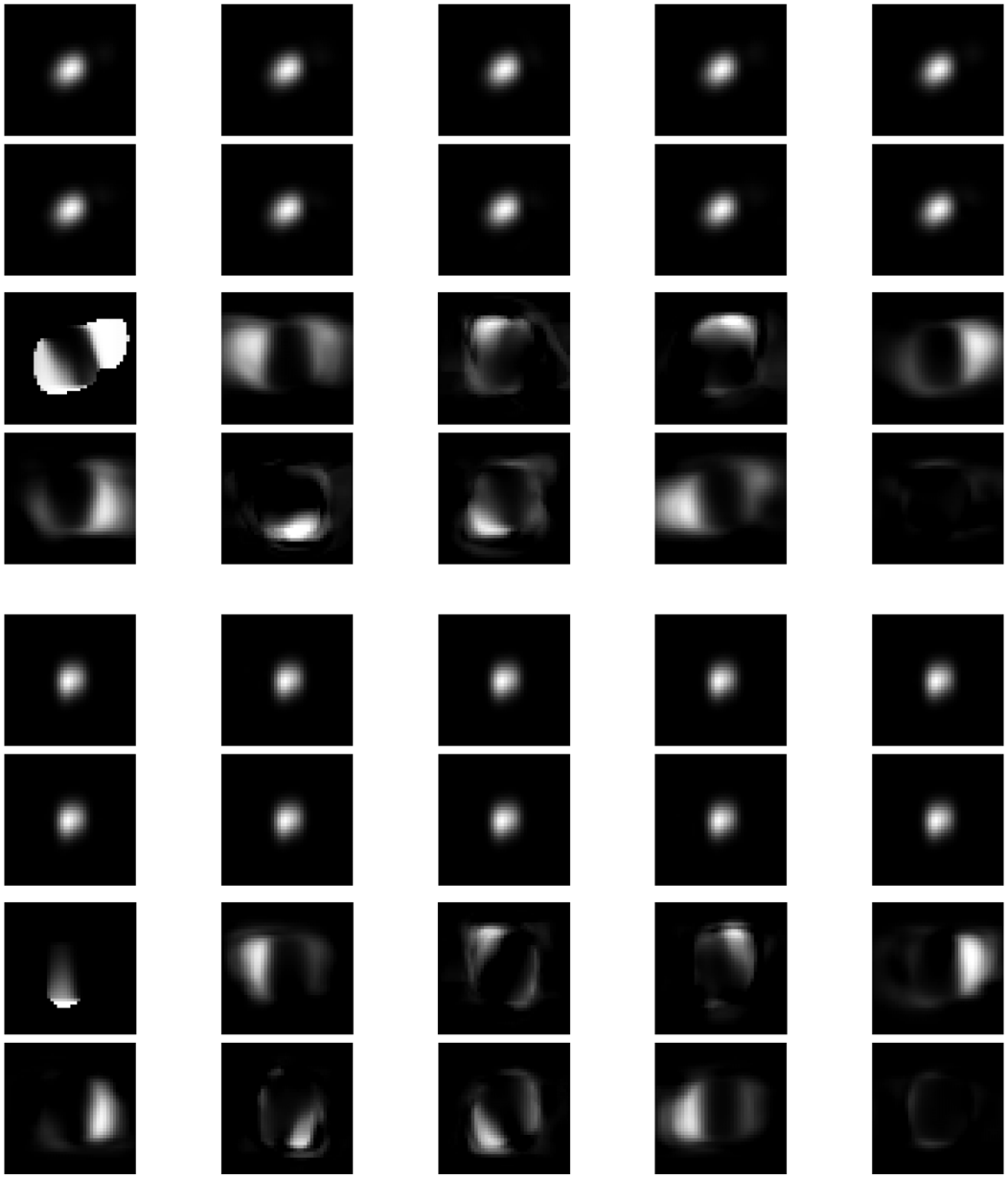


Figure A.6.: Row *1 to 4* and *5 to 8* correspond each to an example. *First and second row:* Dendrite intensity slice (*top left*) and the reconstructions using models M_d^1 to M_d^9 (*left to right*). *Third and fourth row:* Spine probability map (*top left*) and the 9 predictions based on dendrite intensity slice and models M_d^k and M_s^k. *5^{th} to 8^{th} row:* Analogously to first example.

