Visual computing techniques for the reconstruction and analysis of anatomically realistic neural networks

Vincent Jasper Dercksen
Zuse Institute Berlin

Dissertation
submitted to the
Department of Mathematics and Computer Science
at the Freie Universität Berlin
for the degree of
Doctor of Natural Sciences

Berlin, 2015

Referees:
Prof. Dr. Ch. Schütte (Freie Universität Berlin)
Prof. Dr. M. Hadwiger (King Abdullah University of Science and Technology)

Oral defense: December 11, 2015

Summary

To understand how the brain translates sensory input into behavior, one needs to identify, at the cellular level, the involved neural circuitry and the electrical signals it carries. This thesis describes methods and tools that enable neuroscientists to obtain important anatomical data, including neuron numbers and shapes, from 3D microscopy images. On this basis, tools have been developed to create and visually analyze anatomically realistic 3D models of neural networks:

1. An automatic segmentation method for determining the number and location of neuron cell bodies in 3D microscopy images. Application of this method yields a difference of merely ∼4% between automatically and manually counted cells, which is sufficiently accurate for application in large-scale counting experiments.

2. A method for the automatic alignment of 3D section volumes containing filamentous structures. To this end, an existing point-matching-based method has been adapted such that sections containing neuron and microtubule fragments could be successfully aligned.

3. The *Filament Editor*, a 3D proof-editing tool for visual verification and correction of automatically traced filaments. The usefulness of the Filament Editor is demonstrated by applying it in a validated neuron reconstruction pipeline to create 3D models of long-range and complex neuronal branches.

4. The tool *NeuroNet*, which is used to assemble an anatomical model of a neural network representing the rat *barrel cortex* (or subnetworks therein, e.g. individual cortical columns), based on reconstructed anatomical data, such as neuron distributions and 3D morphologies. The tool estimates synaptic connectivity between neurons based on structural overlap between axons and dendrites.

5. A framework for the interactive visual analysis of synaptic connectivity in such networks at multiple scales. It works from the level of neuron populations down to individual synapse positions on dendritic trees. It comprises the *Cortical Column Connectivity Viewer*, developed to analyze synaptic connections between neuron populations within and between cortical columns.

The usefulness of these methods is demonstrated by applying them to reconstruct and analyze neural networks in the rat *barrel cortex*. Finally, I describe several applications of these methods and tools by neuroscientists, yielding significant biological findings regarding neuron anatomy and connectivity.

Zusammenfassung

Um zu verstehen wie das Gehirn Sinnesreize, die zu bestimmtem Verhalten führen, verarbeitet, muss man in zellulärer Auflösung die beteiligten neuronalen Schaltkreise und die von ihnen übertragenen elektrischen Signale identifizieren. Diese Dissertation präsentiert Methoden und Werkzeuge, die es Neurowissenschaftlern ermöglichen, wichtige anatomische Daten wie Neuronenanzahl und -form aus 3D-Mikroskopaufnahmen zu rekonstruieren. Darauf aufbauend wurden Werkzeuge entwickelt, um anatomisch realistische Modelle neuronaler Netzwerke zu bilden und visuell zu analysieren:

1. Eine Methode zur automatischen Erkennung und Lokalisierung von Nervenzellkörpern in 3D-Mikroskopbildern. Die Anwendung dieser Methode ergibt eine Differenz von lediglich ∼4% zwischen automatisch und manuell gezählten Zellen. Diese Genauigkeit ermöglicht die automatische Verarbeitung von großen Datenvolumen.

2. Eine Methode zur automatischen Ausrichtung von 3D-Schnittvolumen, die linienartige Strukturen enthalten. Dazu wurde eine auf *Point-Matching* basierende Methode so adaptiert, dass Neuronen- und Mikrotubulifragmente erfolgreich ausgerichtet werden können.

3. Der *Filament Editor*, ein Werkzeug für die visuelle Validierung und interaktive Korrektur von automatisch segmentierten linienartigen Strukturen. Der Nutzen des Filament Editors wird durch Anwendung zur 3D-Rekonstruktion von langen und komplexen axonalen Verzweigungen eines Neurons gezeigt.

4. Das Tool *NeuroNet*, mit dem anatomisch realistische dreidimensionale Modelle eines neuronalen Netzes aus rekonstruierten anatomischen Daten zusammengesetzt werden können. Dieses Werkzeug schätzt die synaptische Konnektivität zwischen Neuronen basierend auf räumlicher Überlagerung von Dendriten und Axonen.

5. Ein interaktives System zur visuellen, skalenübergreifenden Analyse synaptischer Konnektivität in 3D-Modellen neuronaler Netze. Der darin enthaltene *Cortical Column Connectivity Viewer* ermöglicht die Analyse synaptischer Verbindungen zwischen Neuronenpopulationen innerhalb und zwischen kortikalen Säulen.

Die entwickelten Methoden werden anhand des Nachbaus eines Netzwerkes im *Barrel Cortex* der Ratte, in dem Sinnesinformationen der Barthaare verarbeitet werden, demonstriert. Abschließend werden einige Studien vorgestellt, in denen Neurowissenschaftler die beschriebenen Methoden angewendet haben und dadurch wesentliche neurobiologische Erkenntnisse gewinnen konnten.

Publications

The following list contains all peer-reviewed publications of which I am main author, equally contributing second author, or co-author. This thesis is based on the publications marked with a star (⋆). Other publications have been cited in this work (•) or are unrelated (◦).

Main author

⋆ VJ Dercksen, HC Hege, and M Oberlaender (2014). "The Filament Editor: an interactive software environment for visualization, proof-editing and analysis of 3D neuron morphology". *Neuroinformatics*, 12(2), pp. 325–339

⋆ VJ Dercksen, M Oberlaender, B Sakmann, and HC Hege (2012). "Interactive visualization – a key prerequisite for reconstruction of anatomically realistic neural networks". In: *Visualization in Medicine and Life Sciences II*. ed. by L Linsen, H Hagen, B Hamann, and HC Hege. Springer-Verlag Berlin, pp. 27–44

⋆ VJ Dercksen, R Egger, HC Hege, and M Oberlaender (2012). "Synaptic connectivity in anatomically realistic neural networks: modeling and visual analysis". In: *Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)*. Norrköping, Sweden, pp. 17–24

⋆ VJ Dercksen, B Weber, D Günther, M Oberlaender, S Prohaska, and HC Hege (2009). "Automatic alignment of stacks of filament data". In: *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Boston, MA, pp. 971–974

• VJ Dercksen, C Brüß, D Stalling, S Gubatz, U Seiffert, and HC Hege (2008). "Towards automatic generation of 3D models of biological objects based on serial sections". In: *Visualization in Medicine and Life Sciences*. Ed. by L. Linsen, H. Hagen, and B. Hamann. Springer-Verlag Berlin Heidelberg, pp. 3–25

◦ VJ Dercksen, S Prohaska, and HC Hege (2005). "Fast Cross-sectional Display of Large Data Sets". In: *IAPR Conference on Machine Vision Applications*. Tsukuba, Japan, pp. 336–339

Equally contributing second author

⋆ M Oberlaender, VJ Dercksen, R Egger, M Gensel, B Sakmann, and HC Hege (2009). "Automated three-dimensional detection and counting of neuron somata". *Journal of Neuroscience Methods*, 180(1), pp. 147–160
Contribution of main authors: MO: method (binary segmentation, model-based splitting), evaluation, manuscript. VJD: method (watershed- and model-based splitting), manuscript.

⋆ R Egger, VJ Dercksen, D Udvary, HC Hege, and M Oberlaender (2014). "Generation of dense statistical connectomes from sparse morphological data". *Frontiers in Neuroanatomy*, 8(129)
Contribution of main authors: RE: concept, evaluation, manuscript. VJD: implementation, computations, manuscript.

Co-author

⋆ R Egger, VJ Dercksen, CPJ De Kock, and M Oberlaender (2014). "Reverse engineering the 3D structure and sensory-evoked signal flow of the rat vibrissal cortex". In: *The Computing Dendrite*. Ed. by H Cuntz, MWH Remme, and B Torben-Nielsen. Springer Series in Computational Neuroscience. Springer, pp. 127–145

• H Pfister, V Kaynig, CP Botha, S Bruckner, VJ Dercksen, HC Hege, and JBTM Roerdink (2014). "Visualization in connectomics". In: *Scientific Visualization - Uncertainty, Multifield, Biomedical, and Scalable Visualization*. Ed. by CD Hansen, M Chen, CR Johnson, AE Kaufman, and H Hagen. Springer, pp. 221–245

⋆ M Oberlaender, CPJ De Kock, RM Bruno, A Ramirez, HS Meyer, VJ Dercksen, M Helmstaedter, and B Sakmann (2012). "Cell type-specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissal cortex". *Cerebral Cortex*, 22(10), pp. 2375–2391

⋆ S Lang, VJ Dercksen, B Sakmann, and M Oberlaender (2011). "Simulation of signal flow in three-dimensional reconstructions of an anatomically realistic neuronal network in rat vibrissal cortex". *Neural Networks*, 24(9), pp. 998–1011

• A Kuß, M Gensel, B Meyer, VJ Dercksen, and S Prohaska (2010). "Effective techniques to visualize filament-surface relationships". *Computer Graphics Forum*, 29(3), pp. 1003–1012

◦ U Homberg, R Binner, S Prohaska, VJ Dercksen, A Kuß, and U Kalbe (2009). "Determining Geometric Grain Structure from X-Ray Micro-Tomograms of Gradated Soil". In: *Workshop Internal Erosion*. Vol. 21, pp. 37–52

- C Papazov, VJ Dercksen, H Lamecker, and HC Hege (2008). "Visualizing morphogenesis and growth by temporal interpolation of surface-based 3D atlases". In: *Proceedings of the 2008 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Paris, France, pp. 824–827

- PW De Bruin, VJ Dercksen, FH Post, AM Vossepoel, GJ Streekstra, and FM Vos (2005). "Interactive 3D segmentation using connected orthogonal contours". *Computers in Biology and Medicine*, 35(4), pp. 329–346

Acknowledgements

The work described in this thesis has been carried out at the department of Visualization and Data Analysis at the Zuse Institute Berlin (ZIB).

First of all I would like to thank Prof. Dr. Ch. Schütte and my supervisor and head of our department Hans-Christian Hege for the opportunity to graduate in the exciting and innovative field of scientific visualization.

I also would like to thank all members of the Visualization department at ZIB for many fruitful and inspiring discussions (in the coffee kitchen, on our daily way to the mensa, or elsewhere). The undergraduate students I supervised, Maria Gensel, Jonas Hörsch, and Marianne Krabi, helped me with software development and trying out new ideas. Steffen Prohaska's incredible technical and organizational skills helped me to continuously improve my way of working and my scientific output.

This work emerged in cooperation with the Max Planck Institute for Medical Research, Max Planck Florida Institute, Max Planck Institute of Neurobiology (Prof. B. Sakmann), and the Max Planck Institute for Biological Cybernetics (Dr. M. Oberländer, R. Egger). Their scientific and financial support made this work possible. Marcel and Robert, it has been a pleasure working with you all these years. Many people collaborated with us and contributed data and advice. I cannot list them all here, but the presented results would have been possible without them.

Finally, I would like to thank Antje for her love, continuous support and patience repeatedly waiting for "only one more paper, then I'll write up, really!", and my kids Jari and Emmi for many wonderful moments.

Thank you.

Contents

1 Introduction

1.1 Neurobiological motivation

1.1.1 Understanding neural information processing

The study of the brain provides inspiration for better treatment for brain injury or neurological disorders and the development of more sophisticated computers and algorithms (National Academy of Engineering, n.d.). To understand how the brain works, it is important to have a structural description of its neural elements and their connections, the *connectome* (Sporns et al., 2005), because structural organization of neural circuitry is an important determinant of brain function. Continuous advances in staining and imaging techniques have increased the accessibility of connectivity information in the brain, both structural and functional, and from the micro- to the macro-scale (Kleinfeld et al., 2011). Complementary development of image processing, data analysis and visualization techniques is however required to extract, represent, integrate and analyze connectome data (Pfister et al., 2014).

For example, one fundamental question in neuroscience is how the brain translates sensory information into behaviour. An initial step to answer this question is to identify the local and long-range microcircuits of interconnected neurons that mediate this translation (Fig. 1.1a). A precise structural description of the neurons comprising the circuit and their synaptic connectivity provides a basis for investigating functional properties, e.g., using numerical simulation (Helmstaedter et al., 2007).

1.1.2 Reverse engineering of neural microcircuits

To comprehensively describe neural microcircuits, ideally, one would use dense reconstruction of electron-microscopic images to identify all neuronal branches and their synaptic connections in a volume of interest (Bock et al., 2011; Briggman et al., 2011). However, this is currently only feasible for relatively small volumes (e.g., $0.008\,mm^3$ (Bock et al., 2011), $0.06\,mm^3$ (Briggman et al., 2011)).

For larger volumes, for example comprising a cortical column (see Section 1.1.3) having a volume of $\sim 0.2\,mm^3$ (R Egger et al., 2012; VC Wimmer et al., 2010), *reverse engineering* approaches for reconstructing circuit anatomy and synaptic wiring have been suggested (Helmstaedter et al., 2007; Markram, 2006). These statistical approaches for the reconstruction of an 'average' anatomical network aim to estimate the mean values and variability of morphological parameters for individual neurons, as well as anatomical and physiological parameters for connections between cells (Helmstaedter et al., 2007).

To this end, a sparsely sampled set of reconstructed neurons, representative of the entire population, is integrated into a common reference frame. These reconstructions are cloned and repositioned to obtain a network realization that satisfies a measured neuron density. Based on this explicit 3D network representation, the synaptic connectivity between (populations of) neurons can be estimated (Kozloski et al., 2008; Lang et al., 2011). The resulting anatomically realistic models of neural circuits yield insight into the structural organization and allow for numerical (*in silico*) computer simulations (Lang et al., 2011), which will help to unravel circuit functions and their underlying mechanistic principles, for example during sensory-evoked behaviors.

In this thesis, methods and software tools are described to obtain required anatomical data from microscopic images, particularly the spatial distribution and 3D shape of neurons, and to assemble 3D neural network models based on this information. The effectiveness of these methods and tools is demonstrated by applying them to create a model of the rat barrel cortex. The whisker-barrel system of the rat has several properties that make it a popular model system for neuroscientific research (DE Feldman and Brecht, 2005), particularly the one-to-one representation of individual whiskers by discrete functional and anatomical areas in the vibrissal cortex: *cortical barrel columns*. It has therefore been extensively studied, making it a suitable model system to pursue the reverse engineering approach.

1.1.3 Model system: the whisker-barrel system of the rat

Rats, like most nocturnal rodents, use their whisker hairs as complex tactile sensory organs to explore their environment. The tactile somatosensory pathway from whisker to the cortex in rodents is a convenient model system for studying neural processes, because it provides a well-defined system for exploring the link between molecular mechanisms, circuits of synaptically coupled cells, and behavior (CCH Petersen, 2007) (see Fig. 1.1b).

The whisker is anchored to the skin by the follicle. Mechanoreceptor neurons respond to rotation of the whisker follicle by its muscles or to deflection of the whisker shaft by external contacts. They convert mechanical energy into action potentials, encoding information about velocity, direction and duration of whisker displacements and torques. These nerve cells terminate in the trigeminal nucleus (TN) in the brainstem (Diamond et al., 2008).

Sensory information from the whiskers reaches the cortex via three different pathways (Diamond et al., 2008). In this work, we are primarily interested in the lemniscal pathway, which conveys tactile information to the primary somatosensory cortex (S1, also called barrel cortex) in a per-whisker segregated manner as follows.

Neurons in the principal TN are clustered into *barrelettes*: small, whisker-related cell groups. These brainstem neurons transmit whisker-related information to somatosensory regions of the contralateral thalamus. The major recipient of TN input is the medial division of the ventral posterior nucleus (VPM), which, like the TN and S1, contains groups of neurons, each related to the information from a single whisker. In VPM these groups are in the form of elongated ovoids, called *barreloids* (Oberlaender, 2009).

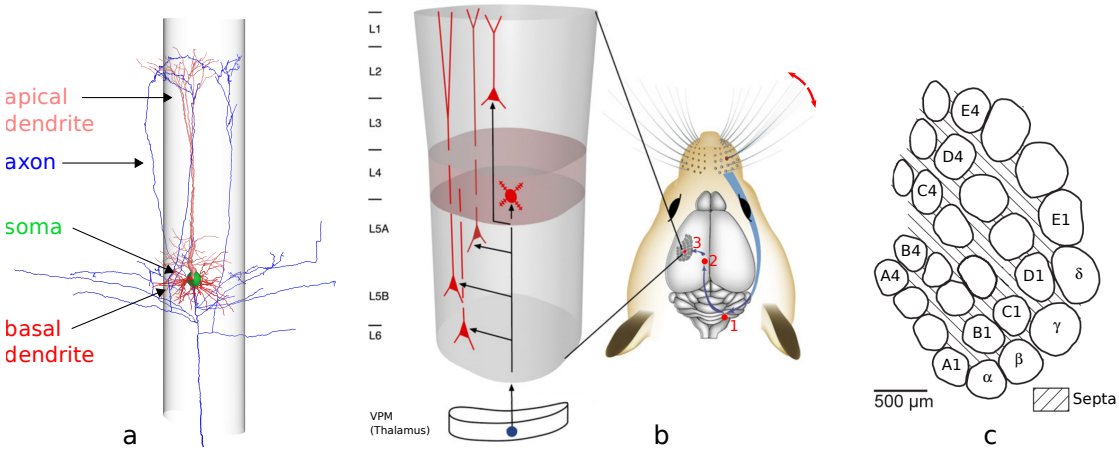

Figure 1.1: (a) 3D reconstruction of a Layer 5 thick-tufted neuron within an abstract cortical column, showing morphological substructures involved in signal processing: the cell body (soma, green), multiple dendritic branches (red, pink), and one axonal tree (blue). Chemical or electrical signals received by the dendrites are integrated at the soma, potentially resulting in an action potential that travels along the axons and results in output signals transferred to other nerve cells (Kandel et al., 2000) (Data: M. Oberlaender). (b) The whisker-barrel system of the rat. Information from individual whiskers is conveyed by the brain stem (1) to the thalamus (2) and finally to the primary somatosensory cortex (3). The cortical columns comprising S1 are innervated by axons from the VPM. Image modified from (Helmstaedter et al., 2007). (c) Tangential view onto the barrel field. The layout of the columns in S1 corresponds to the spatial organization of the whiskers on the rat's snout.

The axons of VPM neurons project to the primary somatosensory cortex (S1), where they terminate in *barrels*: discrete, well-defined anatomical structures consisting of dense clusters of small neurons in layer IV, in a whisker-specific manner (Woolsey and Van Der Loos, 1970). Thus, the barrelettes, barreloids and barrels form anatomical maps in which whiskers are represented individually, thereby retaining whisker segregation along the entire pathway. The somatotopic arrangement of the barrels is almost identical to the layout of the whiskers on the snout (see Fig. 1.1c).

The mammalian cortex, including the rat cortex, is organized into six layers (L1–L6, see Fig. 1.1b). A cortical barrel column can be defined as the vertical thickness of the neocortex, laterally bounded by the width of the L4 barrel (Lefort et al., 2009). The barrels are separated by cell-sparse zones, called septa (see Fig.1.1c). S1 receives input from, or transfers output towards other brain areas, such as the thalamus. Hence, many anatomical and functional studies of cortical columns in S1 revealed neural microcircuits between and within layers, from the thalamus to the cortex (thalamocortical) and vice versa (corticothalamic) (Oberlaender, 2009). The cortical (barrel) column is related in a one-to-one fashion to a single facial whisker hair on the animals snout. Functional

output from this network, based on single whisker information, is sufficient to trigger simple behaviors, such as decision making (Celikel and Sakmann, 2007).

The anatomical description of the whisker-barrel system above is rather qualitative. A realistic 3D anatomical model involving quantified numbers of neurons and their interconnections, could, together with measured and/or simulated functional data, yield new insights and understanding of principle mechanisms that explain how the brain translates environmental input into behavioral responses.

1.2 Problem formulation

The goal of this thesis is to present

> a complete set of methods and tools to create and analyze an anatomically realistic 3D model of neural networks in the rat barrel cortex, consisting of populations of neurons and their synaptic connectivity.

To this end, the following problems have to be solved:

Determination of the number and spatial distribution of neurons in the brain area of interest To model anatomically realistic neuron populations, the number and location of neurons must be known. Exhaustive counting is preferred over sampling and extrapolation, which could be inaccurate since neuron density varies across brain regions. Three-dimensional imaging techniques (e.g., confocal laser scanning microscopy) together with suitable stains, like NeuN (Mullen et al., 1992), labeling all neuron somata (cell bodies), provide images with sufficient resolution and contrast to identify individual somata, although they often appear as overlapping. All somata, appearing as roundish objects, need to be extracted automatically, to be able to quantify neuron densities in any non-trivial brain volume, such as a cortical column, which contains ∼18 thousand neurons (HS Meyer et al., 2010). From an informatics point of view, this amounts to the following problem:

P1 Given a 3D confocal laser scanning microscopy image containing roundish, possibly touching objects, identify the number and location of the center of mass of each object.

Reconstruction of the 3D axonal and dendritic morphology of a representative set of neurons of all occurring cell types Reconstruction of all neurons within any non-trivial brain volume, such as a cortical column, is currently not possible, neither using dense nor sparse labeling. Instead 3D reconstructions of a representative sample of neurons of all occuring cell types are to be generated, which can be duplicated to attain the measured number of neurons in the modeled population. A representative sample is also required to determine the location-specific mixture of cell types, which varies with cortical depth.