B. Software

During this thesis different software was developed. In addition to the software that is required to generate training data, register it and learn the models also an user application for spine detection, segmentation and statistical value computation was implemented. In this chapter we will give a rough overview about the software and what it can be used for. The software itself is divided in to a library (SpineCheckerLib, see Section B.1) and the application itself (SpineChecker, Section B.2).

The software is still a research application but can be used in laboratories from instructed users. The software is implemented in C++ and uses Insight Segmentation & Registration Toolkit - `http://www.itk.org` (ITK), Visualization Toolkit - `http://www.vtk.org` (VTK) and QT Framework - `http://www.qt-project.org` (QT). Because of larger memory requirements only 64 bit versions are useful. Depending on the input data the memory peak can be more than 8 GB.

B.1. SpineCheckerLib

The SpineCheckerLib library processes all core computation and holds the images and so on. It has methods and classes to optimize the backbone, predict spines based on the statistical dendrite and spine probability models, integrate backbone parallel features and compute the final binary spine candidates. Furthermore, one component computes for all found spines the statistical values like mean intensity per channel and others (see Table 7.2).

The administrative parts of SpineCheckerLib are a class holding the spine prediction model, a central class managing all spine data, additional tools for IO interaction of images and settings and a class managing the individual process steps. All these classes are used to coordinate at a central location all information during all the processing steps. Thanks to the class managing the process steps an application does not have to deal with the core analysis classes but can just call the corresponding methods of the processing steps class.

The core of analysis is divided into different classes with its specific task each one. In these classes the corresponding tasks are solved and the information is directly loaded from the central data management. Encapsulated are all these steps and tasks like already mentioned in a class managing the processing steps. The algorithmic background of all these core analysis classes was described within the thesis. We will not introduce in more detail the software design of the library as it is a research application and the design itself is out of scope of this thesis.

B.2. SpineChecker

SpineChecker is the application using SpineCheckerLib and providing a Graphical User Interface (GUI) for the user. It provides the possibility of stepwise processing of single

Figure B.1.: *Left:* In the first step an interaction with the input is possible. The seed points for the backbone can be clicked and some settings are possible. *Middle:* Result of the spine prediction as volume rendering. *Right:* Final binarized result as spine surfaces combined with the volume rendering of the input data. The *green* spine is picked for interaction (e.g. deleting). For a different spine the statistic is displayed.

images or even batch-processing of complete time series. We will give a short overview of the user interaction and what the user gets as final results.

Stepwise Processing

By the stepwise processing the user is guided through the analysis process and gets some intermediate results. This stepwise processing is also motivated to give the possibility of changing settings and see the differences ad hoc.

In the first step the user can:

- Load an image
- Preview the image
- Crop the processing region
- Change the volume channel
- Click seed points (multiple pieces possible)

Clicking the seed points and cropping are the only manual interactions the user can and has to do. Instead of clicking seed points an initialization by 3D binary thinning is also feasible but much less instable. Theoretically also a backbone initialization from any other algorithm/software would be feasible.

Afterwards the user gets the result of backbone initialization presented (thinning or seed points) and can in the next step pick the model to be used. Then the prediction result is displayed for control purposes. Integrated are already all features, also the backbone-parallel ones. In a last step the user picks whether to set the manual threshold or use the more adaptive automatic segmentation to get the spines or spine candidates. As final result the user gets presented a volume rendering of the input data with overlaid surfaces of the spines. The spine statistics can be exported and an interaction with all spines is possible (removing, merging and splitting spines).

In Figure B.1 we give an overview of the most important steps and results. From *left to right* we show the image preview and clicking of seed points, the spine prediction result and the final rendering of the result combined of the volume rendering of the input and the spine surfaces.