Because neuron function depends to a large extent on its shape, morphologies must be traced with high accuracy. At the same time, long and complex axonal arbor requires large areas to be imaged. Currently, these structures can only be reconstructed from individually stained neurons that have been physically sectioned and imaged using light microscopy techniques. To obtain a sufficiently large population sample in reasonable time, the reconstruction method must be relatively high-throughput. Although many automatic tracing methods exist (Donohue and GA Ascoli, 2011; Meijering, 2010; Oberlaender et al., 2007), different types of background, staining and other image properties have a degrading impact on their results, necessitating an efficient and effective proof-editing tool to visually validate tracings (Peng et al., 2011) and correct errors (Luisi et al., 2011). Further, tracings from adjacent sections must correctly be merged to attain complete reconstructions. Third, neuron reconstructions should be augmented by anatomical landmark structures, such as barrel, white matter or pia contours, to be used as features during registration in a reference frame (R Egger et al., 2012), or morphometric analysis. Finally, statistics on neurite lengths differentiated with respect to neuron substructure (axon, apical/basal dendrite) and containment by reference volumes (e.g. cortical columns) are of interest in order to characterize cell types and quantify axonal innervation patterns. Thus, from an informatics point of view, the following problems have to be solved:

P2 Given multiple 3D data sets containing filamentous structures as polylines, obtained from adjacent sections, find a 2D rigid transformation for each data set, such that filaments in neighboring sections match.

P3 Given multiple data sets containing a potentially large number (>1M vertices) of filamentous structures, traced in 3D images obtained from adjacent sections, provide a proof-editing tool with the following functionalities:

- data structure to represent the tracing as a graph embedded in 3D space that can be edited at interactive rate,
- effective visualization allowing comparison of a tracing with the original image data,
- efficient and effective interactive editing of the graph data structure,
- annotation of substructures for visualization and morphometric analysis,
- manual delineation of reference structure contours,
- morphometric analysis of branch length, with respect to substructure annotations and user-defined reference volumes.

Assembly of the anatomical neural network and computation of synaptic connectivity After collecting the anatomical data described above, these have to be combined into a 3D model representing a neural network in the brain area of interest. To this end, the data is first registered into a common reference frame, which in our case of the rat

vibrissal cortex is given by the geometry of the cortical columns, and the pia and white matter surfaces (R Egger et al., 2012). A software tool is required that generates a 3D neural network model, based on this data:

P4 Given a distribution of neuron somata, a representative set of 3D neuron reconstructions of all occurring cell types and location-specific cell type mixtures, registered in a common coordinate frame representing the brain area of interest, provide a tool

- to assemble an anatomically realistic network, satisfying the given anatomical constraints,
- to compute estimates of spatial distributions of synaptic contacts between (groups of) neurons, based on local axon-dendritic overlap,
- to compute realizations of synapse positions on the dendrites of postsynaptic neurons, satisfying the computed distribution.

Visual and quantitative analysis of synaptic connectivity at multiple scales The models generated by the tool above are explicit representations of an anatomically realistic neuron populations and are therefore amenable to visual and quantitative analysis of morphological and connectivity properties at multiple scales: from neuron populations down to the subcellular scale. The information contained in the model allows neuroscientists to answer questions like: Where does a neuron or group of neurons obtain input from and where does it project to? Are there quantitative differences in innervation strength by thalamic axons between individual cortical columns? How are synapses distributed on a postsynaptic cell? However, to be of general use, this intrinsic, quantitative information must be made accessible. Therefore a visualization tool is required that allows the user to explore data, and supports a drill-down workflow to answer increasingly specific questions:

P5 Given an explicit representation of a 3D neural network, consisting of a neuron population and an algorithm to compute number of the synaptic contacts between (groups of) neurons as well as the synapse locations on individual neurons, provide an interactive visualization tool that allows the user

- to efficiently select (groups of) pre- and postsynaptic neurons, based on cell type and cortical column,
- to visualize and quantify the connection strength between these selected (groups of) neurons,
- to visualize the predicted distribution of synapses on a selected postsynaptic cell.

1.3 Contributions

This thesis presents new methods and tools in the field of image segmentation, 3D geometric modeling and interactive visualization to solve the problems mentioned above. In particular, this thesis presents the following scientific contributions:

1. An automatic method to detect and count neuron somata in 3D images (Oberlaender et al., 2009). The presented method segments these roundish objects, paying specific attention to correctly split touching objects.

2. The Filament Editor: an interactive proof-editor for validating, correcting, aligning, merging, annotating, and analyzing tracings of filamentous structures obtained from multiple brain sections (Dercksen et al., 2014). The Filament Editor is a key element in a workflow for creating high-resolution 3D neuron reconstructions, including long and complex axonal arborizations, from sparsely labeled image stacks.

3. An automatic method for the 2D rigid alignment of 3D sections containing filamentous structures, such as neuronal arborizations (Dercksen et al., 2009). The alignment method builds on an existing point-matching approach (Baum, 2007) to quickly align boundary points in neighboring sections.

4. A tool to create an anatomical model of a neural network representing the rat barrel cortex, based on reconstructed anatomical data, such as neuron distributions and 3D morphologies (Dercksen et al., 2012). The tool estimates synaptic connectivity between neurons from structural overlap between axons and dendrites.

5. A framework for the interactive visual analysis of synaptic connectivity at multiple scales, i.e. from the level of neuron populations down to individual synapse positions on dendritic trees (Dercksen et al., 2012). An important part of the framework is the The Cortical Column Connectivity Viewer, a visual analysis tool to analyze synaptic connections between neuron populations within and between cortical columns.

Finally, the effectiveness, usefulness or even indispensableness (Dercksen et al., 2012) of the developed methods and tools is illustrated by reviewing how they have been applied to answer neuroscientific questions regarding anatomy (HS Meyer et al., 2010; Oberlaender et al., 2011; Oberlaender et al., 2012), synaptic connectivity (R Egger et al., 2014; Oberlaender et al., 2012), and, using numerical simulation of electrical activity, structure-function relationships (R Egger et al., 2014; Lang et al., 2011) in neural networks in the rat barrel cortex.

2 Determination of the number of neurons and their spatial distribution

2.1 Introduction

The number of neurons in the brain and their varying density between different brain regions is thought to be a fundamental determinant of brain function (Donaldson, 1895; Williams and Herrup, 1988; Williams and Rakic, 1988). During the last 60 years, great effort has been made to estimate neuron densities quantitatively, first addressed by Abercrombies article "Estimation of Nuclear Population from Microtome Sections" (Abercrombie, 1946). The estimation of absolute numbers of neurons, densities or rates of density change in neuron populations used to be based on random, sparse sampling methods (Cragg, 1967; Rockel et al., 1980) such as stereology (Sterio, 1984). These methods determine cell densities by inspecting a representative sub-volume of tissue and extrapolating the obtained density values to a reference volume. However, for more accurate numbers and quantification of density changes across a large volume, e.g. $\sim 0.5\, mm^3$ for a cortical column in the rat barrel cortex (HS Meyer et al., 2010), it would be favorable to count the absolute number of neurons and derive the detailed three-dimensional neuron distribution of the brain area of interest.

Recently available three-dimensional imaging techniques (mosaic/optical-sectioning confocal laser scanning, 2-photon or widefield microscopy) and suitable neuronal stains opened new possibilities for the determination of neuronal densities within entire volumes. Neuronal stains, like NeuN (Mullen et al., 1992) labeling all neuron somata (cell bodies), or GAD67 (Kaufman et al., 1986) labeling GABAergic interneuron somata, as well as genetically encoded labels of specific neuron populations in transgenic mice or drosophila (Akemann et al., 2004; L Luo et al., 2008) allow in principle the quantitative determination of density differences between neuron populations at high level of detail (e.g. between or within cortical layers).

Imaging these stained specimen results in 3D data sets with the somata appearing as roundish objects of approximately equal size, sometimes prolongated by the onset of the dendritic arbors that have taken up some stain. The somata may touch; the contact area may range from very small, resulting in a thin "neck" between the somata, to very large, making soma boundaries hard to distinguish.

The problem is to determine the exact number of somata in the 3D image and their approximate centroid position. Clustered somata therefore need to be separated to avoid systematic underestimation.

Here, we present a method to automatically detect and count NeuN-stained neuron somata in 3D confocal images. The method consists of 3 steps. First, the somata are separated from the background in a binary segmentation step. Second, clustered somata connected by a thin neck are separated using watershed-based morphological operators. Third, remaining clusters are split based on a statistical model of soma volumes. The first step is stain/microscope-dependent. We describe the method for NeuN-stained somata imaged with a confocal microscope. This step can be slightly adapted for other stain/microscope combinations (Oberlaender et al., 2009), after which the remaining steps can be applied unaltered.

The method is applied to several data sets obtained from the rat somato-sensory cortex. Comparison of the automatically determined soma positions to manually placed counterparts yields that the deviation in landmark position is negligible and that the difference between the numbers of manually and automatically counted neurons is less than 4%. In consequence, this novel approach for neuron counting is a reliable and objective alternative to manual detection.

2.2 Related work

The simplest way to detect all somata in such 3D images is by manual marking. This is, for example, achieved by moving a 2D image slice through the data set and clicking on the center of each soma to place a landmark at its approximate centroid. The center is set on the slice where the soma has the largest diameter. Although this procedure is quite accurate, with a reported inter-user variability of 2% (HS Meyer et al., 2010), it is very time-consuming. For the quantification of larger brain volumes, e.g. a cortical column containing many thousands of neurons, an automatic method is required.

There is a large body of literature on the automatic segmentation of near-round objects (blobs) in 2D or 3D images. The objects of interest in these images are frequently entire cells or sub-cellular structures, like nuclei or neuron cell bodies. Comparison of these segmentation methods yields a number of basic processing steps that by themself or in combination constitute the individual methods to achieve the segmentation goal. These central processing steps are described below.

Image enhancement is often required to compensate for image characteristics that complicate the analysis, for example limited signal-to-noise ratio (SNR), or non-uniform illumination in x/y and/or z-direction. These characteristics are often specific to the image acquisition technique. Segmentation methods therefore often incorporate acquisition-type-dependent techniques to counter these effects. These techniques typically comprise one or more image filters, for example for smoothing (Long et al., 2007), denoising (Kharma et al., 2007; G Lin et al., 2003), shading correction (G Lin et al., 2003; Malpica et al., 1997) or filling unstained interiors (Long et al., 2007).

Foreground extraction. For foreground extraction both bottom-up techniques (delineating objects based on pixel, edge or region information), and top-down model-based methods (using *a priori* knowledge about the expected shape) are employed, as well

as combinations of both. Bottom-up, intensity-based approaches include edge detection (Liu et al., 2008; Raman et al., 2007) and thresholding (Jung et al., 2010; Kharma et al., 2007; Long et al., 2007; Malpica et al., 1997; Wu et al., 2000). Top-down, model-based approaches include active contours (Cheng and Rajapakse, 2009; Fatakdawala et al., 2010), level set segmentation (H Chang and Parvin, 2006), template matching (Byun et al., 2006) and graph-cut algorithms (Daněk et al., 2009; Al-Kofahi et al., 2010). In the resulting image the background is usually set to black; the foreground is either set to white or the original gray-value is retained.

Marker/seed detection. The goal of seed detection is to find a single small region for each individual object that marks its approximate location. The seed position is used to initialize some method that finds the entire region comprising the object. The number of seeds therefore equals the number of segmented objects. Seed/marker detection can be done on a gray-value image (either the entire image or the gray-value foreground) or on the binary foreground. Methods for seed detection in gray-value images include voting along the image gradient (H Chang and Parvin, 2006; Parvin et al., 2007; Qi et al., 2012) and peak response detection of a Laplacian-of-Gaussian filter (Byun et al., 2006; Al-Kofahi et al., 2010). In binary images often the regional maxima in the distance transform of the foreground are used as markers (Vincent, 1993). To remove spurious maxima, which would result in over-splitting, often the H-maxima transform is applied (Cheng and Rajapakse, 2009; Jung et al., 2010; Malpica et al., 1997).

Cluster splitting. Several approaches can be identified to split clustered objects. Geometrical criteria have been used to split clustered objects in the foreground of 2D images, e.g. by fitting ellipses to the foreground (Kharma et al., 2007), or by connecting contour concavities (Fatakdawala et al., 2010; Raman et al., 2007). Alternatively, a watershed-based method can be used, e.g. on the distance transform (Long et al., 2007) or the gradient-weighted distance transform of the foreground (G Lin et al., 2003). However, this frequently results in oversegmentation, and subsequent region merging is therefore required. Criteria for deciding which regions to merge include convexity (Long et al., 2007) or the fit to a model shape (G Lin et al., 2003; G Lin et al., 2005) or one of multiple models, thereby classifying the shape (G Lin et al., 2007). Finally, clusters can be split by computing the corresponding regions of the markers found using any of the techniques mentioned above. Methods to achieve this include marker-based watershed segmentation (Cheng and Rajapakse, 2009; Jung and C Kim, 2010; Malpica et al., 1997), level sets (H Chang and Parvin, 2006; Qi et al., 2012), fitting a Gaussian Mixture Model (Jung et al., 2010) or a graph-cut algorithm (Daněk et al., 2009; Al-Kofahi et al., 2010).

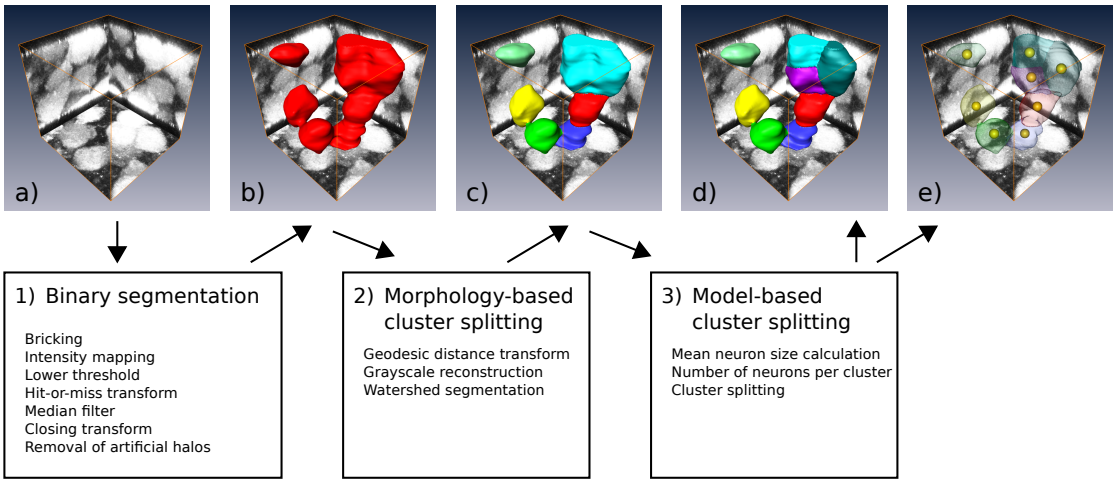

Figure 2.1: Summary of the neuron somata segmentation pipeline. The pipeline is illustrated using a small volume of a confocal stack containing NeuN stained somata. (a) Maximum intensity projection of the volume. (b) First, a sequence of image filters, including a threshold-based segmentation produces a binary image, illustrated by the isosurface. (c) In a second step, objects connected by narrow links are separated using a marker-based watershed algorithm. (d) In the last step, any remaining clusters are split into the most likely number of somata, based on a statistical model of soma volume. (e) The centroids of the resulting regions are the final soma positions.

2.3 Method for automatic detection of neuron somata in 3D images

2.3.1 Method overview

Here we present an automated 3D neuron counting method that combines a thresholding, watershed and model-based approach into a novel high-throughput system for detection of neuron somata (alternatively called neurons throughout this chapter) in confocal images. The processing pipeline consists of three steps and is summarized in Figure 2.1.

The goal of the first threshold-based step is to create a binary image separating foreground (i.e. stained neurons) from background. It consists of a number of image processing steps, including compensation for imaging or staining artifacts such as bleaching, shading or uneven uptake of the stain, and binarization by local thresholding. This step is data-specific and tailored towards confocal image stacks of NeuN-labeled neurons. Oberlaender et al. (Oberlaender et al., 2009) describe how this step can be adapted for images acquired with other stain/microscope combinations, in particular neurons stained with Ca^{2+}-sensitive dye images imaged with two-photon microscopy and NeuN-stained neurons imaged with widefield microscopy.

This threshold-based approach is usually not sufficient to detect the true number and position of neurons. High neuron densities and limited microscope resolution result

in clusters of neurons that cannot be separated by the local threshold step. The first processing step is therefore regarded as a pre-processing step that guarantees a similar input to the second (watershed-based) and third (model-based) processing steps. The implementation of the latter two steps is independent of the data type.

In the second watershed-based step, clusters of neurons which are connected by narrow links are separated by a morphological filtering process, resulting in an image of distinct watershed regions (3D objects of connected foreground voxels, identified by a label number), and ideally representing individual neurons. Some clustered neurons appear however like a single, large and uniformly stained neuron. The morphological filters are not capable of splitting such clusters into distinct watershed objects.

The third, model-based processing step addresses this problem. We assume a single dominant neuron population within the image stacks with a Gaussian-distributed soma volume. The mean neuron volume and its variance are calculated from a volume histogram of the watershed regions. Undivided clusters are then split according to their volume, assuming that it has to be an integer multiple of the mean soma volume. An additional advantage of this constraint is that its parameters are not specified by the user but automatically calculated during the image processing.

2.3.2 Threshold-based filtering (pre-processing)

Fluorescent images can suffer from two kinds of artifacts. First, shading or bleaching of the stain leads to an uneven illumination across the images and is usually caused by the image acquisition itself. It results in different signal-to-noise ratios (SNRs) across and between individual image planes. The second artifact is caused by uneven uptake of the fluorescent dye, resulting in varying intensity values across individual neurons. The first issue is addressed by subdividing each image plane into rectangular bricks and processing each image plane individually as will be described in Section 2.3.2.2/2.3.2.3. The second issue is addressed by processing each three-dimensional object (e.g. neuron soma) individually as will be described in Section 2.3.2.7.

2.3.2.1 Bricking

Uneven illumination across individual image planes prohibits the application of global image operators. Further, illumination deviations between individual image planes limit a general application of 3D image operators. Each image plane (i.e. one field of view; 1024×1024 pixels) is hence subdivided into 2D bricks. In the case of the confocal microscope at $40\times$ magnification the bricks size is 256×256 pixels. This size is linearly adjusted for different magnifications (or pixel resolutions), in order to guarantee a similar brick area (in mm^2) for all kinds of images.

2.3.2.2 Intensity mapping

Direct application of a lower threshold operator, setting all pixels having a value smaller than the threshold value to zero, to each 2D brick proved to be problematic. The

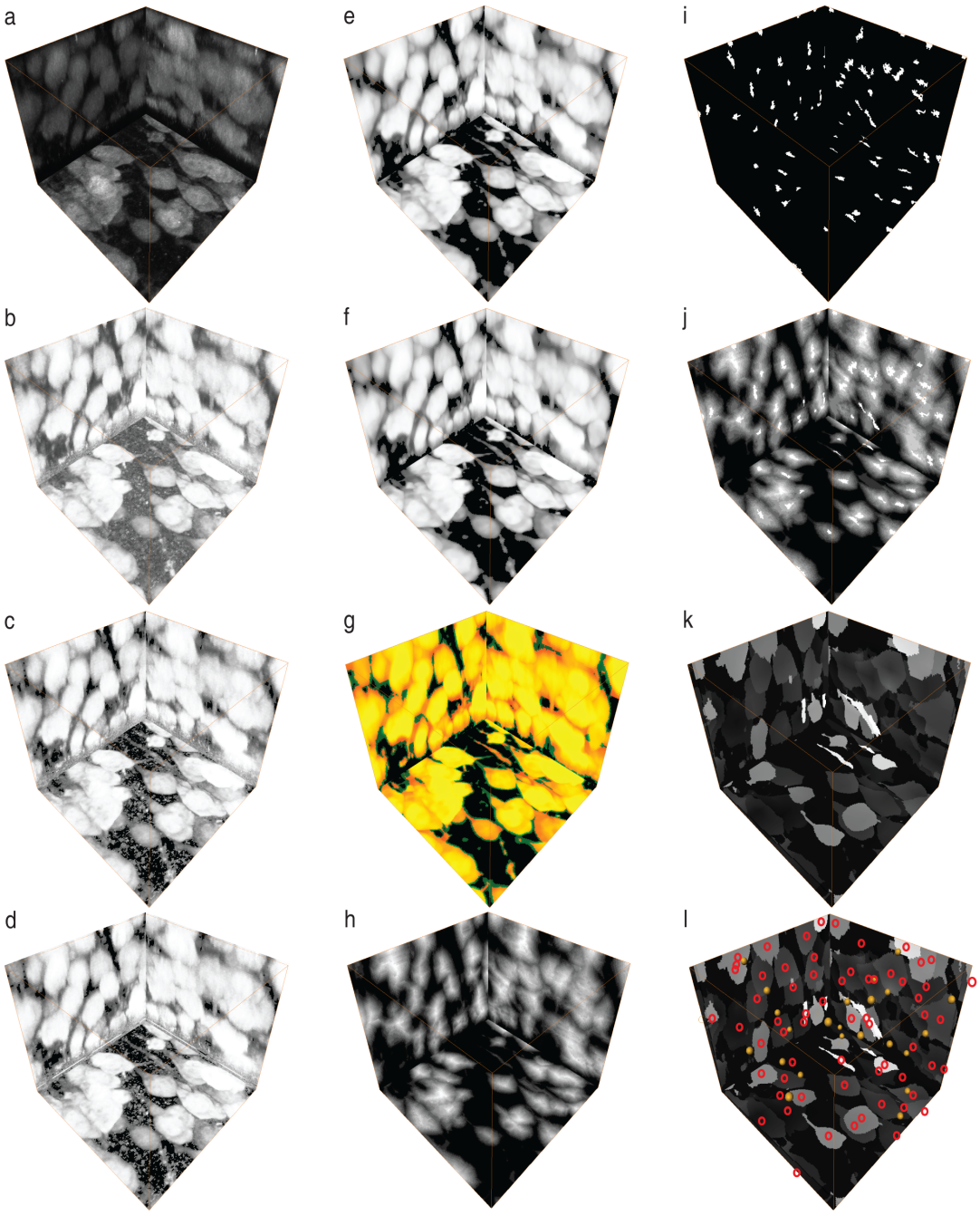

Figure 2.2: Overview of the processing pipeline. Each panel shows x/y-, x/z- and y/z-projections of a $80 \times 80 \times 80\,\mu m^3$ sub-volume of a confocal image containing NeuN-stained somata. (a) Original stack, (b) intensity mapping, (c) lower threshold, (d) hit-or-miss transform, (e) median filter, (f) closing filter, (g) removal of artificial halos (green), (h) Euclidean distance transform, (i) regional maxima (markers), (j) modified distance field to be flooded by the watershed algorithm, (k) marker-driven watershed segmentation, (l) model-based splitting: soma positions (yellow) and their projections (red).

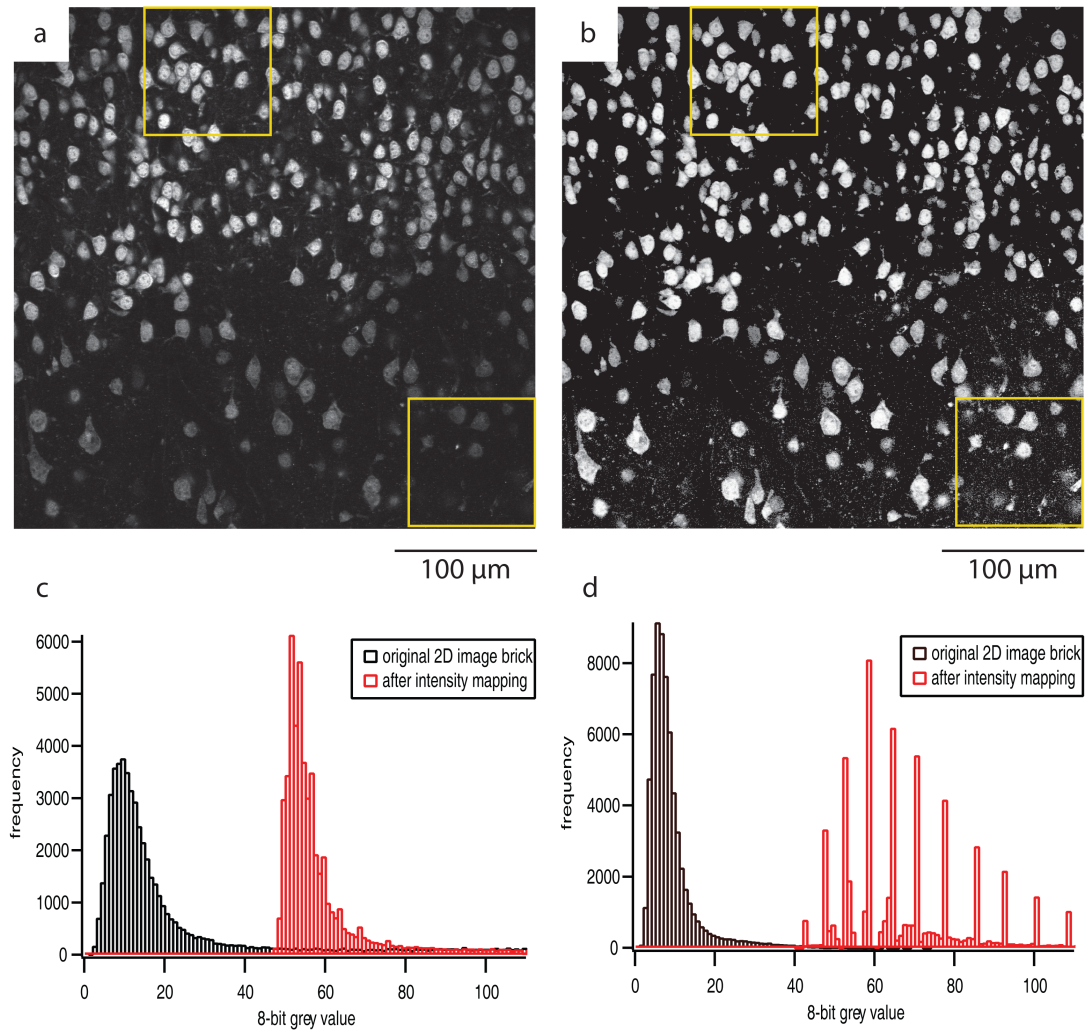

Figure 2.3: Illustration of local intensity mapping and local threshold function. (a) Single x/y image plane affected by uneven illumination. The high-contrast in the upper brick results in a broad gray-value histogram (c, standard deviation σ of black histogram $= 6.22$), whereas the low-contrast bottom brick has a narrower histogram (d, black, $\sigma = 3.58$). (b) Image plane after local intensity mapping. The neurons across the entire plane are of similar intensities. The noise in the prior low-contrast brick is high, resulting in a much broader histogram (d, red, $\sigma = 56.52$) whereas the width of the histogram of the high-contrast brick remains more or less unchanged (c, red, $\sigma = 4.36$). (c) The uneven illumination results in different histogram width after intensity mapping. In bricks with high contrast, the intensity mapping leads to amplification of the somata and attenuation of the background. Hence the lower threshold should be approximately the mean value of the mapped image. (d) In low-contrast images, the structures and a significant part of the background are amplified, resulting in a broad histogram. Hence the lower threshold needs to be higher than the mean value after mapping, which is lowered by amplified background. This is realized by the threshold function defined in Eq. 2.3. For high-contrast images t_1 is essentially μ_{new} (here: $\mu_{new} + 1$), whereas in the low-contrast case t_1 is significantly increased (here: $\mu_{new} + 17$).

SNR is usually too low to set an adequate threshold value that separates somata from background. Hence the intensities for each brick are mapped by a non-linear sigmoid-shaped filter (Eq. 2.1) (Ibanez et al., 2005) onto a new range:

$$I' = \frac{255}{1 + e^{-\frac{(I-\beta)}{\alpha}}}, \tag{2.1}$$

where I' and I denote the new and old intensity (gray) values, respectively. β ideally represents the center and α the width of the neuron's intensity range. This filter progressively attenuates intensity values outside this range and produces a very smooth and continuous transition to the specific intensity range of interest (Fig. 2.2b). It results in a per-brick amplification of the neurons with respect to their surroundings (Fig. 2.3a/b). Systematic testing yielded an intensity range of neurons that is best described by the following values for β and α:

$$\beta = \mu_{original} + 0.75\,\sigma_{original}, \quad \alpha = \sigma_{original} \tag{2.2}$$

where $\mu_{original}$ and $\sigma_{original}$ refer to the mean gray value and standard deviation of each 2D image brick, respectively.

2.3.2.3 Lower threshold

Once the neurons are amplified with respect to their surrounding, a lower threshold t_1 is applied in order to separate the neuron somata from background:

$$t_1 = \mu_{new} + 1.1 \cdot \frac{\sigma_{new}}{\sigma_{original}} \tag{2.3}$$

where μ_{new} and σ_{new} refer to the mean gray value and standard deviation of each 2D image brick after intensity mapping. This thresholding step sets all voxels below t_1 to background, i.e. to value zero. The local threshold function comprises a term inversely proportional to the standard deviation of the image brick prior the intensity mapping. This accounts for possible uneven illumination within each image plane. The width of the intensity distribution ($\sigma_{original}$) can vary significantly from one brick to the next. Two extreme situations occur when neurons are surrounded by low background, resulting in broad intensity distributions (Fig. 2.3a, top brick/Fig. 2.3c), and when neurons are surrounded by background values similar to the neurons' intensities, resulting in a narrow distribution (Fig. 2.3a, bottom brick/Fig. 2.3d). In the latter case, the width of the intensity distribution after mapping will be much larger than the original one ($\sigma_{new} \gg \sigma_{original}$), resulting in an approximate threshold of $t_1 \approx \mu_{new} + \sigma_{new}$ (Fig. 2.3d). In the other case, the two widths will be of similar order of magnitude, resulting in a threshold of $t_1 \approx \mu_{new}$ (Fig. 2.3c). Thus, this filter is capable of discriminating neurons surrounded by high and low background values (Figs. 2.2c and 2.3b).

2.3.2.4 Hit-or-miss transform

The two previous filtering steps result in image stacks of significantly reduced background. However, small speckle artifacts are usually still present. The image planes are therefore subjected to a hit-or-miss transformation with rectangular frame masks of increasing size as structuring elements (Gonzalez and Woods, 2002; Oberlaender et al., 2007). The transformation is applied to every image plane. Isolated foreground objects that are completely surrounded by a frame are converted to background (Fig. 2.2d). Beginning with a radius of one pixel and increasing the frame size subsequently to three pixels, small and isolated artifacts are removed.

2.3.2.5 Median filtering

In order to smooth the intensity distribution within neurons, a median filter is applied as implemented by the ITK (Ibanez et al., 2005). Each voxel is assigned a new intensity value that is the median value of its surrounding voxels (Fig. 2.2e). A neighborhood size of $5 \times 5 \times 5$ voxels was chosen. This filter is computed in three dimensions because neurons are 3D objects, which consist of 2D planes that may vary systematically in gray values. 2D median filters would not decrease these inter-plane deviations.

2.3.2.6 Closing transform

Next, a grayscale closing filter (Gonzalez and Woods, 2002) is applied as implemented by the ITK (Ibanez et al., 2005). Its geometrical interpretation is that a "sphere" rolls along the outside boundary of a foreground object (i.e. neuron soma). It tends to smooth contours, fuses narrow breaks, eliminates small holes, and fills small gaps in the neurons (Fig. 2.2f). The 3D structuring element (sphere) has a radius of five voxels (approximately 2/3 of a soma radius).

2.3.2.7 Removal of artificial halos

The uneven uptake of stain results in neurons of weak intensities in intermediate neighborhood to neurons with high intensity values. The intensity mapping described above causes amplification of the surrounding of such weakly pronounced neurons and sometimes fuses them with other neurons (halos (Fig. 2.2g, green)). By removing "halos" that were introduced by the intensity mapping, the neurons are cropped to their original volume (Fig. 2.2g, yellow/orange). This is realized by processing each 3D object of connected foreground pixels individually. For each 2D plane of each object, a lower threshold value t_2 is calculated, defining the foreground for this object plane:

$$t_2 = \mu_{filter} - 1.2 \cdot \sigma_{filter} + \frac{40}{\sigma_{original}} \tag{2.4}$$

where μ_{filter} and σ_{filter} refer to the mean gray value and standard deviation of each 2D object plane after the closing filter. The local threshold function comprises a term

inversely proportional to the intensity deviation of the 2D object plane $\sigma_{original}$ in the unprocessed image, again compensating for uneven illumination between the bricks as described in Section 2.3.2.3. Bricks with weakly pronounced objects have a relatively small $\sigma_{original}$, resulting in a larger threshold value. Thus, an outer layer of higher-intensity voxels is removed from such objects, while retaining it for objects in high-contrast bricks. The parameters for the threshold function are obtained by systematic testing. This is the final result of the first (pre-processing) step.

2.3.3 Watershed-based splitting of object clusters

The filtering described in Section 2.3.2 results in a stack of 2D gray-value images, which is then transformed into a binary stack by setting all pixels with a gray value larger than 0 to 255. In the following such stacks will be considered as single 3D binary images. In order to find the total number of neurons in the image, one could simply count the total number of 3D connected foreground objects (groups of connected voxels) in the image. However, the limited resolution of light microscopy imaging systems in addition to high neuron densities results in clusters of neurons that cannot be separated by the pre-processing pipeline. In consequence, direct counting produces total neuron numbers which are generally too low, because a single foreground object may consist of multiple connected neurons. We therefore divide such clusters into their constituent neurons, using a method described by Vincent and Dougherty (Vincent and Dougherty, 1994). This method consists of three steps:

1. Computation of a distance transform for each 3D foreground object of the binary image.

2. Finding exactly one marker for each neuron (i.e. multiple markers for neuron clusters), where marker refers to a single voxel or a group of connected voxels.

3. Computation of a watershed transformation, using the markers as initial basins.

2.3.3.1 Geodesic distance transform

First, a distance transform (Gonzalez and Woods, 2002) is computed for each 3D foreground object in the binary image. This results in intensity values for each voxel that resemble the physical Euclidean distance to the closest background voxel. Thus, voxels in the interior of objects have high values; object voxels close to the boundary have low values; and background voxels have zero value (Fig. 2.2h). These values are computed by repeatedly (binary) eroding (Gonzalez and Woods, 2002; Ibanez et al., 2005) the objects in the pre-processed image, successively peeling their outer boundaries. During each erosion step and for each foreground object, the physical Euclidean distance to the prior erosion level is assigned to voxels of the current outermost layer. A $3 \times 3 \times 3$ voxel binary cross is used as the structuring element for erosion.

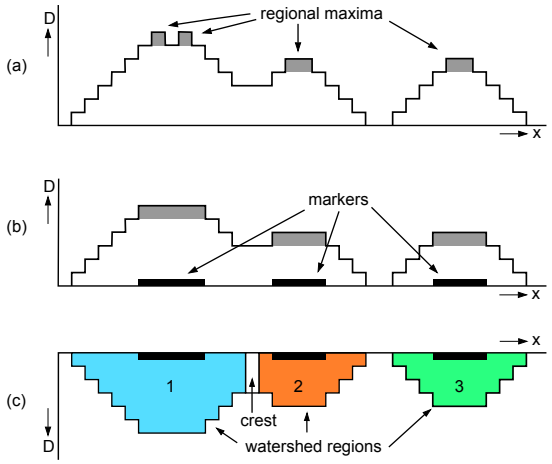

Figure 2.4: One-dimensional illustration of the separation of touching somata by marker-based watershed segmentation. Objects containing two (or more) maxima in the distance field separated by a minimum are assumed to consist of two (or more) clustered neurons and need to be separated. (a) Regional maxima of the distance field D. Due to contour irregularities multiple maxima per object may appear (left object), which is undesired. (b) By subtracting 1 from the distance field at the positions of the regional maxima, and computing the regional maxima of this modified distance field, better markers are obtained. (c) The adapted distance field is "flooded" using the marker positions as initial basins (D has been inverted to better illustrate the basin-flooding metaphor). Positions where the levels from different watersheds meet, are marked as crest regions. After the flooding is completed, crest regions are turned into background. As a result clustered neurons are split at the minima of the distance field, i.e. where the connection between two neurons is thinnest.

2.3.3.2 Grayscale reconstruction

The second step will ideally generate a single marker for each neuron within a foreground object (Fig. 2.2i). One approach to realize this would be to compute the regional maxima of the distance image (Fig. 2.4a). A regional maximum M of a grayscale image I is defined as a connected region of voxels with a given value h (plateau at altitude h), such that every voxel in the neighborhood of M has a value strictly lower than h. An efficient method to compute regional maxima uses a morphological operation called grayscale reconstruction (Vincent, 1993) as implemented by the ITK (Ibanez et al., 2005). Computing all regional maxima results however in too many markers: some objects have multiple very close regional maxima, due to contour irregularities and discretization artifacts. Usually these markers differ by only one level in the distance function. We therefore reconnect such regional maxima by subtracting 1 from the values in the distance image at all marker positions and compute the regional maxima (Fig. 2.2i) of this modified distance function (Fig. 2.4b).

2.3.3.3 Watershed segmentation

The final step uses the set of markers M to assign the set of foreground voxels belonging to each neuron. These neuron regions are found using a procedure called marker-driven watershed segmentation.

The general watershed algorithm uses intensity information to divide a gray-value image into foreground regions (catchment basins), separated by watershed lines (background). This algorithm can be illustrated by a landscape flooding metaphor. The inverted distance image $D' = -D$ can be regarded as a landscape where the minima of D correspond to valleys (Fig. 2.4c). When this landscape is flooded, the water level starts to rise from the valleys (or catchment basins) until the different basins meet at the watershed lines (or crests). Afterwards, each basin corresponds to one of the desired neuron regions and is bounded by the watershed lines and/or the image background.

The marker-driven watershed segmentation ensures that we obtain exactly one region for each marker, by creating an input image $D*$ adapted from D such that (a) its only regional minima are located at the marker positions and (b) its only crest-lines are the highest crest-lines of D that are located between the minima (marker positions). The image $D*$ is then flooded as in the general watershed case. For details we refer to Vincent and Dougherty (Vincent and Dougherty, 1994). We use the flooding algorithm described by Soille (Soille, 1999) in order to efficiently find the neurons corresponding to the markers (Fig. 2.2k).

2.3.4 Model-based cluster splitting

The preceding processing steps result in individual 3D objects (neuron regions) that represent the neuronal somata within the image stack. However, some clusters of neurons are still not separated because they have similar intensities and are so close to each other in the original image that they appear like a single, almost spherical neuron. To separate such clusters we chose a model-based filter. We assume that within each stack there is a single dominant neuron population of neurons with Gaussian distributed volumes and that most clusters could be split by the filters of the first two pipeline steps.

2.3.4.1 Volume histogram and mean neuron size calculation

Regarding the assumption above, we calculate the voxel volume of each watershed region and create a histogram of these volumes. The first histogram bin always comprises many small artificial objects. Hence, taking the second peak of the histogram as the mean volume of the dominant neuron population, a Gaussian distribution is fitted to the histogram at this value (Fig. 2.5a). The distribution for individual neurons (i.e. not part of a neuron cluster) is parameterized by three values, the mean value, the standard deviation and the peak amplitude (μ_1, σ_1, A_1). Clusters of $n = 2, 3, \ldots, N$ connected neurons will also be parameterized by Gaussian distributions. Their according mean value μ_n is simply given by:

$$\mu_n = 1.1 \cdot n \cdot \mu_1 \tag{2.5}$$

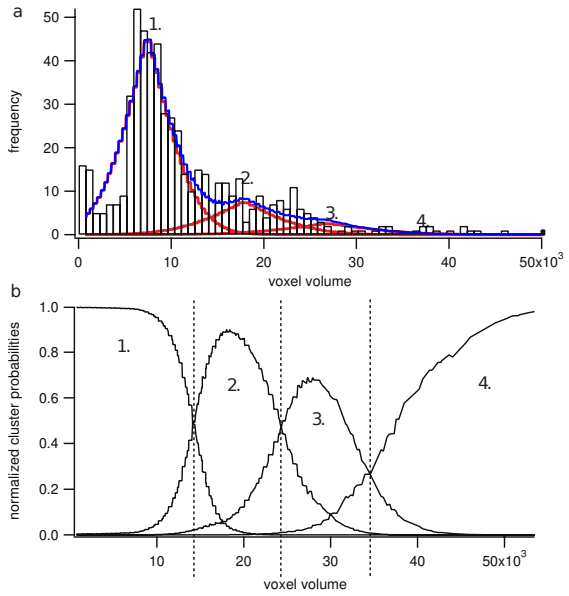

Figure 2.5: (a) Volume histogram of watershed objects. The first mode is assumed to reflect the dominating neuron population and is fitted by a Gaussian (red 1). The higher modes are assumed to be integer multiples of the first mode (here: 3 more Gaussians representing clusters of 2, 3 and 4 neurons respectively). The blue plot represents the sum of the four distributions. (b) Normalized probability values for each object volume. According to the above distributions four probability values are calculated for each object volume and normalized to 1. The resulting 4 plots are shown. The intersections (vertical dashed lines) represent volumes that separate the clusters.

The additional multiplication with 1.1 is due to the assumption that the unresolved gaps between the neurons add a small amount to the cluster volume, and is derived experimentally. The standard deviation for each cluster type σ_n is calculated by standard error propagation:

$$\sigma_n = \sqrt{n} \cdot \sigma_1 \tag{2.6}$$

and the according amplitude A_n is calculated by averaging the height of the five bins around μ_n. This is done until the largest 3D region (object) in the watershed image is covered by a distribution (Fig. 2.5a).

2.3.4.2 Evaluation of cluster type

Each watershed object can only belong to a single distribution. Therefore the probability p_{mn} for each object and for each distribution is calculated, where m refers to the object number and n to the nth cluster distribution (number of connected neurons). Each p_{mn}

value is normalized to one (Fig. 2.5b), resulting in N normalized cluster probabilities for each watershed object.

$$\overline{p}_{mn} = \frac{p_{mn}}{\sum_{n=1}^{N} p_{mn}} \tag{2.7}$$

Each object is regarded to consist of as many neurons as its highest normalized probability value. However, objects that are smaller than the mean neuron volume minus twice the standard deviation are considered to be artifacts and ignored during the further processing.

2.3.4.3 Splitting of clusters by k-means cluster analysis

Once each watershed object is assigned a most likely cluster type (i.e. $n = k = 1, 2, \ldots, N$ neurons), the according k reference voxels (landmarks) are calculated. We therefore use a k-means clustering algorithm as implemented by the ITK (Ibanez et al., 2005). The k-means algorithm works as follows:

1. The input for each watershed object are k ($= n$; i.e. most likely number of neurons in a cluster) initial mean values (default landmarks) specified as k random voxels within the object.

2. Each voxel of a watershed object is assigned to its closest landmark among the k mean values.

3. Calculation of each k-means cluster's mean from the newly assigned landmark voxels within the objects and hence updating the k mean values of a neuron cluster.

4. Repetition of step 2 and step 3 until the termination criterion is met, here if no voxel changes its cluster membership from the previous iteration.

This results in k landmark voxels for an object (consisting of $k = n$ neurons). Each landmark is used as position reference of an individual neuron and a list of these landmark voxels is visualized in Amira (Stalling et al., 2005) (Fig. 2.2l).

2.3.4.4 Correction for 2nd population of larger neurons

The assumption of a single Gaussian distributed neuron volume across the entire image stack can cause miss-counting if two neuron populations of significantly different volumes are present. Here we assume that there is only a small spatial overlap between these populations and that one of them is dominant. We argue that this is a reasonable assumption for the presented image sizes of $375\,\mu m \times 375\,\mu m \times 50\,\mu m$ (confocal microscope with $40\times$ objective). If the minor population is smaller in volume than the majorities' mean volume (i.e. GABAergic interneurons), these neurons will still be counted as one. However, if the minority population consists of large neurons (Fig. 2.6) the prior described splitting will result in an overestimation of the neuron density.

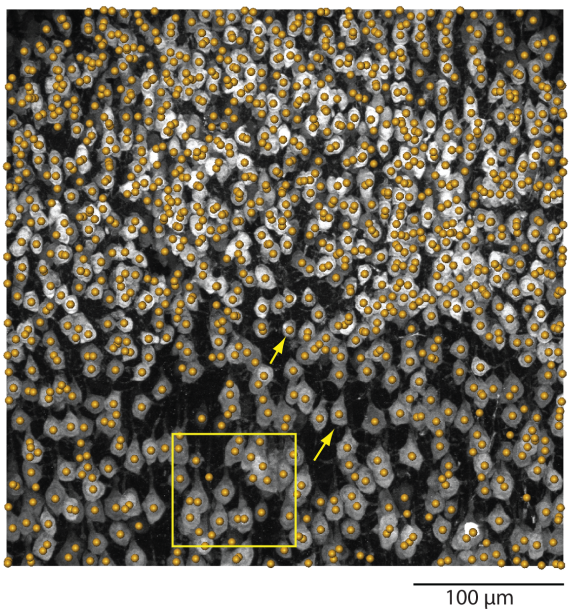

Figure 2.6: Maximum z-projection of NeuN-stained confocal image stack. The yellow box represents the representative volume shown in Fig. 2.2. This stack is in an area of the cortex where layer 4 neurons (upper part) overlap with larger layer 5 neurons (lower part). Even though two neuron populations with significantly different volumes are present, the correction algorithm avoids miscounting (e.g. neurons indicated by arrows).