Batch Processing

In batch processing all processing steps are running in background. The user can pick a initial file (first time point) and click seed points. Furthermore, it can be specified which files (time points) should be processed within the same directory. The processing is then fully automatic over all time points and the backbone of each time point is initialized by the backbone of the previous time point. All time points are aligned to the first time point. As result the user gets some result images and the summary about the spine candidates statistics (identical to statistic showed in Figure B.1, *right*). By a Matlab post analysis script the user can then get the traces of statistical spine values. Also the path finding over all spine candidates takes place in the post analysis script. The process of tracking the spines in the time series is described in more detail in Section 7.2.4.

The batch processing is a required component such that the software can and is used in biological experiments. This corresponds also to the requirement that biologists want to find from time series, by less interaction as possible, an analysis of all spines over all time points.

B.3. Tracking and Visualization

The spine detection and segmentation software delivers result files but does not provide a tracking of the spines nor a visualization of time series results. Therefore, in Matlab scripts we implemented the tracking concept described in Section 7.2.4. Furthermore, in these scripts a visualization of the spine changes over time and the spine detection over the multiple time points is given. The decision of using Matlab scripts and not to integrate it into the SpineChecker software was the higher flexibility provided for biologists. Especially, the visualization and computation of classification into groups can be very specific to the experiments. Matlab is well accepted and a common tool for biologists. Therefore, we try to give the biologists with the scripts a high freedom of adapting the visualization and further processing of the analysis results. All this without having deeper knowledge in programming C++ and to setup a more complex environment to change the SpineChecker software. Conceptually SpineChecker provides from batch analysis a text file with the spine candidates and its statistical values. The biologist has full choice on how to further process these files or to use the tracking scripts. We show results of the tracking and visualization in Section 7.3. We refer the reader to Section 7.2.4 for a deeper understanding of the conceptual details of tracking the spines.

In principle the tracking and visualization scripts are targeting at results of time series. However, it is also feasible to use them for single time points and to generate visualization. Such a result is shown in Section 6.6.2, Figure 6.10.

List of Figures

List of Tables

List of Abbreviations

1D 1-Dimensional . . . 16
2D 2-Dimensional . . . 115
2PA 2 Photon Absorption . . . 12
2PM 2-Photon Microscopy . . . 111
2PEF 2-Photon Excitation Fluorescence . . . 13
3D 3-Dimensional . . . 115
3DMM 3D Morphable Model . . . 24
4D 4-Dimensional (3-Dimensional plus time) . . . 3
ASM Active Shape Model . . . 24
AAM Active Appearance Model . . . 24
BFM Basel Face Model . . . 27
CAD Computer Aided Design . . . 21
CARS Coherent Anti-stokes Raman Scattering . . . 13
CLEM Correlative Light Electron Microscopy . . . 108
CT Computer Tomography . . . 10
DNA Deoxyribonucleic acid . . . 9
DRFI Digitally Reconstructed Fluorescence Image . . . 115
DFT Discrete Fourier Transformation . . . 35
EM Electron Microscopy . . . 108
ER Endoplasmic Reticulum . . . 109
FFT Fast Fourier Transformation . . . 35
FN False Negative . . . 108
FP False Positive . . . 108
FWHM Full Width Half Maximum . . . 33
GFP Green Fluorescent Protein . . . 11
GUI Graphical User Interface . . . 127
ITK Insight Segmentation & Registration Toolkit - `http://www.itk.org` 127
LSCM Laser-Scanning Confocal Microscopy . . . 111
LSM Laser-Scanning Microscopy . . . 12
MIP Maximum Intensity Projection . . . 103
MPM Multi-Photon Microscopy . . . 10

MRI Magnet Resonance Imaging . . . 10
MRT Magnet Resonance Tomography . . . 10
NIR Near Infrared . . . 13
NA Numerical Aperture . . . 33
PC Principal Component . . . 119
PCA Principal Component Analysis . . . 115
PMT Photon Multiplier Tube . . . 12
PSF Point Spread Function . . . 108
QT QT Framework - `http://www.qt-project.org` . . . 127
ROI Region Of Interest . . . 113
SBFS-EM Serial Block-Face Scanning Electron Microscopy . . . 115
SHG Second Harmonic Generation . . . 13
SNR Signal Noise Ratio . . . 112
SVD Singular Value Decomposition . . . 56
TEM Transmission Electron Microscopy . . . 21
TN True Negative . . . 108
TP True Positive . . . 108
VTK Visualization Toolkit - `http://www.vtk.org` . . . 127

Bibliography

[1] Thomas Albrecht and Thomas Vetter. Automatic fracture reduction. In *Mesh Processing in Medical Image Analysis 2012*, pages 22–29. Springer, 2012.

[2] Carlo Arcelli, Gabriella Sanniti di Baja, and Luca Serino. Distance-driven skeletonization in voxel images. *IEEE PAMI*, 33(4), April 2011.

[3] Matthias Asal. Segmentierung und Klassifikation dendritischer Spines. Master's thesis, Albert-Ludwig-Universität, Freiburg, 2008.

[4] Cagri Aslan, Aykut Erdem, Erkut Erdem, and Sibel Tari. Disconnected skeleton: Shape at its absolute scale. *IEEE PAMI*, 30(12), December 2008.

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of Siggraph 99*, pages 187–194, 1999.

[6] Sylvain Bouix, Kaleem Siddiqi, and Allen Tannenbaum. Flux driven automatic centerline extraction. *Medical Image Analysis*, 9:209–221, 2005.

[7] Michael J. Byrne, M. Neal Waxham, and Yoshihisa Kubota. The impacts of geometry and binding on camkii diffusion and retention in dendritic spines. *Journal of Computational Neuroscience*, 31(1):1–12, 2011.

[8] Raymond B. Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

[9] Martin Chalfie, Yuan Tu, Ghia Euskirchen, William W. Ward, and Douglas C. Prasher. Green fluorescent protein as a marker for gene expression. *Science*, 263:802–805, January 1994.

[10] Jie Cheng, Xiaobo Zhou, Eric Miller, Rochelle M. Witt, Jinmin Zhu, Bernardo L. Sabatini, and Steven T.C. Wong. A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *Journal of Neuroscience Methods*, 165(1):122–134, 2007.

[11] Jie Cheng, Xiaobo Zhou, Eric L. Miller, Veronica A. Alvarez, Bernardo L. Sabatini, and Stephen T. C. Wong. Oriented markov random field based dendritic spine segmentation for fluorescence microscopy images. *Neuroinform*, 8:157–170, 2010.

[12] Dmitri B. Chklovskii, Shiv Vitaladevuni, and Louis K. Scheffer. Semi-automated reconstruction of neural circuits using electron microscopy. *Current Opinion in Neurobiology*, 20(5):667–675, 2010.

[13] Guilherme Cintra, editor. *The Pharmaceutical Industry and Global Health: Facts and Figures*. International Federation of Pharmaceutical Manufacturers and Associations (IFPMA), 2011 edition, 2011.

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

[15] Guy Cox and Colin J.R. Sheppard. Practical limits of resolution in confocal and non-linear microscopy. *Microscopy Research and Technique*, 63(1):18–22, January 2004.

[16] Peter Dayan and Laurence F. Abbott. *Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2005.

[17] Cees J. de Grauw, Jurrien M. Vroom, Hans T. M van der Voort, and Hans C. Gerritsen. Imaging properties in two-photon excitation microscopy and effects of refractive-index mismatch in thick specimens. *Applied Optics*, 38(28):5995–6003, October 1999.

[18] Herve Delingette, Xavier Pennec, Luc Soler, Jacques Marescaux, and Nicholas Ayache. Computational models for image-guided robot-assisted and simulated medical interventions. *Proceedings of the IEEE*, 94(9):1678–1688, September 2006.

[19] Winfried Denk and Heinz Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biology*, 2(11):1900–1909, November 2004.

[20] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting face images using active appearance models. In *Proceedings of Automatic Face and Gesture Recognition*, pages 300–305, April 1998.