This issue is addressed by evaluating the local surrounding of each watershed object before model-based splitting. If more than 10% of the watershed objects in a surrounding box of $300 \times 300 \times 100$ voxels have the same cluster probability (larger than 1), the object will not be split (Fig. 2.6). This filter also corrects for systematic errors of the pre-processing step. In bricks with low SNR the neurons are slightly enlarged compared to bricks of high SNR. If significant SNR gradients are present within the image stack, the systematic increase in volume in low SNR bricks can lead to an artificial second neuron type. This means that the neuron volume in some regions of the image can be systematically larger compared to the original image stacks. This effect is also compensated by the described correction filter.

2.4 Methods for visual validation

The development of a segmentation algorithm that consists of a sequence of image filters, each with its own parameters, is often a trial-and-error process that requires a frequent evaluation of intermediate results. As an objective quantitative measure is not always available, this evaluation is commonly done visually. For an efficient

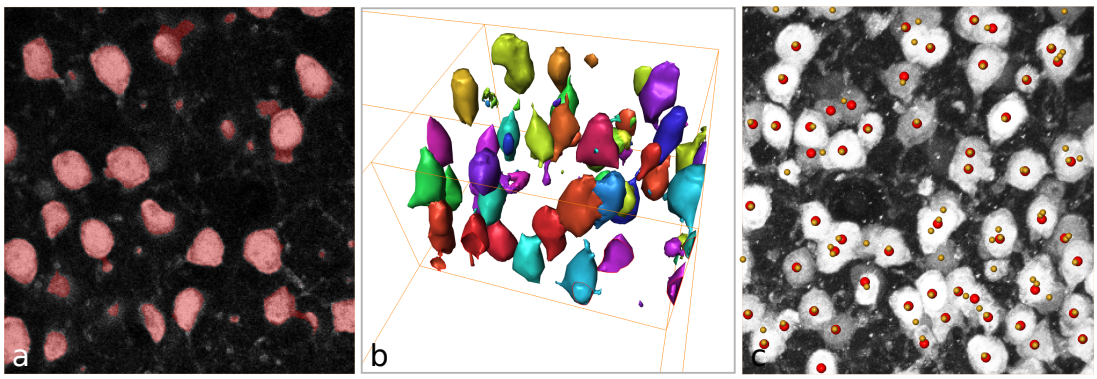

Figure 2.7: Visual algorithm validation. (a) The result of the binary segmentation step is visualized by blending the binary mask (red) with the original image data. (b) A 3D surface representation of each region allows for validation of the cluster splitting tasks. Clipping planes (front, left) reduce visual clutter. (c) 3D validation of location and number of soma centers by simultaneous display of automatically (yellow) and manually (red) generated landmark sets. Image data, such as a maximum intensity projection, aids in deciding on correctness of the landmarks.

workflow during algorithm development, easy-to-use visualization tools are therefore indispensible (Dercksen et al., 2012).

A common way to judge the result of a binary segmentation algorithm is to display a gray-value image slice overlaid with the segmented foreground in a semi-transparent color, for example, using Amira (Stalling et al., 2005) (Fig. 2.7a). By moving the 2D slice, the algorithm developer can quickly verify the quality of the segmentation result. If a foreground voxel object needs to be separated into its constituent parts, as, for example, in the cluster splitting steps of the automatic soma detection method, the 2D slicing approach is insufficient. The boundary between two touching objects should be located at a thin 'neck', which is very dificult to verify on 2D slices. In this case, the generation of object boundary surfaces using the Generalized Marching Cubes algorithm (Hege et al., 1997) has proven very useful (see Fig. 2.1c, d; Fig. 2.7b). As in this surface representation each object has a distinct color, the correctness of the splitting result can be determined almost instantly.

Visualization is also very useful to compare differences between automatically computed soma positions and those of a reference data set, e.g., a 'gold standard', created manually by an expert. In order to discover whether a particular false positive or negative result is caused by the automatic algorithm, by expert error, data artifacts, or differences in treatment of only partially imaged boundary objects, etc., one has to look closely at each differing data point in combination with the image data. In Amira, one can for example display both landmark sets with different colors, and show the image data using 2D textured slices with contrast control (Fig. 2.7c), or 3D direct volume ren-

dering. Together with a simple 3D viewer allowing for interactive zooming and rotating, this is a very effective environment for visual method validation.

2.5 Quantitative validation

To evaluate the automatic soma detection method, we compare sets of landmarks as computed by our method to manually created sets that are considered as the "gold standard". It should however be emphasized that the results of the manual counting yielded an inter-user variability of 2.1% (HS Meyer et al., 2010). In general, the scientific interest focuses on neuron densities. Hence the absolute number of neurons within the stack volume and the deviation of the neuron positions from manually defined locations need to be investigated. In addition, we perform a false positive (FP) /false negatives (FN) analysis.

The comparison was performed "double blind", meaning stacks were first evaluated manually by different individuals and afterwards processed by the automated pipeline described above, without prior inspection of the manual counts.

2.5.1 Input image data and gold standard

The evaluation of the automated counting pipeline is done for 11 confocal image stacks of NeuN-stained neuron somata, kindly provided by Hanno Sebastian Meyer (Max Planck Florida Institute). They were randomly chosen from a large data pool that was used for evaluation of neuron densities within a cortical column of the primary somato-sensory cortex in rats (HS Meyer et al., 2010). The stacks were taken at various magnifications, resulting in different resolutions and stack volumes, at various cortical depths ($300 - 1800\,\mu m$ from the pia surface), yielding different dominant neuron populations in each sample (e.g. layer 4 spiny stellates or layer 5 pyramidal neurons). The stacks were acquired from $50\,\mu m$ thick physical sections from cortical tissue of rats. Large three-dimensional confocal image stacks were generated by mosaic/optical sectioning. Mosaic refers to multiple overlapping images (e.g. 4×4), each representing one microscopic field of view. In the following, the pipeline was always applied to single fields of view (i.e. 1024×1024 pixel wide image stacks). Pixel size was $0.232\,\mu m \times 0.232\,\mu m$ ($63\times$ magnification) for Stack 1 in Table 2.1, and $0.366\,\mu m \times 0.366\,\mu m$ for Stacks 2–11 ($40\times$). Sampling distance along the z-direction during optical sectioning was $0.61\,\mu m$. For further image acquisition details we refer to Oberlaender et al. (Oberlaender et al., 2009) and Meyer et al. (HS Meyer et al., 2010).

The "gold standard" was created as follows. Image stacks containing single mosaic tiles were loaded into Amira 4.0 or 4.1 (Stalling et al., 2005). Landmarks (3D voxel coordinates) were assigned manually to the center of all neuron somata during a careful examination of the image planes (optical sections). Objects at the stack border in the x- or y-direction were always counted. The x/y overlap was set to approximately $5\,\mu m$. In consequence neurons at the x/y border (overlap area) of mosaic tiles were detected twice. By aligning the mosaic images, twice detected neuron somata coincided and could

Stack	#Manual	#Automatic	Abs. # diff.	Rel. diff. (%)	Pos. dev. (μm)
1	430	425	5	1.16	4.54
2	846	896	-50	-5.91	3.93
3	570	594	-24	-4.21	4.23
4	871	915	-44	-5.05	4.74
5	1219	1237	-18	-1.48	4.75
6	809	776	33	4.08	1.92
7	719	670	49	6.82	2.05
8	739	680	59	7.98	2.07
9	1035	983	52	5.02	2.70
10	862	846	16	1.86	2.96
11	975	976	-1	-0.10	3.16

Table 2.1: Comparison of manual and automated neuron counts. 14 image stacks from different cortical layers, at different magnification and acquired with different imaging techniques were compared. Differences in numbers (relative and absolute) and landmark positions are of acceptable quality and similar to inter-user variability of manual counts.

be erased. If the mosaic area is chosen larger than the area of interest, x/y border effects can hence be completely neglected. The x/y border rule was not applied at the z-borders of the image stacks. Here neurons were regarded to be within the image stack if their diameter increased to a maximum value and decreased again or was constant for three more optical sections before reaching the stack border. A detailed description of the manual counting, the validation of the border criteria and the approximate inter-user variability of 2.1% can be found in (HS Meyer et al., 2010).

2.5.2 Counting difference

Table 2.1 shows the results for the comparison of automatically and manually detected landmarks. We observe that no systematic miscounting is performed by the automated pipeline. The number of detected neurons differs similarly in both directions resulting in an average counting difference that is less than 1% (0.92% averaged over the number of data sets). However, the average absolute counting difference in landmark numbers is around 4%. This value is regarded as the absolute error of the automated counting.

2.5.3 Deviation in landmark position

The average radius of neurons in S1 is between 5 and 15 μm. If the deviation between a manually placed landmark and its automated counterpart is sufficiently lower than the smallest radius value of 5 μm, the error in position can be neglected and the absolute difference in landmark numbers is regarded as a meaningful error value.

For evaluation of position deviation the following algorithm was used:

1. Calculate the distance from each automated landmark to each manual landmark.

Stack	FN[%]	FP[%]
1	9.30	2.78
2	7.23	6.31
3	4.71	7.58
4	4.65	5.35
5	3.19	3.01
Mean	5.82	5.01
Std	2.43	2.08

Table 2.2: False positive landmarks (FP), automated neuron position references with no manual counterparts and false negative landmarks (FN), manual position references with no automated counterparts were investigated for 5 confocal NeuN image stacks. Both values are similar and therefore compensate each other, explaining the low values of relative neuron number differences.

2. Sort the distances, starting with the shortest one and assignment of this manual landmark as the nearest neighbor (NN) of the automated one.

3. Check for multiple times assigned NN. If more than one automated landmark has the same NN, the closest automated one will keep this NN, the other automated landmarks are assigned to have their second closest manual landmark as their NN.

4. Repeat step 3 until no manual landmark is assigned to more than one automated landmark.

5. Compute the average distance of all automated landmarks to their assigned manual NN (see Table 2.1).

The average deviation in position of $3.37 \pm 1.11\,\mu m$ is sufficiently less than $5\,\mu m$. We also observe that the position deviation for each dataset is smaller than $5\,\mu m$. Hence the error in position can be neglected, because each automatically determined landmark has a manual counterpart within reasonable distance. It is therefore justified to state that the absolute difference between manually and automatically detected neuron somata is 4%.

2.5.4 False positive/negative landmarks

In addition to the above considerations about average differences between manual and automated counts, we performed a false positive/negative analysis for five randomly chosen stacks. We visualized the two corresponding landmark sets in Amira and manually deleted corresponding landmarks. The remaining automated landmarks were regarded as FP and the remaining manual landmarks were regarded as FN. These FP/FN objects are usually ambiguous cases of touching neurons that could for instance either be counted as one or two neuron somata. Table 2.2 shows the results for this analysis.

Both, the average FP and FN values are around 5%. This compensatory effect explains the average relative counting difference of less than 1%.

Further, the automated approach is much faster, reducing manual labor of approximately $4\,h$ per stack to a few minutes and additional computing time of about $1\,h$.

2.6 Discussion

We presented a novel approach for fast automated detection of neuron somata in 3D images. The processing pipeline is based upon three steps: threshold-based pre-processing, watershed-based and model-based cluster splitting. The method was illustrated by applying it to NeuN-stained confocal image stacks. However, by adjusting the pre-processing step, the pipeline can be adapted to deconvolved widefield stacks of NeuN-labeled neurons and to *in vivo* 2-photon images of Ca^{2+}-sensitive neurons (Oberlaender et al., 2009).

By comparing the automated results with manually generated counterparts and regarding these manual counts as the "gold standard" (i.e. neglecting the 2.1% inter-user variability), the automated system correctly reproduces manually detected neuron somata with more than 90% accuracy. The deviations originate from detection of ambiguous objects, meaning touching neurons that could for instance be counted as one or two neurons. Compensatory effects of approximately 5% FP as well as FN detections from ambiguous clusters of neuron somata result in average relative counting differences of less than 1%. The average error in absolute numbers is approximately 4%.

Further, the automated approach is much faster than manual counting, reducing manual labor of approximately $4\,h$ per stack to a few minutes and additional computing time of about $1\,h$.

However, since the last processing step is model-based and hence restricted by constraints, the approach presented here has certain limitations. We assume that a single dominating neuron population with Gaussian-distributed volumes is present within each field of view (1024×1024 pixel wide image stack). If a second minor population of large neurons is present within the image, it is assumed to be spatially separated from the dominant population. In summary, the first mode of the watershed object volume histogram should not be blurred by a second or third population. These constraints are met for the presented "small" image size of $375 \times 375 \times 50\,\mu m^3$ for most brain regions. Nevertheless, isolated large neurons (e.g. in Layer 5B of S1) can lead to a significant overestimation of neuron density At this stage these singular events are corrected by manually editing the landmark file while superimposing the original image stack in Amira.

We would like to argue that our method is not only more accurate than classical sparse sampling methods, but will reveal new insights in brain organization and will hence produce deviating results from previous studies. By investigating a volume of $\sim 8\,mm^3$ of S1 in rats, we observed significant density variations along all three axes within each cortical layer (HS Meyer et al., 2010). Hence, sparse sampling methods will yield strongly deviating results, depending on the sample position and orientation

within the cortex. By counting the absolute neuron number and hence obtaining the 3D neuron distribution across the entire area of interest, our method is independent of sample orientation, position or shrinkage.

In summary, the here presented counting pipeline is a fast, reliable and objective alternative to manual counting of neurons. It is applicable to different brain areas, yields absolute numbers and 3D neuron distributions. It is therefore possible to obtain statistically valid neuron densities for functional subunits in the brain, revealing their detailed anatomical basis.

3 Automatic alignment of sections containing filamentous structures

3.1 Introduction

Physical sectioning combined with optical sectioning enables researchers to image large tissue volumes at submicrometer resolution. Optical sectioning microscopy, e.g. using confocal laser scanning or transmitted light brightfield microscopy, permits the three-dimensional imaging of the internal structure of tissue by varying the focal plane of the microscope. However, the resolution that can be obtained is limited by the thickness of the tissue, due to light scattering. Thick tissue can be imaged by first cutting the specimen into thin sections, which are mounted and optically sectioned individually to produce a stack of thin, but three-dimensional subvolumes (Bajcsy et al., 2006; Oberlaender et al., 2007). When using electron tomography instead of light microscopy, individual cells can be studied at even smaller scale (McIntosh, 2007).

To reconstruct the entire object, the subvolumes representing the physical sections have to be aligned, because the individual sections may have different positions and orientations on the microscope slides. There are basically two different approaches to this problem: image-based and feature-based alignment. Problems with image-based alignment include imaging artifacts like noise, uneven brightness and poor contrast (particularly towards the top and bottom of the section, the crucial regions for correct alignment), as well as the potentially large data sets (multiple gigabytes per section). For volumes containing filamentous structures, the information that could be used to obtain a correct alignment is often very sparse, and may be difficult to discriminate from other, non-reliable structures in the image. It is therefore often beneficial to first extract features from the image data and perform the alignment on the features. This is the approach taken here.

We present an automatic method to align a pair of subvolumes containing filamentous structures (published previously (Dercksen et al., 2009)). The problem can be described more specifically as follows (see Fig. 3.1): given two subvolumes S_0 and S_1 containing a potentially different number of filaments (polylines), we are looking for a transform T which maps the end points of filaments in S_1 onto the corresponding end points in S_0 in an optimal way. A prerequisite for computing T is the assignment of correspondences between filaments in both slices, for which a solution is presented as well. We assume that enough corresponding filaments are available to find a correct match and that the slices have not been severely deformed beyond a rigid transform. We demonstrate the

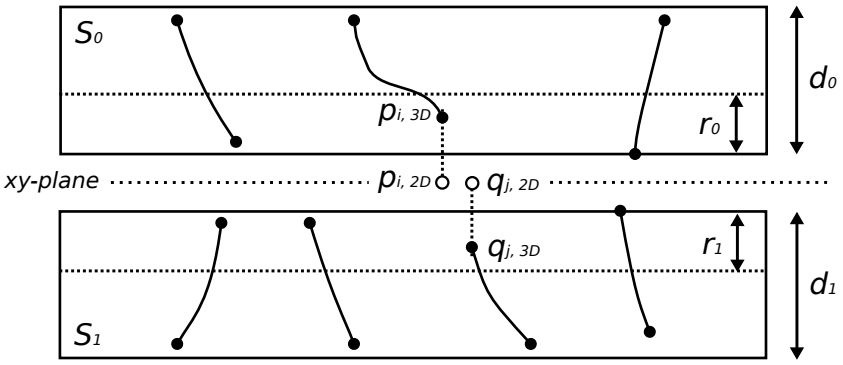

Figure 3.1: Alignment of sections containing filamentous structures. The two sections S_0 and S_1 have to be aligned. In our approach, the set of filament end points $\{\mathbf{p}_{i,3D}\}$ in S_0 and $\{\mathbf{q}_{j,3D}\}$ in S_1 in boundary regions, defined as a fraction $r_{\{0,1\}}$ of section thickness $d_{\{0,1\}}$, are projected on the xy-plane. The projected point sets $\{\mathbf{p}_{i,2D}\}$ and $\{\mathbf{q}_{j,2D}\}$ are aligned in 2D. The resulting transformation is applied to the entire section.

effectiveness of the fast and robust method by applying it to neuron and microtubule data sets.

3.2 Related work

Many different methods exist for alignment based on image data directly (Dercksen et al., 2008; Maintz and Viergever, 1998; Oliveira and Tavares, 2014; Szeliski, 2006) and image-based feature descriptors (Mikolajczyk and Schmid, 2005; Zitová and Flusser, 2003). Such methods however cannot be used in our case, because our objects of interest are very thin and difficult to distinguish, especially in noisy images with background structures in the same intensity range.

In our case, the filaments are obtained by manual or automatic tracing (Meijering, 2010; Weber et al., 2012) and the features to be aligned are the filament end points. Several methods have been proposed to solve the challenge of simultaneous optimization of 1) point correspondences between two point sets of possibly different size and 2) the transform with respect to a particular objective function.

The Softassign Procrustes Matching Algorithm (Rangarajan et al., 1997) casts the combined continous and combinatorial optimization problem into a nonlinear optimization problem by regarding the entries in a match matrix as probabilities. The solution is obtained by iteratively optimizing the correspondences and the transformation. Chui (Chui and Rangarajan, 2003) extends this method for non-rigid point matching. Luo (B Luo, 2001) also extends the Procrustes analysis, but casts the problem into an Expectation Maximization framework, similarly to Myronenko et al. (Myronenko and Song, 2010), who provide a closed-form solution to the maximization step of the EM

algorithm for a general multidimensional case. These methods usually require a suitable initialization in order to converge to the optimal solution.

Hogrebe et al. (Hogrebe et al., 2012) align axonal branches in neighboring sections by a rigid and a non-rigid method. The rigid alignment establishes an initial correspondence between axon end points by comparing a point's histogram containing the number of points within a set of distances (the bins). The matching is optimized by a RANSAC (Fischler and Bolles, 1981) approach. For the non-rigid registration, the correspondence is computed by comparison of each point's 2D polar histogram of point positions, filtered by criteria such as maximum distance and incidence angles. The correspondences are refined after warping the points using a B-spline transform with a control grid of increasing resolution.

Bajcsy et al. (Bajcsy et al., 2006) solved the problem in order to align segmented extracellular matrix proteins and blood vessels contained in a stack of Confocal Laser Scanning Microscopy subvolumes. A rough initial transform is computed by finding a matching of filaments with similar cross-section area and similar distances between the centroids. The transform is then optimized taking into account also angular constraints. In (SC Lee and Bajcsy, 2008), the transform for the found matching is further optimized. A polynomial function is fit to the filaments and extrapolated onto a plane between the subvolumes. The optimal transform is then computed by minimizing position deviation of the extrapolated points or by maximizing curve smoothness.

Characterization of molecule similarity, important in drug design, is another problem often posed as 3D point matching (Akutsu, 1996; Baum, 2007). Kirchner (Kirchner, 2007) optimizes an objective function involving the fraction of matched points and the root mean square distance (*rmsd*) between matched points after transformation. He shows that there exists a fully polynomial time approximation scheme (FPTAS) for this problem, which however runs in polynomial time of high degree.

3.3 A point-matching-based method for section alignment

Of all methods methods mentioned above, we chose to use the point matching approach of Baum (Baum, 2007) as a basis for the alignment of filament data, because it is fast, robust, flexible and provides a good initialization of the optimization process. In its original form, it computes the correspondence between two 3D point sets of possibly different size and the rigid transform that optimally maps one point set onto the other (in a least squares sense). We extend it to the alignment of 2D point sets obtained from traced filaments, describe how to incorporate an optional uniform scaling in the transform and present some efficiency optimizations.

Given two sections containing multiple polylines, our method computes an optimal transform, i.e. a rotation angle around the z-axis, a 2D translation parallel to the xy-plane and an optional uniform scaling, using the following steps:

1. Find the sets of points $P = \{\mathbf{p}_i\}$ from the first slice S_0 and $Q = \{\mathbf{q}_j\}$ from the other slice S_1, to be matched. We use the line end points in the section boundary

regions r_0 and r_1 respectively (see Fig. 3.1) and obtain 2D point sets by orthogonal projection onto the xy-plane. The region size is chosen as a fraction of the total slice thickness: $r_0 = \beta d_0, \beta \in (0, \frac{1}{2}]$.

2. Find a set of candidate matchings and compute a starting transform for each matching.

3. For each starting transform, optimize the matching and transform with respect to a scoring function. The result is the transform corresponding to the optimal score.

3.3.1 Computation of candidate point matchings

Given two point sets $P = \{\mathbf{p}_i\}$ and $Q = \{\mathbf{q}_i\}, \mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^2$, the first step to find the optimal matching and the corresponding transform, is to compute a set of candidate matchings.

A matching M defined on two finite sets P and Q is a bijective function $M : \tilde{P} \to \tilde{Q}, \tilde{P} \subseteq P, \tilde{Q} \subseteq Q$. M can be written as a set M^* of pairs from $P \times Q$ with $(\mathbf{p}, \mathbf{q}) \in M^* \Leftrightarrow M(\mathbf{p}) = \mathbf{q}$. The number of pairs $|M^*|$ will be referred to as the size of M.

Candidate matchings are found by looking at the Euclidean distances between point pairs. The goal is to find subsets of points in P and Q that have the same spatial pattern, i.e. similar mutual distances. In particular, the goal is to find a matching $M : \tilde{P} \to \tilde{Q}$ such that the distance between each point pair $(\mathbf{p}', \mathbf{p}'')$ in $\tilde{P}$ and the distance of the corresponding point pair $(\mathbf{q}', \mathbf{q}'')$ in $\tilde{Q}$ differ by no more than a given distance threshold d:

$$\big|\|(\mathbf{p}' - \mathbf{p}'')\| - \|(\mathbf{q}' - \mathbf{q}'')\|\big| \leq d. \tag{3.1}$$

If this condition holds for each point pair in $\tilde{P}$, we say that M is a d-bounded matching. The points in both sets then have a similar positional pattern and M is thus considered a candidate matching.

The parameter d is data-dependent. It should be chosen small for tracings obtained from tissue that was little deformed and larger for more deformed specimens.

Usually, multiple d-bounded matchings exist. In order to find them all, we use a method based on clique detection in the *distance compatibility graph* (DCG). The vertices of this graph consist of all $(\mathbf{p}', \mathbf{q}')$ pairs. Thus, there are $|P| \cdot |Q|$ vertices. Two vertices $(\mathbf{p}', \mathbf{q}')$ and $(\mathbf{p}'', \mathbf{q}'')$ are connected by an edge, if the matching M given by $M^* = \{(\mathbf{p}', \mathbf{q}'), (\mathbf{p}'', \mathbf{q}'')\}$ is d-bounded (see Eq. 3.1). A clique in a graph is a subset of vertices that induce a complete subgraph, i.e. each vertex in the subset is connected to all other vertices in the subset. A clique is maximal when it is not part of any larger complete subgraph. In the following, we use clique to refer to a maximal clique. We find all cliques (and thus all candidate matchings) using the Bron-Kerbosch (Bron and Kerbosch, 1973) algorithm.

For each candidate matching we compute a transform that minimizes the *root mean square distance* (*rmsd*) of the matched points. The *rmsd* of two point sets P and Q with respect to a matching M and a rigid body transform T is defined as:

$$rmsd(P,Q,M,T) = \sqrt{\frac{\sum_{(\mathbf{p},\mathbf{q})\in M^*} \|\mathbf{p} - T(\mathbf{q})\|^2}{|M^*|}} \tag{3.2}$$

This transform is called the *matching transform* of M and can be computed in $\mathcal{O}(|M^*|)$ time, using the algorithm described in (Kabsch, 1978). The matching transforms of all candidate matchings are the starting transforms $\{T^{init}\}$ serving as the input for the following optimization step.

3.3.2 Optimization of matching and rigid transform

In this step, we find the optimal matching M and transform T with respect to a scoring function *score*:

$$score(P,Q,M,T) = \frac{|M^*|}{min(|P|,|Q|)} \cdot e^{-\alpha \cdot rmsd(P,Q,M,T)} \tag{3.3}$$

The scoring function measures the number of matched points and the root mean square distance of the matching points after transformation. Its range is $[0,1]$, where 1 corresponds to the best score. The optimal score is achieved when all points have been matched and the transform maps the corresponding points exactly onto each other. The parameter $\alpha \in [0,1]$ weighs the contradicting goals of a large match (α small) versus small positional error (α large). In an iterative algorithm, we maximize *score* alternatingly with respect to M and to T, see Algorithm 1. M is optimized by a simple and fast greedy approach (Kirchner, 2003), see Algorithm 2. The scoring function ensures that no point pairs will be assigned to M that are too far apart. T is optimized by computing the *matching transform* for a given M. The *score*-function is maximized once for each starting transform T_i^{init}, using the respective T_i^{init} as initialization. The final result of the alignment algorithm is the matching and transform corresponding to the highest *score* across all computed *score* maxima.

3.3.3 Extension: Uniform scaling for deformed data

For data that suffered from severe deformations, an additional uniform scaling can lead to a better alignment. The scaling factor s for the matching transform is computed as follows:

$$s = \frac{\sum_{i=1}^{|M*|} \mathbf{p}_i' \cdot \mathbf{q}_i'}{\sum_{i=1}^{|M*|} \mathbf{q}_i' \cdot \mathbf{q}_i'}, \quad \mathbf{p}_i' = \mathbf{p}_i - \mathbf{c}_{P'}, \quad \mathbf{q}_i' = R(\mathbf{q}_i - \mathbf{c}_{Q'}), \tag{3.4}$$

where $\mathbf{c}_{P'}$ and $\mathbf{c}_{Q'}$ are the average point positions of the matched sets P' and Q' respectively. R is the rotation computed as in the rigid case. However, this approach only works if a suitable clique can be found. The points in such a clique are usually spatially concentrated, as the effect of the global scaling on their absolute mutual distances is limited (see e.g. Fig. 3.2b).

Algorithm 1: Optimize matching and transform

Input: Point sets P and Q, set of starting transforms $\{T_i^{init}\}$
Output: Near-optimal matching M_{opt}, transform T_{opt}

foreach T_i^{init} **do**
  $s_{new} \leftarrow 0$
  $T_i \leftarrow T_i^{init}$
  repeat
    $s_i \leftarrow s_{new}$
    $M_i \leftarrow greedyPointMatching(P, Q, T_i)$
    $T_i \leftarrow computeTransform(P, Q, M_i)$
    $s_{new} \leftarrow score(P, Q, M_i, T_i)$
  until $s_{new} \leq s_i$
end
$(M_{opt}, T_{opt}) \leftarrow \arg\max_i score(P, Q; M_i; T_i)$

3.3.4 Performance optimization

As the computation of the starting transforms is independent of the point matching optimization, we can generate a set of starting transforms using different point sets and use these for the optimization step. When these point sets are smaller than the original, computing time for clique detection will be reduced. In the case of microtubuli, for example, one could select the endpoints of filaments which are nearly parallel to the z-axis (and thus have a good chance of having a matching counterpart) for the computation of the starting transforms, resulting in much smaller point sets and a significant speed-up. Of course one can also use the reduced point set for the optimization step, as we do in Section 3.5.2. As long as enough reliable points remain, a correct alignment can be achieved much faster.

A second efficiency improvement is a reduction of the number of starting transforms to compute the optimal matching for, by retaining only transforms corresponding to cliques larger than a certain minimal size c (we set $c = max(2, 0.3 \cdot min(|P|, |Q|))$).

3.4 Visual alignment validation method

The automatic alignment method is integrated in the Filament Editor, a software framework for proof-editing reconstructions of filamentous structures (see Ch. 4) from multiple sections. Here, the following visualization techniques are used to aid the user in assessing alignment correctness:

1. Filament coloring by section. The same color is applied to the filaments within each section. A small set of clearly discernable colors is assigned repeatedly to ensure that section membership can be easily determined (Fig. 3.2a).

Algorithm 2: Greedy Point Matching

Input: Finite point sets P and Q, transform T
Output: Near-optimal matching M_{opt}

$M_0^* \leftarrow \emptyset$
$P' \leftarrow P$, $Q' \leftarrow Q$
$k \leftarrow 0$
while $\exists p \in P', q \in Q'$ **do**
  $(p, q) \leftarrow \arg \min_{p \in P', q \in Q'} \|\mathbf{p} - T(\mathbf{q})\|$
  $M_{k+1}^* \leftarrow M_k^* \cup \{(p, q)\}$
  $P' \leftarrow P' \setminus \{p\}$
  $Q' \leftarrow Q' \setminus \{q\}$
  $k \leftarrow k + 1$
end
$M_{opt} \leftarrow \arg \max_k \text{score}(P, Q; M_k^*; T)$

2. Point coloring and scaling. Filament end points are colored according to the following scheme: matched points obtain the same (random) color, unmatched points are black, unused points (i.e., end points not in the boundary regions) are colored white (Figure 3.2b,c; Figure 4.2b). Unmatched and unused points are rendered smaller to improve the visibility of the matched points.

3. Filament and point visibility. For an unobscured view, rendering of points and filaments of user-selected sections can be temporarily turned off (Fig. 4.2c).

4. 3D viewer with interactive camera movement, rotation and zooming.

3.5 Results

3.5.1 Alignment of fragments of neural processes

The automatic alignment algorithm was applied to a stack of 30 slices containing fragments of neural branches (axons) obtained from brightfield microscopy (see Fig. 3.2a). The number of endpoints to match ranged from 3–101.

Almost all slices could be aligned, except two at the top and one at the bottom, where there were too few points (3–6) to produce a valid result. A good match, and thus a good alignment, is characterized by a matching fraction that is large enough to rule out randomness and by the absence of mismatches. The results obtained with parameter values $d = 10\mu m$ and $\alpha = 0.25$ fulfilled these criteria, as they resulted in a match size of 32–100% with a minimum of 5 points and no visually apparent mismatches. The computation time ranges from 0.02–8.5s.

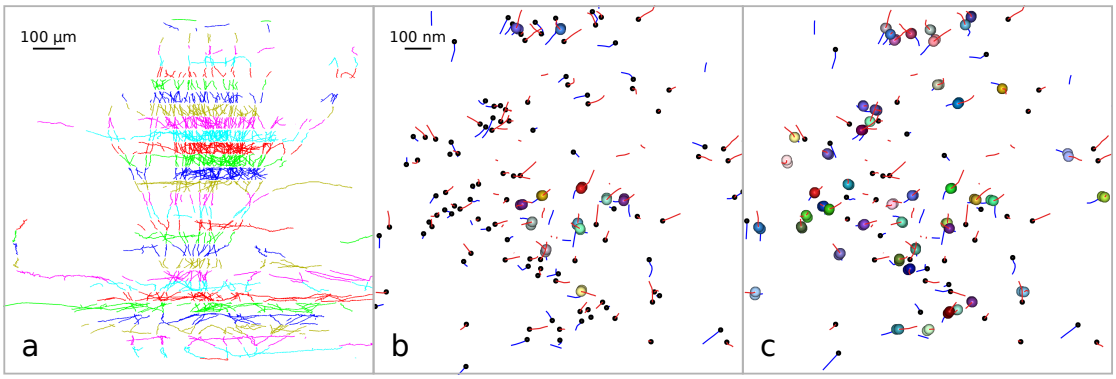

Figure 3.2: (a) Aligned stack containing neuron fragments (filaments colored by section). (b) Rigidly aligned slice pair containing microtubuli. The matched points are clustered in the center of the data set (colored point pairs). Further away from the center, there are many unmatched points (black). Many of those could be matched when using an additional scaling (c).

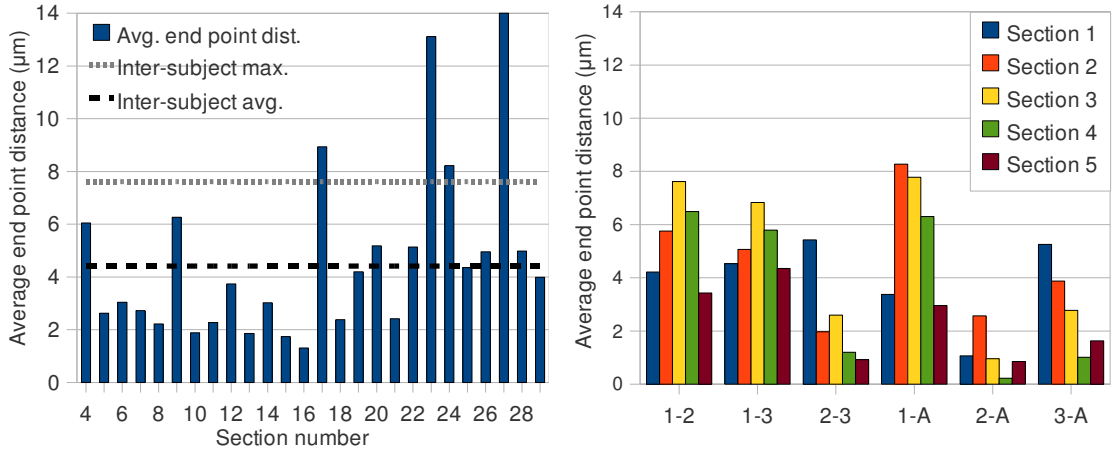

Figure 3.3: Left: Average distance between manually and automatically aligned end points ($\overline{\Delta}_{27} = 25.2\mu m$). Right: Average end point distance between the alignments of three subjects (1, 2, 3) and our automatic method (A). The average and maximum value of all subject-subject combinations are indicated by a line in the left figure.

The successfully aligned sections (4–29) were compared to their manually aligned counterpart. The manual transform T_M and automatic transform T_A were computed for each slice with respect to the untransformed predecessor section. Then the average distance $\overline{\Delta}$ between the filament end points $\mathbf{q}_i$ that served as input for the alignment was computed: $\overline{\Delta} = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \|T_M(\mathbf{q}_i) - T_A(\mathbf{q}_i)\|$. The results are displayed in Figure 3.3 (left).

For comparison we assessed the inter-subject differences in manual alignment. A 6-slice data set was aligned automatically and by three individuals. The average point distance $\overline{\Delta}$ was computed as above for all combinations of manual and automatic alignments (see Fig. 3.3, right). After visual comparison of the outlier sections 23 and 27, the expert who carried out the manual alignment confirmed that the automatic alignment was at least as good. From these results we conclude that the automatic alignment results in similar quality as the manual alignment.

3.5.2 Alignment of microtubule fragments

We also applied our algorithm to the alignment of a stack of 9 sections containing microtubuli, obtained by manual tracing of electron tomography data using the IMOD (Kremer et al., 1996) software (electron tomograms and tracings were kindly provided by A. Hyman and Q. de Robillard, MPI of Molecular Cell Biology and Genetics, Dresden). We failed to manually align the sections as there are many filaments (∼700 per section) but no prominent, clearly identifiable features.

For the automatic alignment the number of points was reduced using the angle-based selection (see Sec. 3.3.4) to increase performance. All lines having an angle smaller than 70 degrees with the xy-plane were ignored, reducing the average point set size from 436 to 61. Parameter values were set to $d = 40\,nm$ and $\alpha = 0.1$.

Our method was able to align six out of eight slice pairs within a few seconds (0.10–6.7s). Visual inspection showed that the matchings are generally correct and of sufficient size (18–66% of the points, with a minimum of 10) to obtain a reliable transformation.

However, an additional number of matching point pairs that were not reported by the automatic algorithm could usually be determined by visual inspection (see Fig. 3.2b). Firstly, this can be explained by the defensive choice for α resulting in almost no false pairings, but also in a reduced number of total pairings. Secondly, the results indicate a non-rigid deformation of the sections. To test this, we added a uniform scaling s as an additional degree of freedom (see Sec. 3.3.3). The resulting scaling factors ranged between 0.5–10%, and particularly for the more severely deformed sections resulted in significantly better alignments (see Fig. 3.2c).

Two slice pairs could not be aligned for reasons that are not yet understood. Analysis is difficult as manual alignment is not feasible.

3.6 Conclusion

We presented an automatic method for the alignment of data stacks containing filamentous structures. It uses a point matching approach to find the optimal rigid transform (with optional uniform scaling) for each section. The application to neuron data showed that it is fast and produces accurate results. Also microtubuli data could be aligned, something that is infeasible to do manually.

4 Reconstruction and analysis of 3D neuron morphologies

4.1 Introduction

During the past 20 years, many technical barriers for reconstructing single neurons have been overcome. Labeling neurons using intracellular (Horikawa and Armstrong, 1988) or cell-attached (Pinault, 1996) pipettes has allowed reconstructing large parts of individual neurons (e.g. (Binzegger et al., 2004; Broser et al., 2008; Broser et al., 2004; Oberlaender et al., 2011), thus linking their structure with activity patterns *in vitro* (e.g. (Feldmeyer and Sakmann, 2000; Schubert et al., 2006) and *in vivo* (Oberlaender et al., 2012). In addition to such conventional techniques, genetic labeling methods based on fluorescent proteins have started the identification and reconstruction of relatively uniform, molecularly identified cell populations (Groh et al., 2010). Further, digital imaging has advanced rapidly. New imaging methods, such as high-speed mosaic/optical-sectioning widefield (Oberlaender et al., 2009; Oberlaender et al., 2007) and confocal (Kleinfeld et al., 2011) systems, as well as block-face two-photon (Ragan et al., 2012) and light-sheet (HU Dodt et al., 2007) microscopes promise high-resolution imaging of large brain regions. Finally, the limitations on archiving terabyte data sets have disappeared with the falling costs of hard disk drives.

As a result of these developments various manual, semi- and fully automated approaches for reconstructing single neuron morphologies have been reported (see (Donohue and GA Ascoli, 2011; Meijering, 2010) for reviews). The hence rapidly increasing number of reconstructed neuron morphologies gave rise to collaborative efforts that collect single neuron reconstructions – the most comprehensive being the NeuroMorpho.org repository (GA Ascoli, 2006; GA Ascoli et al., 2007) – or assemble neuronal network models for computer simulations of cortical signal flow, e.g., the Blue Brain Project (Markram, 2006). However, the vast majority of single neuron tracings has so far been obtained from *in vitro* preparations, i.e., individual neurons are labeled within a tissue slice of usually $300\,\mu m$ thickness, e.g., the Blue Brain Project relies entirely on *in vitro* tracings (SL Hill et al., 2012). Unfortunately, *in vitro* tracings suffer from cut off dendrites and axons (Oberlaender et al., 2012). Reconstructing the complete 3D dendrite and axon morphology of individual neurons thus requires *in vivo* labeling in combination with histological sectioning of the brain (i.e., due to penetration limits of staining and imaging methods) and subsequent imaging of large tissue volumes. Tracing of faintly labeled, long-range projecting thin neurites, and recovering across-section continuity of neuronal branches, thus render reconstructions of *in vivo* labeled neurons as

a major challenge in neuroscience research (e.g., see 'DIADEM' competition (Svoboda, 2011)).

Nevertheless, for sparsely labeled tissue, reconstruction results are usually assumed to be highly reliable. However, little validation of this reliability has been reported (Helmstaedter et al., 2011). Reconstruction validation requires tools that allow users (i) to assess the correctness of the tracings by visual comparison with the image data, and (ii) to interactively correct incomplete or erroneous tracings (Luisi et al., 2011). In addition, in case of *in vivo* data, alignment and interconnection of tracings across brain sections must be established and verified. In consequence, we argue that at present, reconstruction reliability limits the determination of complete 3D morphologies from *in vivo* data.

The *Filament Editor* (FE) presented here (and published previously (Dercksen et al., 2014)) is an integrated software environment specifically designed to reconstruct and validate single neuron tracings from *in vivo* preparations. It comprises tools for visualizing and interactively correcting 3D neuron tracings, alignment, and across-section continuity, allowing for efficient proof-editing within and across brain sections. In addition, the FE incorporates advanced annotation and morphometric analysis functionalities. We illustrate the applicability of the FE on frequently occurring use cases and demonstrate that the proof-editing routines can result in unambiguous tracings of *in vivo* labeled axons.

4.2 Related work

Two common approaches to obtain 3D neuron morphology reconstructions are sparse labeling (only a small fraction of the neurons in a tissue volume is stained, often only one), followed by light-microscopic (LM) imaging of the tissue, and dense labeling, followed by electron-microscopic (EM) imaging.

Helmstaedter (Helmstaedter, 2013) provides an overview of current dense reconstruction approaches. He contends that purely automatic methods lack sufficient accuracy, therefore human intervention is required. Two approaches can be distinguished. First, neurites are traced manually and highly redundantly, followed by automated computation of the consensus skeleton (Helmstaedter et al., 2011) and potentially a volume reconstruction (e.g. (Andres et al., 2008)). Alternatively, data sets are segmented automatically, often involving a classifier trained on user input, followed by interactive proofreading of the result (Mishchenko, 2009; Sommer et al., 2011). The user proofreads the result by inspecting the volumetric segmentation, visualized by color-blending the labeling onto the original image data; Jeong et al. (Jeong et al., 2010) provide also 3D volume rendering. Corrections are performed, for example, by modifying the training set of the classifier (Sommer et al., 2011), or splitting, merging and relabeling objects (Kaynig et al., 2013).

However, neurons may innervate several cubic millimeters and extend over tens of centimeters. Currently only sparsely labeled LM images can be used for their recon-

struction, due to the limited volume (e.g. $60 \times 350 \times 300\,\mu m^3$ (Briggman et al., 2011)) that can currently be imaged using EM. Therefore, we will focus on sparse reconstruction below.

Peng et al. (Peng et al., 2011) argue that despite great advances in automated tracing methods (for a review see (Donohue and GA Ascoli, 2011; Meijering, 2010)), proof-editing remains a necessary, but laborious process. Besides a fully automated proof-editing system that learns by example to predict the different error types and their bounds, Peng et al. regard a highly ergonomic 3D interactive WYSIWIG (What You See Is What You Get) system as a solution to this problem. Although a wide array of digital tracing tools are available (reviewed in (Parekh and GA Ascoli, 2013)), only few software packages implement a system that provides effective visual verification and fast interactive correction of tracings. The FARSIGHT Trace Editor (Luisi et al., 2011), V3D (Peng et al., 2010), Neuromantic (Myatt et al., 2012) and NeuronStudio (Wearne et al., 2005) are some examples of tools that provide simultaneous display of image and tracing data for verification and a set of tools to modify the latter. Some aspects in which the above mentioned tools differ are the degree of usability (e.g., undo/redo functionality), whether the tracing can be viewed and edited in 3D (supporting human pattern recognition to quickly resolve ambiguities), the available measurement functions and supported platforms (Windows, Linux, Mac).

The FE was designed and implemented to meet high standards with respect to all of these aspects. In addition, the FE provides alignment and splicing functionalities to merge multi-section data sets, allowing for a final quality control by checking continuity of tracings across sections, which none of the above-mentioned packages does.