[21] Mans Ehrenberg. The green fluorescent protein: discovery, expression and development. `http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2008/advanced.html`, 2008.

[22] Julien Ghaye, Giovanni De Micheli, and Sandro Carrara. Simulated biological cells for receptor counting in fluorescence imaging. *BioNanoScience*, 2(2):94–103, June 2012.

[23] Louis Guttman. Some necessary conditions for common-factor analysis. *Psychometrika*, 19(2):149–161, June 1954.

[24] Tiancheng He, Zhong Xue, Yong Kim, and Stephen T. Wong. Three-dimensional dendritic spine detection based on minimal cross-sectional curvature. In *International Symposium on Biomedical Imaging (ISBI)*. IEEE, May 2012.

[25] Bernd Heisele, Tomaso Poggio, and Massimiliano Pontil. Face detection in still gray images. Technical report, Massachusetts Institute of Technology, 2000.

[26] Fritjof Helmchen and Winfried Denk. Deep tissue two-photon microscopy. *Nature Methods*, 2(12):932–940, December 2005.

[27] Moritz Helmstaedter, Kevin L. Briggman, and Winfried Denk. High-accuracy neurite reconstruction for high-throughput neuroanatomy. *Nature Neuroscience*, 14(8):1081–1088, August 2011.

[28] Niklaus Holbro, Asa Grunditz, and Thomas G. Oertner. Differential distribution of endoplasmic reticulum controls metabotropic signaling and plasticity at hippocampal synapses. *PNAS*, 106(35):15055–15060, 2009.

[29] Hanno Homann. Implementation of a 3d thinning algorithm. *Insight Journal*, October 2007.

[30] John L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185, 1965.

[31] Donald A. Jackson. Stopping rules in principal component analysis: A comparison of heuristical and statistical approaches. *Ecology*, pages 2204–2214, 1993.

[32] Firdaus Janoos, Kishore Mosaliganti, Xiaoyin Xu, Raghu Machiraju, Kun Huang, and Stephen T.C. Wong. Robust 3d reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical Image Analysis*, 13(1):167–179, February 2009.

[33] Verena Kaynig, Bernd Fischer, Elisabeth Müller, and Joachim M. Buhmann. Fully automatic stitching and distortion correction of transmission electron microscope images. *Journal of Structural Biology*, 171(2):163–173, August 2010.

[34] Ingrid Y. Y. Koh, W. Brent Lindquist, Karen Zito, Esther A. Nimchinsky, and Karel Svoboda. An image analysis algorithm for dendritic spines. *Neural Computation*, 14(6):1283–1310, 2002.

[35] Christoph H. Lampert, Matthew B. Blaschko, and Thomas Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE PAMI*, 31(12), December 2009.

[36] Gaetan Lehmann. Fft based convolution. *Insight Journal*, 2010.

[37] Qing Li and Zhigang Deng. A surface-based 3-d dendritic spine detection approach from confocal microscopy images. *IEEE Transactions on Image Processing*, 21(3), March 2012.

[38] Michael J. Nasse and Joerg C. Woehl. Realistic modeling of the illumination point spread function in confocal scanning optical microscopy. *Journal of the Optic Society of America*, 27(2):295–302, January 2010.

[39] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEE Trans. on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

[40] James B. Pawley, editor. *Handbook of Biological Confocal Microscopy*. Springer, third edition, 2006.

[41] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Proc. of International Conference on Advanced Video and Signal based Surveillance (AVSS)*, pages 296–301. IEEE, September 2009.

[42] Hanchuan Peng, Zongcai Ruan, Deniz Atasoy, and Scott Sternson. Automatic reconstruction of 3d neuron structures using a graph-augmented deformable model. *Bioinformatics*, 26(12):38–46, June 2010.

[43] Alfredo Rodriguez, Douglas B. Ehlenberger, Dara L. Dickstein, Patrick R. Hof, and Susan L. Wearne. Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PlosOne*, 3(4), 2007.

[44] Alfredo Rodriguez, Douglas B. Ehlenberger, Patrick R. Hof, and Susan L.Wearne. Three-dimensional neuron tracing by voxel scooping. *Journal of Neuroscience Methods*, 184:169–175, 2009.

[45] Stephan Schmitt, Jan Felix Evers, Carsten Duch, Michael Scholz, and Klaus Obermayer. New methods for the computer-assisted 3-d reconstruction of neurons from confocal image stacks. *NeuroImage*, 23:1283–1298, 2004.

[46] Wei Shen, Xiang Bai, Rong Hu, Hongyuan Wang, and Longin Jan Latecki. Skeleton growing and pruning with bending potential ratio. *Journal of Pattern Recognition*, 44(2):196–209, February 2011.

[47] Peng Shi, Xiaobo Zhou, Qing Li, Matthew Baron, Merilee A. Teylan, Yong Kim, and Stephen T.C. Wong. Online three-dimensional dendritic spines morphological classification based on semi-supervised learning. In *Proc. of Intl. Symposium on Biomedical Imaging*, pages 1019–1022, 2009.

[48] O. Shimomura. The discovery of aequorin and green fluorescent protein. *Journal of Microscopy*, 217(1):3–15, January 2005.

[49] Osamu Shimomura, Frank H. Johnson, and Yo Saiga. Extraction, purification and properties of aequorin, a bioluminescent protein from the luminous hydromedusan, aequorea. *Journal of Cellular Physiology*, 59(3):223–239, June 1962.

[50] J . Son, S. Song, S. Lee, S. Chang, and M. Kim. Morphological change tracking of dendritic spines based on structural features. *Journal of Microscopy*, 241(3):261–272, 2011.

[51] Hakan Toresson and Seth G. N. Grant. Dynamic distribution of endoplasmic reticulum in hippocampal neuron dendritic spines. *European Journal of Neuroscience*, 22(7):1793–1798, 2005.

[52] Roger Y. Tsien. The green fluorescent protein. *Annu. Rev. Biochem.*, 67(509-5), 1998.

[53] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[54] Tamas Varga and Horst Bunke. Generation of synthetic training data for an hmm-based handwriting recognition system. In *Proc. International Conference on Document Analysis and Recognition ICDAR*, pages 618–622, 2003.

[55] Wikipedia. Multivariate normal distribution. `http://en.wikipedia.org/wiki/Multivariate_normal_distribution`.

[56] Wikipedia. Neuron. `http://en.wikipedia.org/wiki/Neuron`.

[57] Wikipedia. Precision and recall. `http://en.wikipedia.org/wiki/Precision_and_recall`.

[58] Xiaosong Yuan, Joshua T. Trachtenberg, Steve M. Potter, and Badrinath Roysam. Mdl constrained 3-d grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7(4):213–232, 2009.

[59] Bo Zhang, Josiane Zerubia, and Jean-Christophe Olivo-Marin. Gaussian approximations of fluorescence microscope psf models. *Applied Optics*, 46(10):1819–1829, 2007.

[60] Yong Zhang, Kun Chen, Matthew Baron, Merilee A. Teylan, Yong Kim, Zhihuan Song, Paul Greengard, and Stephen T.C. Wong. A neurocomputational method for fully automated 3d dendritic spine detection and segmentation of medium-sized spiny neurons. *NeuroImage*, 50(4):1472–1484, May 2010.

[61] Yong Zhang, Xiaobo Zhou, Rochelle M. Witt, Bernardo L. Sabatini, Donald Adjeroh, and Stephen T.C. Wong. Automated spine detection using curvilinear structure detector and lda classifier. In *Proc. of Intl. Symposium on Biomedical Imaging*, pages 528–531, 2007.

[62] Wengang Zhou, Houqiang Li, and Xiaobo Zhou. 3d dendrite reconstruction and spine identification. In *Proc. of MICCAI*, pages 18–26, 2008.

[63] Warren R. Zipfel, Rebecca M. Williams, and Watt W. Webb. Nonlinear magic: multiphoton microscopy in the biosciences. *Nature Biotechnology*, 21(11):1369 – 1377, November 2003.