4.3 Method for proof-editing, visualization and analysis of 3D neuron reconstructions

The Filament Editor has been developed based on the Amira software (FEI Visualization Sciences Group, 2013a), a visualization framework, implemented in C++, which is frequently used in neuroscience research (e.g. (Ertürk et al., 2012; Halavi et al., 2012), NeuroMorpho.org accepts Amira file format).

4.3.1 Input data

The FE takes as input 3D tracings of neuronal branches (or other types of fibrous structures). These must be represented as polylines, i.e. connected linear line segments. In addition, image data from which the branches were traced can be provided, either as 3D stacks, or in 2D, e.g., as a maximum intensity projection. Further, tracings from multiple image stacks, e.g., consecutive brain sections, can be provided for alignment and merging.

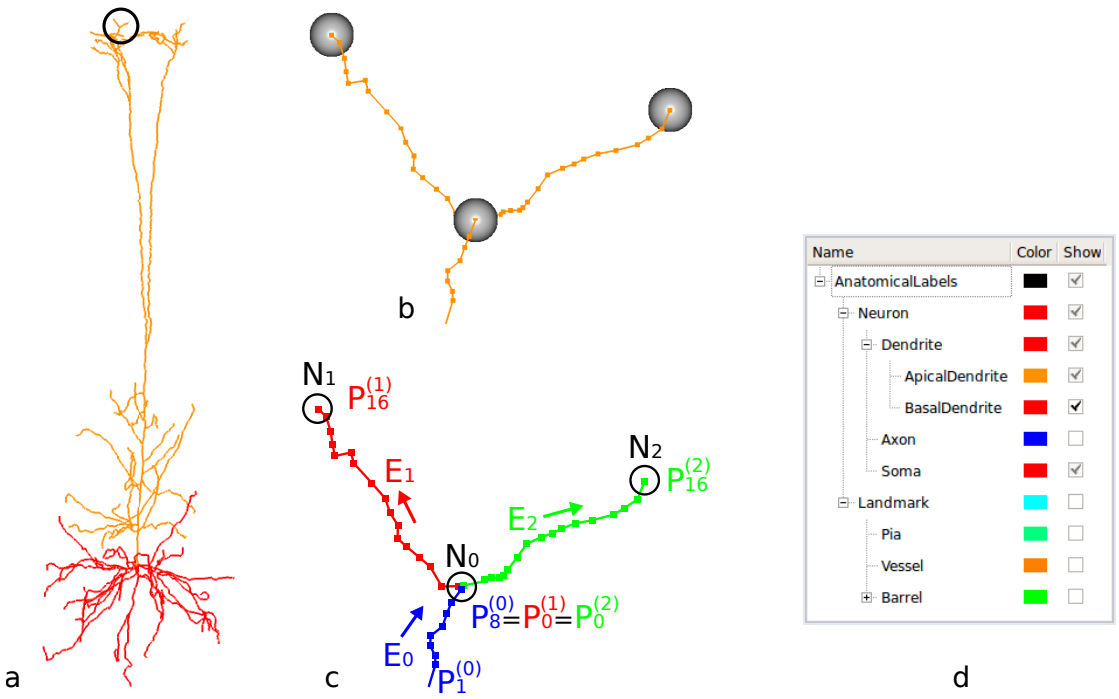

Figure 4.1: The SpatialGraph data structure. (a) Dendritic tree of a layer 5 pyramidal neuron (Oberlaender et al., 2012), colored according to the anatomical labels defined in (d). (b) Magnification of the encircled region of (a). The nodes (grey spheres) are connected by edges, represented and displayed as polygonal lines defined by edge points (or points, for short; displayed as squares). (c) Schematic representation of the SpatialGraph object in (b). The object consists of three directed edges (E_0, E_1, E_2) which interconnect nodes N_0, N_1, N_2 (the source node of E_0 is not displayed). At branching nodes, the last point of the incoming edge ($P_8^{(0)}$ on edge E_0) coincides with the first point of the outgoing edges ($P_0^{(1)}$ and $P_0^{(2)}$ on E_1 and E_2 respectively). (d) Semantic information can be associated with the morphology using labels. The Label viewer *is used for editing label hierarchies, assignment of labels to graph elements, and selection of these elements for editing, visualization and analysis.*

4.3.2 Data structure

Neuron morphology is represented in the FE by the *SpatialGraph* data structure (Fig. 4.1), which is similar to previously reported graph formats (Mayerich et al., 2011). Specifically, the topological skeleton of the graph is defined by a set of nodes, connected by a set of (unbranched) edges. The edge direction is defined by specification of the source and target node. Edges can be treated as undirected by ignoring this information. The graph is embedded in space by associating 3D coordinates with each node. The edge

trajectory is defined by a sequence of 3D points (vertices). The first and last point of each edge coincides with the source and target node, respectively.

The data structure was designed to allow interactive editing and rendering of large 3D tracings (e.g. >10k edges, >1M points). For traversing the graph, an adjacency list is maintained for each node. This approach is more memory-efficient than adjacency matrices (Skiena, 1998) for storing the present sparse graphs. Further, the explicit representation of the (high-level) topological structure using nodes and edges allows for more efficient traversal than through linked lists of (low-level) points (e.g. *SWC* (Cannon et al., 1998)) or segments (e.g. *MorphML* (Crook et al., 2007)). Points are stored as coordinate arrays per edge for efficient rendering as line strips.

Different types of attribute data can be associated with the nodes, edges and points of the *SpatialGraph* for visualization and morphometric analysis. Labels are used to associate semantic information with substructures of the graph, e.g. 'Dendrite', 'Axon' and 'Soma' (Fig. 4.1a, d). Additionally, numerical attributes can be defined, e.g. a floating-point value representing the radius at each edge point. Multiple attributes can be defined on nodes, edges and points simultaneously. For each attribute defined on nodes, an array is generated holding one value for each node. The array size thus equals the number of nodes. Edge and point attributes are stored analogously. The attribute arrays are kept up-to-date throughout the editing process to match the graph structure. Tracings can be imported into the FE using the *SWC* (Cannon et al., 1998), *hoc* (Carnevale and Hines, 2006) or the *amiramesh* (FEI Visualization Sciences Group, 2013b)) file format. These formats, as well as *MorphML* (Crook et al., 2007) can be used to export the tracings.

4.3.3 Visualization

The FE provides a 2D and 3D viewer for proof-editing a tracing. The 3D viewer of the FE displays the graph using spheres for the nodes, squares for the points and polylines for the edges. The user can inspect the spatial structure of the tracing in 3D by camera rotation, zooming and panning using the mouse, and edit each of the graphs' components. Nodes, edges and points can be colored according to one of the label attributes. Their displayed size is user-adjustable. Besides traced morphologies, the 3D viewer can display additional data using any available Amira display module (FEI Visualization Sciences Group, 2013b). For example, the neuron morphology can be jointly visualized with the image data using volume rendering, 2D slices or intensity projections. The *SpatialGraph* can be visualized as tubes by mapping a radius attribute defined on edge points to cylinder thickness (rendering is based on (Sigg et al., 2006)).

The 2D viewer displays a slice of user-defined thickness of the graph, superimposed on a maximum intensity projection (MIP) of the corresponding 3D image region (FEI Visualization Sciences Group, 2013b). Hiding the remaining image and tracing regions provides an unobstructed view to locally verify the tracing with respect to the image stack. By varying the slice depth and/or orientation, the user navigates through the

volume. The slice MIP and bounding box can be displayed in the 3D viewer to provide additional spatial orientation.

Both viewers can be used either side-by-side or one may be used exclusively. User-selected sets of nodes and edges can temporarily be hidden in both viewers by excluding them from rendering. Images are using standard bitmap formats such as *BMP*, *JPEG*, *PNG*, and *TIFF*, or in the *amiramesh* format. Stacks of 2D images are converted into 3D volumes during import.

4.3.4 Selection tools

Modification of the traced morphology is achieved by selecting one or more elements (nodes, edges, points), followed by the invocation of an operation (e.g. deletion) (McGuffin and Jurisica, 2009). To provide a selection system that is 'powerful' (i.e., allowing efficient selection of any subset of elements) and 'forgiving' (i.e., the selection itself can easily be modified) (Wills, 1996), the following tools are available in the FE:

- *Single-Element Selection*: selects single nodes, edges or points that have been clicked on with the mouse.
- *Connected-Component Selection*: selects the entire subgraph connected to the element that has been clicked on with the mouse;
- *Lasso Selection*: selects all nodes and edges within a user-drawn polygon. Using a modifier key, only connected components that are completely contained within the Lasso polygon are selected.
- *Select-All*, *Clear* and *Invert Selection*: selects the entire *SpatialGraph*, deselects and inverts the current selection, respectively.
- *Label Selection*: selects graph elements with a particular attribute by clicking on the respective label in the Label Viewer (Fig. 4.1d).

Holding a modifier key adds to the current selection. Selected items are highlighted in red in the viewers.

4.3.5 Editing operations

The following operations are available to manipulate the data structure:

- *Deletion* of selected nodes and edges;
- *Splicing* (connecting) edges. Splicing is achieved by selecting either two nodes, two points, or one point and one node, followed by the invocation of the connect operation. Nodes are connected by a new straight edge. Selected points are converted to nodes as they will be branching points; the new nodes are subsequently connected. The splicing operation connects multiple selected elements at once as

follows. First, all nodes and edges connected to the currently selected elements are added to the selection, resulting in multiple connected subgraphs. An edge is added between any two terminal nodes from different subgraphs that have the smallest Euclidean distance until all subgraphs are connected.

- *Point-to-node conversion* converts a selected point into a node, resulting in an edge split. The inverse *Node-to-point* operation concatenates two edges, while removing the intermediate node. The latter can be run for a single selected node or for all intermediate nodes at once. Together with the *Remove-isolated-nodes* operation, all nodes that are neither branching nor terminal nodes can be purged from the graph.

- *Edge-smoothing.* Edges are smoothed by repositioning their edge points as follows: the user specifies the neighborhood size N, which must be odd. The position of each point $\mathbf{p}_i$ on the edge is replaced by the average of its own position and the $M = (N-1)/2$ points along both directions:

$$\mathbf{p}_i^{'} = \frac{1}{N} \sum_{k=i-M}^{i+M} \mathbf{p}_k \tag{4.1}$$

  The first and last M points on the edge are not moved, effectively leaving all nodes in place.

- *Transformation.* An affine transformation can be applied to tracings, images or any other 3D data set (FEI Visualization Sciences Group, 2013b). For example, a linear scaling could be applied, independently for all dimensions, to compensate for tissue shrinkage.

All editing operations can be undone/redone. Shortcut keys are defined to efficiently switch between the different selection and editing tools, as an alternative to clicking tool buttons (Nielsen and Mack, 1994).

4.3.6 Semantic labels

Semantic information can be associated with nodes, edges and points by assigning labels. A label is a name (i.e. a string), uniquely identified by an integer value and associated with a color. Labels are organized in a hierarchical fashion, i.e. in a tree data structure, providing access to graph substructures at different levels of detail. Upon creation of a node (or edge/point) label attribute, each node (or edge/point) is assigned a label from the tree.

The *Label Viewer* (Fig. 4.1d) is part of the FE user interface allowing the user to (i) define and edit custom label trees, (ii) assign labels to selected substructures of the graph and (iii) select these substructures for editing, visualization and analysis. For example, one could define a hierarchical label tree with root label 'Neuron' having child labels

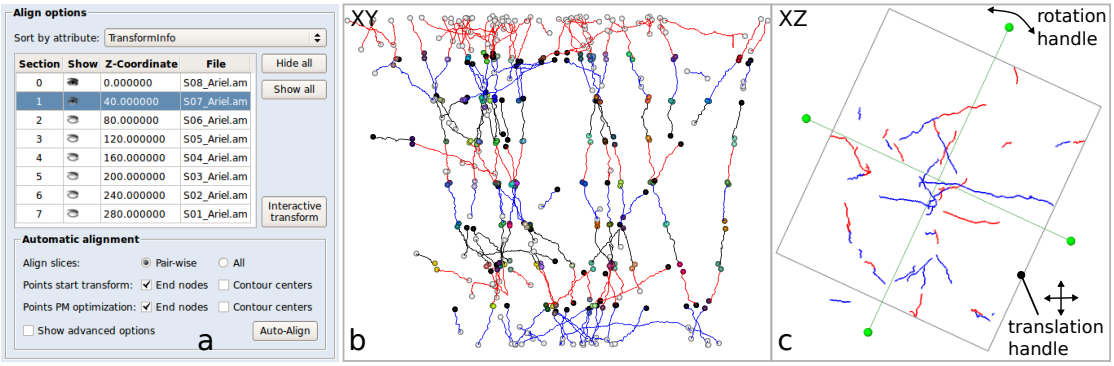

Figure 4.2: Section alignment. (a) The Align *toolbox provides access to both automated and interactive alignment. The section list allows a user to (i) select a section to align, (ii) toggle section visibility, and (iii) manually adjust the z-position of a section. (b) XY-view after alignment of eight sections. Edges are colored by section, alternating red, blue, and black. Nodes are colored by matching result. Matching nodes in different sections are assigned the same label, and thus the same color. Black nodes could not be matched; white nodes were not used for alignment. (c) Interactive alignment using a handle. Only the section that is currently transformed (red, highlighted in the table in (a)) and its predessor (blue) are visible; other sections are hidden not to obscure the view.*

'Axon' and 'Dendrite' and assign these to the respective edges and nodes. Selecting the root label (i.e. 'Neuron') in the Label Viewer highlights all edges and nodes assigned to child labels recursively (i.e. 'Axon' and 'Dendrite'), while selecting 'Dendrite' would highlight the subset of 'Dendrite' elements exclusively. Alternatively, the Label Viewer supports flat label hierarchies (i.e., only one level of labels below the root). For example, the 'identify loops' functionality of the FE (FEI Visualization Sciences Group, 2013b) automatically assigns all edges comprising a loop the same label, pinpointing potential autapses (i.e., an intersecting dendrite and axon from the same cell) or falsely connected branches.

4.3.7 Section alignment

The FE incorporates an automated method (Dercksen et al., 2009) for rigid alignment (Zitová and Flusser, 2003) of tracings obtained from adjacent image stacks, e.g. from consecutive brain sections (see Ch. 3). The automated algorithm is complemented with a user interface for interactive manual alignment (Fig. 4.2a, c).

First, tracings from adjacent image stacks are merged using the *CreateSpatialGraphStack* module. This module generates a new *SpatialGraph* by positioning tracings from image stacks obtained from adjacent brain sections at either fixed distances along the z-axis (i.e. perpendicular to the cutting plane) or such that the bounding boxes adjoin. Tracings from image stacks obtained from the same brain section can be merged with-

out (z-)translation for subsequent manual alignment. The nodes and edges are assigned identifier labels that refer to the corresponding image stack. Transformations can thus be applied to tracings of each individual stack by transforming all nodes and edges with a particular label.

The stack of tracings is aligned by repeated pair-wise alignment of neighboring sections. The automated method (Dercksen et al., 2009) uses a 2D point matching approach to compute the optimal transform, i.e. a rotation angle around the z-axis and a 2D translation parallel to the xy-plane. The points to be matched are the terminal nodes in the top and bottom region of each section, e.g. within the upper and lower 25% (Fig.4.2b). The algorithm seeks to maximize the number of matching points, while minimizing the positional difference between matched points, weighting these conflicting goals, similar to other feature-based alignment methods (Szeliski, 2006). This approach is sufficiently fast to be used in an interactive workflow (Dercksen et al., 2009), with response times ranging from a fraction of a second for a small number of end points ($\sim$30), to several seconds for a larger set ($\sim$100).

Sections can be aligned simultaneously or pair-wise. The resulting alignments immediately appear in the 3D viewer, allowing visual validation (Fig. 4.2c). In cases where the automated alignment result is not satisfactory, the user can interactively translate and rotate each section with respect to its predecessor section in the 3D viewer using handles. All other sections can be hidden to not obscure the view.

4.3.8 Interactive manual tracing

The interactive tracing functionality available in the 2D viewer (FEI Visualization Sciences Group, 2013b) can be used to manually add filamentous and/or anatomical reference structures. The user adds new nodes by clicking on the desired location in the image. The node is added at the mouse cursor position at the depth of the image plane (for 2D images) or the depth with highest image intensity (3D image). The node is automatically connected to the latest created node by a new edge. This feature can be used to augment automatically generated tracings, for example with contours representing anatomical landmarks, such as pia, white matter or blood vessel outlines (Fig. 4.3a), as well as with 2D outlines of the soma in different optical and or histological sections.

4.3.9 Visual and quantitative morphometric analysis

The FE includes the following tools for visual and quantitative morphometric analysis (Fig. 4.3):

- Branch length and node statistics (number of branching, terminal nodes) for the entire graph or for a selected subgraph, grouped by semantic labels.
- Quantitative measurement and visualization of branches within/outside a reference volume (Fig. 4.3a). Closed 3D surfaces representing such reference volumes can be generated from 2D contour outlines of anatomical landmarks, as well as from

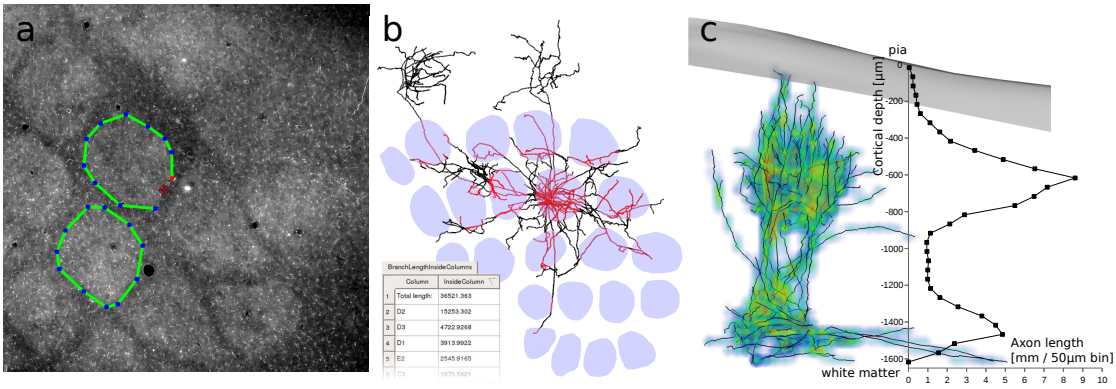

Figure 4.3: Morphometric analysis. (a) Manual drawing of anatomical landmark contours. Contours outlining so called barrel columns *in rat vibrissal cortex (S1) are created by clicking on the MIP image. A new node (red) is automatically connected to the previously created node by an edge (green). (b) Visual and quantitative analysis of semantically labeled neuronal branches with respect to anatomical reference structures. Here, axonal length of a Layer 5 slender-tufted pyramidal neuron is evaluated with respect to the barrel columns in S1 (Oberlaender et al., 2011). The axon length within and outside each column was automatically computed, exported to a spreadsheet, and visualized using the Filament Editor by labeling all edge points as either inside (red) or outside (black) any cortical column. (c) Branch density visualization and quantification. A grid of* $50\,\mu m$ *voxels is superimposed onto a reconstructed thalamocortical axon labeled in rat vibrissal thalamus in vivo (Oberlaender et al., 2012). For each grid cell the total axon length is computed and visualized in 3D or as 1D density profile by accumulating length densities across each xy-plane.*

contours representing neuron somata (Fuchs et al., 1977). The length contained within a volume is computed by intersecting all edge segments with the triangles comprising the bounding surface, performing a point location test (Skiena, 1998) and accumulating the length of the confined branches. The result is output to a spreadsheet. The parts of the morphology contained within a volume can further be visualized by labeling the edge points by the name of the respective structure and coloring the graph according to these labels (Fig. 4.3a).

- 3D density of morphological properties. A 3D grid of user-defined voxel size is superimposed onto the morphology and morphological properties of the tracing within each voxel (Fig. 4.3b) are computed. Particular properties of interest are the number of branching nodes or branch length. Branch length within a voxel is computed by clipping each segment between adjacent edge points against the voxel's bounding box (Liang et al., 1992) and accumulating the resulting lengths. Semantic labels allow distinguishing between substructures.

- 1D profile of morphological properties. By accumulating the values in voxels in each plane (e.g. having the same z-value) of the 3D voxel grid described above, a 1D profile of the property of interest along an axis can be extracted (see Fig. 4.3c and (HS Meyer et al., 2010; Oberlaender et al., 2012)).

4.3.10 Visual and quantitative comparison of tracings

Quantitative comparison of tracings obtained from the same image data set requires defining their correspondence. Because of the intricate relation between topology and geometry, at present, correspondence definitions need to employ heuristics (Gillette et al., 2011; Mayerich et al., 2011).

Here, we implemented such a comparison metric, based on (Helmstaedter et al., 2011), as follows: Given the set of reconstructions $R_1, \ldots, R_N$ (N being the number of users), first, edge points are inserted on all edges (without modifying the trajectory), such that the distance between any two edge points is smaller than a maximum sampling distance D.

Second, the reconstruction R_1 is pair-wise compared to the reconstructions of all other users $R_2 \ldots R_N$. For each edge point $\mathbf{p}$ on R_1 the number of reconstructions is counted that have a point $\hat{\mathbf{p}}$ that corresponds to $\mathbf{p}$. $\hat{\mathbf{p}}$ corresponds to $\mathbf{p}$ if they are no further than a maximum correspondence distance R apart, i.e., $|\hat{\mathbf{p}} - \mathbf{p}| \leq R$. This results in the number of corresponding edge points $C(\mathbf{p}) \in [1, N]$; $C(\mathbf{p}) = 1$ if there is no other reconstruction with a matching point (R_1 agrees only with itself in $\mathbf{p}$), $C(\mathbf{p}) = N$ if all other reconstructions have a matching point.

Third, the total length of all edges of R_1 is divided among N bins $L_1, \ldots, L_N$. A bin L_i represents the length of R_1 that was agreed on by i reconstructions. To this end all straight segments between pairs of successive edge points $\mathbf{p}$ and $\mathbf{q}$ are regarded. If $C(\mathbf{p}) = i$, then half the length of segment $\mathbf{pq}$ is added to L_i, same for $\mathbf{q}$. The length

bins are computed separately for all other reconstructions $R_2, \ldots, R_N$. The length per bin is presented to the user in a spreadsheet.

To visualize the correspondences, an integer attribute is defined on the edge points, storing the value of $C(\mathbf{p})$ for each point. Displaying one or more reconstructions colored according to this value, similar to (Mayerich et al., 2011), effectively pinpoints inter-user differences.

4.3.11 Implementation

The FE is integrated in the Amira (FEI Visualization Sciences Group, 2013a) /ZIBAmira (Zuse Institute Berlin, `http://amira.zib.de`) visualization framework. The FE GUI contains the 2D and the 3D viewer, the Label Viewer, a toolbar containing the selection and editing tool-buttons and a set of toolboxes. First, the *View* toolbox contains GUI elements to control the display of the nodes, edges and points. Second, the *Align* toolbox (Fig. 4.2) provides access to manual and automated alignment functionalities. Finally, the *CorticalColumn* toolbox provides tools for morphometric analysis with and without respect to anatomical landmark contours.

The Amira/ZIBAmira framework is implemented in C++. It uses Qt 4.6 for the graphical user interface (GUI) and the OpenInventor (FEI Visualization Sciences Group, 2013c) library for 3D rendering. The main `FilamentEditor` class is a *Singleton* (Gamma et al., 1994) that holds references to the image and *SpatialGraph* data, the highlighted and visible selections, the viewers and other GUI elements. This class acts as a *Mediator* (Gamma et al., 1994) that (i) accepts requests from tools and GUI elements to change its data members (e.g. editing operations, selection/visibility changes), (ii) applies the changes, (iii) redraws the views and (iv) notifies the tools and toolboxes regarding the changes by emitting differentiated Qt signals. Tools and toolboxes can attach a Qt slot to each of these signals allowing them selectively update their state (*Observer* pattern (Gamma et al., 1994)).

The `SpatialGraphSelection` class represents the selection status of the nodes and edges as bit vectors (for constant-time lookup) and an array of selected points (the number of points can be large, but only few are usually selected at once). Single element selection uses the picking functionality of OpenInventor. The connected component selection tool uses a depth-first traversal (Skiena, 1998) of the graph, starting at the picked graph element. The Lasso selection uses available Amira functionality that (i) lets the user draw a 2D polygon, (ii) rasterizes it resulting in a screen-sized bitmask indicating the polygon containment for each pixel and (iii) answers point containment queries by projecting a given 3D point to screen coordinates and looking up the bitmask value. To select all nodes and edges encircled by the polygon, the Lasso tool first selects all contained nodes. Then all points of edges having both source and end node selected are checked for containment. Edges with all of their points located within the polygon are added to the final selection.

All editing operations are derived from an abstract `Operation` class, following the *Command* pattern (Gamma et al., 1994). Each `Operation` subclass takes the currently

highlighted selection as input and implements an `execute()` and an `undo()` method, the latter reversing the effect of the former. An `OperationSet` is a special subclass that represents a sequence of elementary `Operations`, allowing multi-step modifications to be handled as one. The main `FilamentEditor` class receives `Operations` from the editing tools, calls their `execute()` method and pushes them onto the undo/redo stack.

4.4 Results

4.4.1 Application example: 3D reconstruction of individual axons labeled in vivo

We illustrate a potential workflow using the tools of the FE to generate complete 3D morphologies for the example of *in vivo* labeled axons, reconstructed by a previously reported automated imaging and tracing pipeline (Oberlaender et al., 2007). The example workflow consists of the following steps: (i) preprocessing: sample preparation, imaging, and tracing, (ii) proof-editing of tracings of individual image stacks and (iii) alignment and proof-editing of tracings across multiple image stacks. The steps will be described in detail below.

4.4.1.1 Preprocessing: sample preparation, imaging and tracing

Briefly, individual neurons in rat vibrissal thalamus (for a gallery of reconstructed cells see (Oberlaender et al., 2012)) were filled with biocytin using whole-cell (Margrie et al., 2002) patch-clamp recordings *in vivo*. This technique allows labeling the complete dendritic and axonal projections of individual neurons (Horikawa and Armstrong, 1988). After perfusion, brains were fixed and cut into 50 μm thick consecutive vibratome sections. Histological staining with DAB (Wong-Riley, 1979) enhances the contrast of the biocytin-labeled neuronal branches in each section. Using a custom-designed mosaic/optical-sectioning brightfield microscope (Oberlaender et al., 2009; Oberlaender et al., 2007) and Surveyor image acquisition software (Objective Imaging Ltd.), 3D image stacks of typically $2\,mm \times 2\,mm \times 0.05\,mm$ volumes were acquired in vibrissal cortex at a resolution of $0.184\,\mu m \times 0.184\,\mu m \times 0.5\,\mu m$ per voxel (i.e., at 100× magnification).

Neuronal structures were automatically extracted from eight consecutive image stacks using a previously reported and validated automated tracing software named *NeuroMorph* (Oberlaender et al., 2007). Projection images of each image stack, as well as 3D skeletons of neuronal branches (i.e. *SpatialGraph* files) represent the final results of the automated pipeline for each brain section. A position label is assigned to all edges. The labels indicate whether the connected component containing the edge (i) touches the top of the section (blue), (ii) touches the bottom (green), (iii) passes through (i.e. touches top and bottom; red) or (iv) touches neither top nor bottom (yellow). Thus, a fragmented neuronal process passing through an image stack generally consists of one or more green, zero or more yellow, and one or more blue parts (Fig. 4.4).

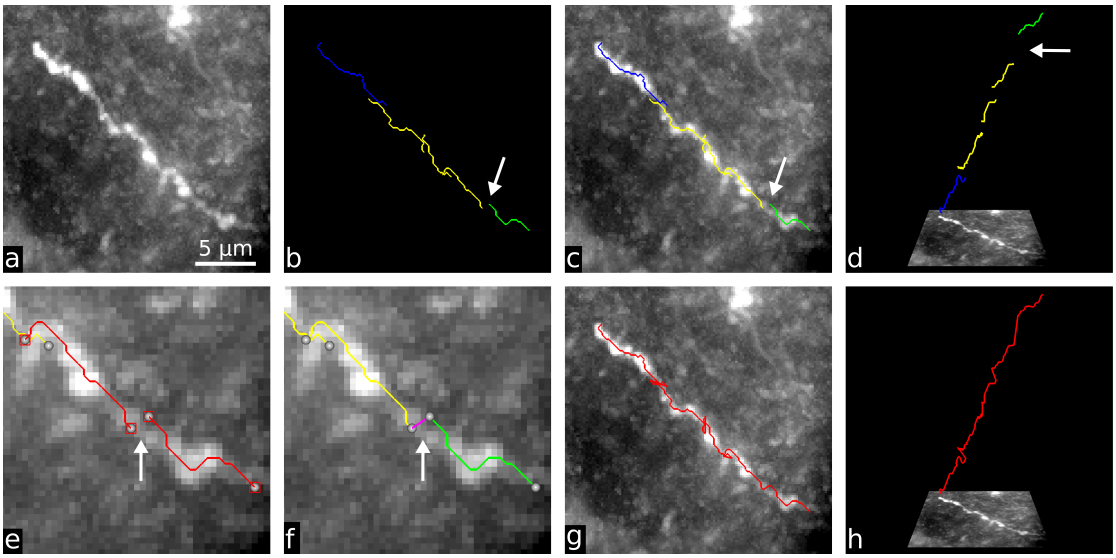

Figure 4.4: Splicing fragmented branches. (a) In the 2D MIP image (from an inverted brightfield image stack) an axon fragment can be clearly distinguished from background. (b) Based on 3D information, the automated tracing algorithm generated a fragmented result (possibly due to faintly stained regions in the axial direction), with connected components labeled as touching top (blue), bottom (green) or neither top nor bottom (yellow). (c) MIP and tracings are superimposed in the FE. (d) Rotating the view immediately reveals the order in which the axonal fragments have to be spliced. (e) After zooming in on the region pointed at by the arrow, two branches are selected. (f) The selected branches are connected using the splicing tool, resulting in an added edge segment (magenta). (g) Splicing of the remaining fragments. (h) The 3D view facilitates visual validation.

4.4.1.2 Proof-editing of tracings from individual image stacks

The FE is used to interactively splice (connect) fragmented edges and to delete falsely traced edges from the automated tracing in each individual section. The user loads the automatically traced neuron fragments and displays them in the 3D viewer, colored by position labels. The tracing data is superimposed on the maximum intensity projection image (MIP) of the respective image stack. Four typical proof-editing situations are illustrated in Figures 4.4–4.7.

Case 1: *Splicing* (Fig. 4.4). In the MIP, neuronal processes are easily distinguishable from background structures (Fig. 4.4a). Nevertheless, the tracing algorithm may generate a fragmented result (Fig. 4.4b, c), for example due to poorly stained regions in the axial direction. In such cases, the position label colors facilitate identification of potentially contiguous parts. However, the lack of depth information in a 2D projection view hampers user decision-making on which fragments are to be connected and in what order. By 3D rotation of the camera (Fig. 4.4d), human pattern recognition can quickly

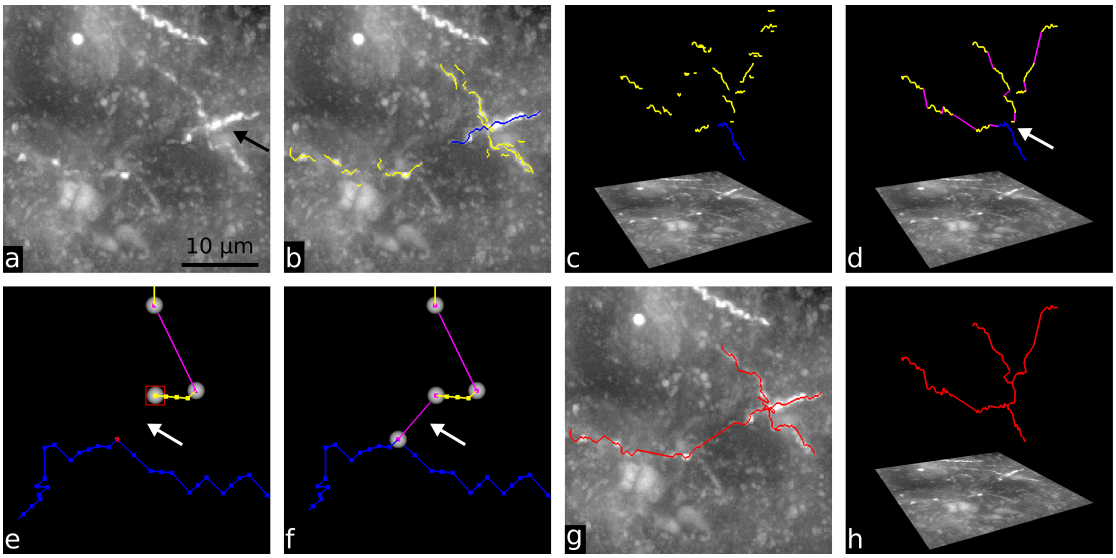

Figure 4.5: Connecting edges at a branching point. (a) Axonal fragments (black arrow), whose 3D configuration is difficult to identify from the 2D MIP. (b) Even when displaying the position labels, the correct configuration remains ambiguous in 2D. (c) Rotation of the camera immediately reveals the three-dimensional configuration of the fragments. (d) After splicing axonal fragments (magenta segments) and removal of false segmentation results, branches may have to be connected at branching points (one such point is indicated by the arrow). (e) Close-up of the region pointed at by the arrow in (d). Nodes are displayed as circles, edge points as small squares. The edge point to be turned into a branching node and the terminal node of the upper edge are selected and (f) spliced, resulting in a point-to-node conversion and a new edge connecting the selected node with the new branching node. (g) The reconstruction result superimposed onto the MIP, and (h) viewed in 3D.

resolve this. Using the selection and splicing tools of the FE, gaps between identified fragments can be closed (Fig. 4.4e–h).

Case 2: *Creating new branching points* (Fig. 4.5). A new branching connection is created by selecting a terminal node and a point on a different edge. The latter will be converted into a branching node (Fig. 4.5d, e) when applying the splice operation (Fig. 4.5f). The most likely location of the new branching node can easily be identified by the operator, using the position-dependent coloring (Fig. 4.5b) and 3D camera rotation of the tracing and MIP (Fig. 4.5c). The *Splicing* operation invokes a *point-to-node* conversion of the selected point. The new node is connected to the selected node by a new edge, turning the former into a branching node. These steps result in a fully connected structure (Fig. 4.5g, h). Any intermediate nodes (having exactly two incident edges) are removed in a final step by invoking the *node-to-point* operation.

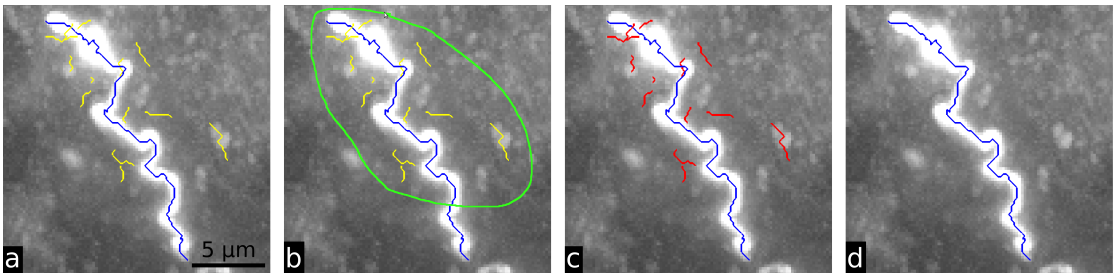

Figure 4.6: Removal of false segmentations. (a) A correctly traced neurite segment (blue) is surrounded by oversegmented fragments (yellow). (b) To remove these, the lasso selection tool is employed, using the modifier key that ensures that only connected subgraphs are selected that are completely contained within the user-drawn polygon (green). (c) Thus only oversegmentations are selected (red), (d) which can be quickly deleted.

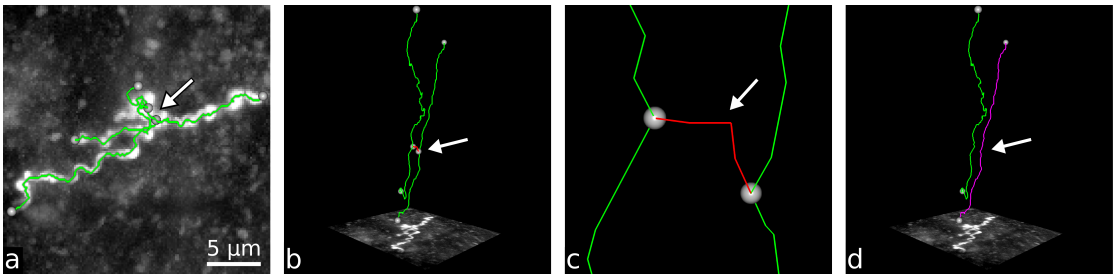

Figure 4.7: Removal of false connections. (a) Axonal branches, superimposed on the MIP. Nodes are displayed as circles. The tracing is difficult to verify from this view. (b) The 3D view reveals that the tracing consists of two almost parallel axonal branches connected by a bridging edge (red) at the arrow, which is to be removed. (c) 3D close-up of the region around the false connection. (d) Resulting two separate branches (colored differently for visualization purposes) after deletion of the selected edge and node-to-point *conversion of its defining nodes.*

Case 3: *Removal of false segmentations* (Fig. 4.6). When all neurites in a particular region have been identified, falsely segmented fragments must be removed. In the present example, the *NeuroMorph* algorithms accept oversegmentation to ensure that no faintly stained axons are falsely discarded. The *Lasso* tool provides a dedicated mode (activated using a modifier key) to select only connected subgraphs that are entirely contained within a user-drawn polygon. This allows to select all small artifacts in the neighborhood of a reconstructed neuronal process at once and to delete them, without affecting the larger structure.

Case 4: *Removal of false connections* (Fig. 4.7). Occasionally, nearby branches may have been falsely connected by the tracing algorithm. Figure 4.7 illustrates how such situations are resolved. In the present example, the *NeuroMorph* algorithms created tracings whose spatial structure is difficult to verify from the 2D MIP image (Fig. 4.7a).

The 3D view (Fig. 4.7b, c) reveals that the structure consists of two parallel axonal branches connected by a 'bridge'. Such bridges originate from limited resolution and are biologically implausible – they could result in loops in the neuronal tree – and must therefore be removed. To correct the false connection the user selects the bridging edge and removes it (Fig. 4.7d). After deletion of the edge, the defining nodes are no longer branching nodes. These intermediate nodes are removed using the *node-to-point* operation, joining its two incident edges (Fig. 4.7d).

4.4.1.3 Alignment and proof-editing across multiple image stacks

An important aspect of reconstructing complete 3D neuron morphologies is the merging of tracings obtained from multiple image stacks (e.g. consecutive brain sections). Here, tracings containing axonal branches from thalamocortical axons in eight consecutive brain sections were proof-edited as described above and then merged into a single *SpatialGraph* using the *CreateSpatialGraphStack* module (see Sec. 4.3). The tracings obtained from each image stack were translated in the z-direction, i.e. perpendicular to the cutting plane, such that their bounding boxes adjoin.

The *Align* toolbox (Fig. 4.2a) in the FE was then used to rigidly align the subgraphs corresponding to each section tracing. The automated algorithm determined the correct transformation for all tracings in the 8-section data set. After alignment, the branches were connected across the section boundaries using the *Splicing* operation. The alignment and splicing process is regarded as the final quality control, as tracings of branches passing through multiple brain sections can be checked for continuity.

The final tracing result is scaled in the z-direction to match the vibratome-defined thickness of the brain sections (i.e. $8 \times 50\,\mu m = 400\,\mu m$), compensating for potential tissue shrinkage. Further, smoothing along the z-direction is applied (N=9, see Sec. 4.3) to remove staircase artifacts in the skeletons due to anisotropic voxel sizes (i.e. $0.184\,\mu m$ in x/y, and $0.5\,\mu m$ in z). As a result, axonal branches of more than $1\,cm$ path length, including 31 branching points, were extracted from this example data set.

4.4.2 Inter-user variability of proof-editing in vivo-labeled neurons

The proof-editing of tracings obtained from individual image stacks, as well as the alignment and interconnection of tracings across image stacks, may introduce inter-user variability to the final 3D neuron reconstructions. Thus, we validate the above described tasks involving the FE by investigating the reproducibility of the final tracing result. To do so, five users proof-edited, merged and aligned the 8 tracings of the example data set. The users had varying degree of experience. One expert user has been involved in the development of the FE, two experienced users have been using the FE for approximately 12 months and two novice users only received a short introductory training.

Table 4.1 shows a comparison across users of the reconstruction results after proof-editing and merging (including smoothing and z-scaling) the 8 input tracings. As a first assessment of the inter-user variability, we compared the number of reconstructed

	U1	U2	U3	U4	U5	Mean	SD	% of mean
#Branches	20	19	20	21	19	20	0.84	4.2
#Branch points	31	31	31	32	30	31	0.71	2.3
Length (μm)	10762	10953	11053	11313	11153	11047	207	1.9
Time intra (min.)	344	209	118	130	405	241	129	53
Time inter (min.)	25	22	20	17	24	22	3.2	15

Table 4.1: Comparison between 5 users (U1–U5) of the morphological properties of the final reconstructions. 'Branches' are edges in the SpatialGraph. 'Time intra' is the accumulated time in minutes required for proof-editing all eight input tracings. 'Time inter' is the time required for inter-section proof-editing, i.e., merging, alignment, interconnection, scaling, and z-smoothing. U1 and U5 are novice users, U2 and U3 are experienced users, U4 is an expert user.

axonal branches (i.e. edges in the *SpatialGraph*) and branch nodes. The results of the 5 users were essentially identical, given these coarse measures (i.e. 20 ± 0.84 branches, and 31 ± 0.71 branching nodes, mean±SD). Second, we investigated whether the extracted branches were similar in path length. We found that the inter-user variability in path length was surprisingly small (SD=2% of the mean) for the present example data set of axonal branches from an *in vivo* labeled thalamocortical neuron.

While we could not observe any differences in tracing reliability across users, the time required for proof-editing deviated substantially with experience. Novice users (U1 and U5) need 2–3 times longer (344 and 405 minutes, respectively) than well-trained users (U2 and U3; 209 and 118 minutes, resp.). The difference between well-trained and the expert user (U4; 130 minutes) was less pronounced. In addition to the coarse comparison, we investigated the agreement between different users on branch trajectories (see Sec. 4.3). The user agreement of a single example section (S01) and of the final result after alignment, interconnection, smoothing and z-scaling (S01–S08) is shown in Figure 4.8a,b and Figure 4.8c–f, respectively. To quantitatively describe the user agreement of the proof-edited tracings, we computed the accumulated length of all edge segments of each reconstruction that was agreed upon by the majority of the users (i.e. at least two other users found a corresponding segment). This length, expressed as a percentage of the total length, is computed for all users and for each of the eight tracings (Table 4.2).

We find that, on average, between 85.7% and 94.5% of the tracings from individual image stacks are agreed upon by at least 3 users. The larger values of the standard deviation of particularly S02, S03 and S04 are mainly due to U5, whose proof-edited tracings have a relatively large fraction of branches not retained by the other users. These numbers have to be interpreted with caution. Inexperienced users (e.g. U5) may pursue a conservative approach, by retaining fragments that cannot doubtlessly be identified as foreground and by revisiting them during the inter-section proof-editing phase, where they can be judged within the context of the other brain sections. Consequently,

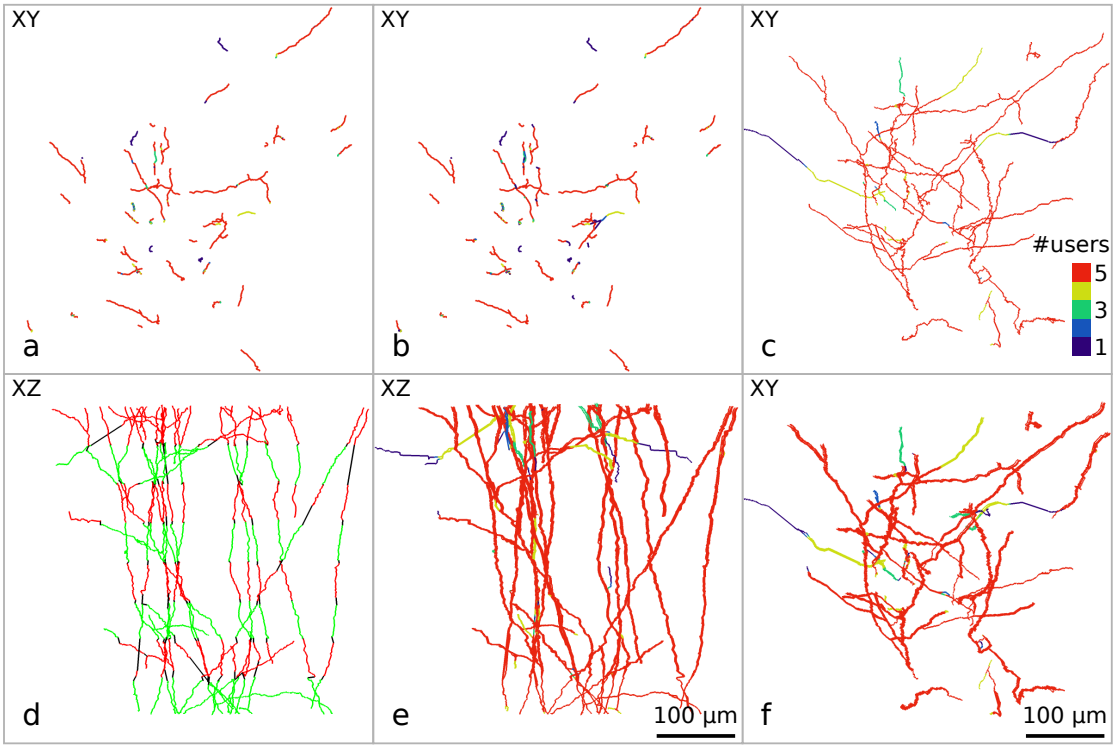

Figure 4.8: User agreement between reconstructions. (a) Reconstruction of section S01 created by one of the users. Each edge point is colored by the number of users that found a corresponding point. (b) S01 reconstruction of all 5 users superimposed. (c) Reconstruction of all 8 sections by one of the users. (d) XZ-view of the reconstruction result of a single user showing the original sections (odd sections in green, even in red) and the interconnecting segments (black). (e) The results of all users superimposed (XZ-view). The bottom section remains untransformed, resulting in a virtually exact overlap between all users. Differences in alignment accumulate towards the top of the stack, resulting in minimally diverging branches. (f) XY-view on superimposed results of all users. Maximum sampling distance for single section (S01) comparison: $D = 0.25\,\mu m$; maximum correspondence distance: $R = 0.5\,\mu m$ (voxel length in z-direction). For comparison of entire stack: $D = 2.5\,\mu m, R = 5\,\mu m$.

	U1	U2	U3	U4	U5	Mean	SD
S01	95.6	96.1	97.0	96.5	87.5	94.5	3.5
S02	91.1	92.3	95.8	85.8	63.6	85.7	11.5
S03	95.6	94.3	96.4	98.7	64.6	89.9	12.7
S04	92.3	88.8	97.1	96.2	62.5	87.4	12.8
S05	88.5	96.8	96.9	95.5	85.5	92.6	4.7
S06	92.9	84.9	98.9	91.1	82.9	90.1	5.8
S07	96.7	92.6	96.4	94.3	88.2	93.7	3.1
S08	97.1	83.1	97.6	99.0	87.4	92.8	6.3
S01–08 (R=1 μm)	59.5	69.9	24.3	77.7	70.2	60.3	18.9
S01–08 (R=5 μm)	100.0	99.6	98.8	97.4	98.4	98.8	0.9

Table 4.2: Fraction of length (%) found by three or more users for each section after intra-section proof-editing and the final reconstruction (S01–08) after inter-section splicing ($R = 0.5\,\mu m$, unless stated otherwise; $D = R/2$).

retaining 'false positive' branches is not necessarily wrong, as long as they are removed during inter-section proof-editing.

Thus, we applied the same procedure for the completed reconstructions comprising the eight aligned and interconnected tracings (Table 4.2, S01–08). We find that for a small correspondence distance ($R = 1\,\mu m$) on average 60.3% of the total axon length is agreed upon by the majority of users. To investigate whether this relatively small fraction is due to missing branches or differences in alignment, the correspondence distance was increased to $5\,\mu m$, resulting in an average agreement of 98.8%. Thus, virtually all branches had counterparts and the large fraction of unmatched branches (i.e. at $R = 1\,\mu m$ resolution) must originate from small differences in alignment, as illustrated in Figure 4.8.

In a previous study (Dercksen et al., 2009) we showed that the automated alignment method yields results comparable to those produced by an expert user when aligning the same data sets. However, in the present example, we compare the alignment between data sets that are slightly different due to inter-user differences after proof-editing tracings of each image stack. All sections in all five data sets could be successfully aligned using the default parameters (distance threshold $d = 10\,\mu m$ and weighting factor $\alpha = 0.25$, see (Dercksen et al., 2009)). Differences in rotation and translation between users are listed in Table 4.3. We find that these differences are small. Rotations differ by at most 0.33 degree, translations by at most $4.04\,\mu m$ and $4.17\,\mu m$ in x and y, respectively.

In summary, comparison of the proof-editing results produced by five different users indicates a high degree of accuracy of the final morphology. First, coarse comparison yielded only small differences in the number of branches (∼1 error per 20 branches), branching nodes (∼1 error per 31 branching nodes) and total length (∼$19\,\mu m$ error per $1\,mm$ axon, see Table 4.1). Second, agreement in trajectories after proof-editing varied between 85.7% and 94.5% across individual image stacks (Table 4.2). Third, differences

	$R(^\circ)$		$T_x(\mu m)$		$T_y(\mu m)$	
	max-min	SD	max-min	SD	max-min	SD
S02	0.11	0.04	1.66	0.58	2.14	0.71
S03	0.03	0.01	0.44	0.18	0.17	0.06
S04	0.25	0.09	2.49	0.81	3.85	1.27
S05	0.33	0.13	3.94	1.52	4.17	1.57
S06	0.13	0.05	1.56	0.60	1.44	0.54
S07	0.27	0.09	1.36	0.43	3.68	1.32
S08	0.13	0.05	4.04	1.44	3.46	1.22

Table 4.3: Inter-user differences in alignment. Each row lists the difference between maximum and minimum rotation angle (R), translation in x- and y-direction (T_x, T_y) of each section with respect to its preceding section (S01 remains untransformed), as well as the standard deviation (SD) of all 5 users.

in alignment are small, i.e. $\sim 4\,\mu m$ (Table 4.3). Consequently, the trajectories of the final reconstructions across multiple image stacks are very similar: on average 98.8% of the reconstructed trajectories were identical across users, using a correspondence radius of $5\,\mu m$. No significant differences were measured between expert and novice users. Specifically, the consensus length was computed for two groups consisting of the three most (U2, U3, U4) and three least experienced (U1, U3, U5) users, respectively. The average consensus length (i.e., at least 2 of 3 users agreed) as a percentage of the total length was 98.7% for the less experienced and 98.8% for the more experienced group (using the correspondence distance of $R = 5\,\mu m$). However, the amount of manual labor required for proof-editing automated tracings decreases with experience from ~6.5 to ~2 hours per centimeter axon.

4.5 Discussion

4.5.1 Applicability and inter-user variability of the Filament Editor

The FE combines various visualization, selection and modification functionalities that allow interactive proof-editing and analysis of 3D neuronal tracings within an easy-to-use and intuitive GUI. Using an example dataset of *in vivo* labeled thalamocortical axons from eight consecutive large, high-resolution image stacks, we illustrated four general situations that may occur during proof-editing any sparsely labeled morphology data. The relative abundance of each of the four use cases will however strongly depend on the image and labeling quality, as well as on the accuracy of the automated or manual tracing methods.

The example data set used here, can be regarded as one of the most challenging cases for reconstructing complete and accurate 3D neuron morphologies. First, the neurons were labeled in rat thalamus *in vivo*, whereas its axonal branches were imaged

within vibrissal cortex, about $3\,mm$ away from the recording site. Thus, in addition to faint staining caused by thin axonal diameters, diffusion of the tracer (i.e., biocytin) along centimeters of axon may have further decreased signal-to-noise ratios of terminal branches. Second, due to the large axonal innervation volume, here $2\,mm \times 2\,mm \times 1\,mm$, the tissue was imaged using a brightfield microscope (at a resolution at the diffraction limit of light) at the cost of contrast and axial resolution. Any fluorescent microscope system of superior contrast and resolution (e.g. 2-photon) would require impractically long imaging times, compared to the ∼24 hours required using the present system (Oberlaender et al., 2009; Oberlaender et al., 2007).

Finally, because of the faint labeling and limited contrast, the automated tracing algorithms (i.e. as implemented within the *NeuroMorph* pipeline) accept oversegmentation to guarantee that all axonal fragments are reconstructed, at the cost of picking up also background structures. The completeness of detecting and tracing all axonal fragments by the *NeuroMorph* system has been validated against manual results generated by human expert users previously (Oberlaender et al., 2007). Consequently, because the FE allows unambiguous proof-editing of the thalamocortical axons presented, datasets of higher contrast, higher resolution or less background will certainly allow for an even faster and equally reliable proof-editing of complete 3D morphologies using the FE.

Using a challenging dataset, we illustrated that the combination of 3D image data (e.g. as a 2D MIP) with (i) 3D tracings (edges are rendered as polylines, nodes as spheres, points as squares), (ii) semantic labeling, (iii) 3D viewing (i.e. camera rotation, translation and zoom), as well as (iv) 3D selecting, hiding and editing is a convenient, intuitive and fast approach to extract reliable 3D morphologies from large sparsely labeled images. Here, about 0.5 terabyte of image data, containing more than one centimeter of axonal fragments could be proof-edited with 98.8% accuracy within 2–6 hours, the proof-editing time depending on the experience of the operator.

4.5.2 The Filament Editor facilitates proof-reading in vivo datasets

In contrast to existing neuron proof-reading tools (see Sec. 4.2), the FE has been designed to meet the specific demands of *in vivo* labeled datasets. The general challenges for validating neuronal tracings from *in vivo* data arise from the (i) the large innervation volumes of individual axons (e.g., $\sim 12\,mm^3$ for L5 slender-tufted pyramidal neurons (L5st, see Figure 4.3b) in rat vibrissal cortex (S1) (Oberlaender et al., 2011)), (ii) the complexity of the axonal arbor (e.g., $86.8 \pm 5.5\,mm$ path length, 216 ± 35 bifurcation points for L5st in S1) and (iii) axonal diameters as thin as $100\,nm$. Consequently, validating *in vivo* tracings can be considered as at least one order of magnitude more complex (i.e., in terms of imaging volume, axonal path length and topology) compared to *in vitro* data from the same species, brain region and cell type (e.g., L5st axon length in S1 from *in vitro* tracings was reported as $7.8 \pm 2.5\,mm$ (Frick et al., 2008)). These challenges are for example illustrated by the fact that, at present, the NeuroMorpho.org repository comprises 5,405 neocortical neurons from various species, but only 3% (188 neurons) of them were labeled *in vivo* and only 0.4% (21 neurons) contain both, recon-

structed dendrites and axons. Moreover, none of these 21 neurons were reconstructed at a magnification of 100× (i.e., as was the present example dataset), which may be critical to reliably detect all axonal projections (Oberlaender et al., 2009).

The Filament Editor provides proof-editing strategies to overcome the challenges described above. First, we incorporated 3D visualization and 3D editing routines that remain functional for imaging volumes and tracing complexities beyond typical *in vitro* datasets. For example (i) the data structure was designed to render and edit large filament data sets, a consequence of the complexity of *in vivo* labeled morphologies and of oversegmented automated tracings results, while maintaining interactive usability. (ii) The selection system allows the user to efficiently highlight the desired part of the tracing based on location (single click, lasso selection), connectivity (connected component), or semantic label for editing or visualization. (iii) Selectively hiding parts of the tracing allows focusing on specific regions of large structures without visual clutter. (iv) As shown in the results, the semantic labeling feature enables position-dependent labels, which visually support user decision making on what neuronal fragments are to be connected or discarded. Second, we integrated alignment and splicing functionalities to merge multi-section datasets, allowing for a final quality control by checking continuity of tracings across brain sections. Again, the semantic labeling eases across-section editing by visualizing edges from different sections in different colors.

Other unique features of the Filament Editor include simultaneous display and quantitative comparison of tracings for assessing inter-user variability, as well as morphometric analysis with respect to 3D anatomical structures. Taken together, the FE will help increasing the so far negligible number of validated 3D neuron tracings from *in vivo* preparations.

4.6 Conclusion

We presented the Filament Editor (FE), a software toolbox integrating components for proof-editing neuron tracings in 3D, across-section alignment and morphometric analysis. The effectiveness of the FE was demonstrated on the example of *in vivo* labeled axonal branches from multiple brightfield image stacks. The FE addresses a clear need for efficient and effective proof-editing, advancing the possibilities for high-throughput reconstruction of accurate and complete 3D neuron morphology. Altogether, the FE facilitates quantitative neuroanatomical studies from *in vivo* labeled data (see Ch. 6), as illustrated on the examples of determining cell types (Oberlaender et al., 2012), axon projection patterns (Oberlaender et al., 2011), plasticity during sensory deprivation (Oberlaender et al., 2012) or simulations of sensory-evoked signal flow (Lang et al., 2011).

5 Assembly of neural networks and visual analysis of synaptic connectivity

5.1 Introduction

Investigation of the cytoarchitectonic structure and synaptic connectivity of the rat barrel cortex, or cortical brain regions in general, is important to gain insight into signal flow between cell types and cortical columns. This potentially yields an understanding of how sensory input is processed and ultimately leads to certain behavior. In order to study the barrel cortex and its synaptic connectivity, an anatomically realistic model is to be created. The model represents the neuron population comprising the barrel cortex, or subnetworks therein. It is created using a reverse engineering approach, integrating different types of detailed anatomical data, including spatial neuron distributions (see Ch. 2) and 3D morphology reconstructions (Ch. 4). Based on the spatial structure of the neuron population, the synaptic connectivity between (groups of) neurons is to be estimated.

The resulting model is an explicit 3D geometrical description of a neural network, annotated with metadata, such as the cortical columns containing the cells, cell types, and subcellular structures (soma, axon, dendrite). As such, the model contains all information to compute the synaptic connectivity between any two neurons or groups of neurons, based on structural axon-dendritic overlap (Peters' rule (Peters, 1979)), yielding estimated connectivity at the subcellular, cellular and population scale.

This wealth of connectivity information is however nontrivial to access, due to the resulting size of the models (∼0.5 million neurons for the entire barrel cortex) and number of connections. Therefore, a visual analysis tool is required that lets the user explore the data, and supports an iterative, drill-down workflow to answer increasingly specific questions. Particular neuroscientific questions of interest that could be answered based on the information contained in the model are:

Q1 Where does a neuron or group of neurons obtain input from, i.e. to which presynaptic cells is it connected? Or, conversely, where does a presynaptic group project to?

Q2 How does thalamic innervation differ between columns and between cell types in the columns?

Q3 How are synapses distributed on the postsynaptic cell? Can cell-type-specific clustering of synapses at particular locations be identified?

Q4 How large is the input of surrounding columns compared to intra-column input for different cell types?

Answers to these questions would help to identify local (intra-column) and long-range (inter-column) microcircuits involved in processing of tactile information obtained from single or multiple whiskers. Furthermore, at the subcellular level, cell-type-specific distributions of synapses may be directly compared to functional imaging data (Varga et al., 2011) to understand how cell function relates to its structural properties.

In the following, first the reverse engineering method for assembling neural networks (Dercksen et al., 2012) is described in detail, including the estimation of synaptic connectivity. The approach has been applied to model connectivity in networks representing two neuron types (VPM and L4ss (Lang et al., 2011)), the excitatory neuron population in a cortical column (Oberlaender et al., 2012), and the entire barrel cortex, including their VPM input. Here, first the assembly of the model representing the entire barrel cortex will be described; the other networks have been created in a similar manner, but with less cortical columns/cell types. Second, a framework (Dercksen et al., 2012) is presented for the visual analysis of synaptic connectivity in such networks across multiple scales, which could be used to answer the neuroscientific questions above.

5.2 Related work

The reverse engineering approach to establish an anatomically realistic neuron population has recently been used to model and analyze the structure (but not the connectivity) of the rat hippocampus (Ropireddy et al., 2012). Burysjuk et al. (Borisyuk et al., 2011) use a similar approach to model a tadpole spinal cord and its connectome, albeit using an axon growth algorithm instead of anatomical reconstructions. No literature was found, however, on modeling of the rat barrel cortex.

Lin et al. (CY Lin et al., 2011) present a tool for issuing spatial queries on a database containing reconstructed *Drosophila* neurons registered into an atlas. Although this provides information about neural connectivity based on spatial proximity, it does not permit quantitative conclusions about connectivity strength as this requires realistic estimates of number and type of neurons.

Brain connectivity can be studied at different scales, ranging from the macroscale (connections between brain regions) to the microscale (connections between individual cells). At the mesoscale, connections between medium-sized neuron populations, e.g. cortical columns, are studied (Sporns et al., 2005). Independent of scale, the connection pattern can be represented by a graph, which, in the context of connectomics, is often represented by a connectivity matrix. The nodes represent anatomical units, e.g. synapses, neurons, groups of neurons or brain regions, depending on the scale. The links represent connections between the nodes and are either binary, indicating the presence of a connection, or weighted, quantifying the connection strength.

The most appropriate way to obtain connectivity information depends on the scale (Pfister et al., 2014). At the macroscopic level, fiber tracking of axons in diffusion tensor

imaging (DTI) is a technique to identify connections (Hagmann et al., 2010). At smaller scales mainly electron- or light-microscopic techniques with appropriate stains are used (Kleinfeld et al., 2011).

Brain networks can be analyzed mathematically, using methods and concepts from the fields of graph/network theory and topology (Kaiser, 2011), and/or visually. Creating easily interpretable visualizations of brain networks is non-trivial, as such a network is often a complex graph embedded in 3D. For example, axonal tracts obtained by DTI fiber tracking can be directly rendered in 3D. However, their number and density causes clutter and hampers interpretation, requiring techniques such as abstraction and filtration (Jianu et al., 2012) to create effective visualizations.

Being a graph, the connection matrix can be visualized using common graph visualization methods. The most common method is direct visualization by a heat map (Fig. 5.4c), where the connection strength between each pair of nodes is color-coded (Bassett et al., 2011). The main disadvantage is that this visualization provides no spatial context. To aid their interpretation, heat maps can be complemented by a 3D visualization of the anatomical elements, represented by the nodes. Often these elements are brain regions, e.g. obtained by segmenting MRI data. They are displayed in 3D using surfaces (see e.g. (Bassett et al., 2011; Gerhard et al., 2011)) or volume rendering (Y Guo et al., 2012), thereby color-coding some network property of the nodes. Alternatively, node-link diagrams based on a force-directed layout algorithm (Hagmann et al., 2008) or connectograms (Irimia et al., 2012) are used. These, however, suffer from clutter when network complexity increases. A disadvantage of multiple complementary views is that switching incurs mental effort.

Concluding, at present anatomically realistic 3D models of the rat barrel cortex at cellular resolution do not exist. In addition, general tools for analyzing connectivity information, such as the heat map or variations of node-link diagrams, have severe drawbacks. Dedicated tools for analyzing multiscale connectivity information between neuron populations within and across cortical columns are currently lacking.

5.3 Assembly of cortical neural networks

5.3.1 Input data

The following data forms the input for the network assembly:

- 3D Soma distribution. The soma distribution is given by a 3D density field; the value of each $50 \times 50 \times 50\,\mu m^3$ grid cell represents the number of somata that it contains. This (preliminary) data is obtained by automatic counting of somata in confocal images, as in (HS Meyer et al., 2010).

- Column properties. The cortical columns, together with the pia and white matter surfaces, form the geometric reference system of the barrel cortex (R Egger et al., 2012). The barrel cortex model (Dercksen et al., 2012) consists of 24 cortical

columns. Each column is described by its center, axis (approximately perpendicular to the pia), radius, and a label, derived from their position in the barrel field (see Fig. 1.1c). Each position is identified by a row (A–E) and arc (1–4), with additional columns α–δ in front of the first arc. These geometrical properties and metainformation are important for neuron positioning and model analysis, as described below. For the single-column model (Oberlaender et al., 2012) only the D2 column was used.

- Cell type metadata. Currently, 10 cell types are modeled: 9 excitatory neuron types in the barrel cortex, and the VPM neurons in the thalamus (axons only). Each cortical cell type is named after the layer in which their somata are predominantly located, with an optional subtype indication, e.g. Layer 4 pyramidal (L4py) neurons. Provided is also the number of presynaptic contact sites (boutons, β) per μm axon and the number of postsynaptic contact sites (spines, σ) per μm dendrite. At present, these are given constants (0.33 and 0.5 resp. (Lang et al., 2011)), and identical for all cell types.

- 3D morphologies of all cell types. Approximately 100 dendritic and ∼60 axon morphologies of 10 different cell types have been reconstructed from brightfield microscopy images (Bruno et al., 2009; Oberlaender et al., 2011; Oberlaender et al., 2012), a representative sample that captures the anatomical variability of the excitatory cell types in the vibrissal cortex (Oberlaender et al., 2012). They have been registered into all 24 columns (R Egger et al., 2012), resulting in ∼3700 morphologies.

- Cell type mixture boundaries. The mixture of cell types varies with cortical depth. To correctly populate the modeled volume, including the septum, the boundaries between different cell type mixtures have to be defined for the entire region. In order to model the correct mixture of cell types at each location, a set of curved, nearly-parallel surfaces bounding different mixture types and spanning the entire barrel field is computed as follows. First, the mixture boundaries within a representative column (e.g. C2) are determined (Oberlaender et al., 2012). To achieve this, the position and cell type is determined for all somata of the dendritic morphologies in the column. Then, the soma positions are projected onto the column axis and binned into 50 μm intervals. Finally, the cell type mixture for each interval is computed. When neighboring intervals have different mixtures, a boundary point is defined between them on the column axis. The resulting set of points is transferred to all other column axes, applying a scaling to reflect the differences in column length. A surface between corresponding boundary points on all column axes is created by Delaunay triangulation. Fig. 5.1a shows the resulting surfaces.

- Orientation field. The apical dendrite of pyramidal neurons in the cortex is usually oriented perpendicular to the pia (for cell types that lack an apical dendrite the neuron orientation is defined as the direction of the main axon towards the

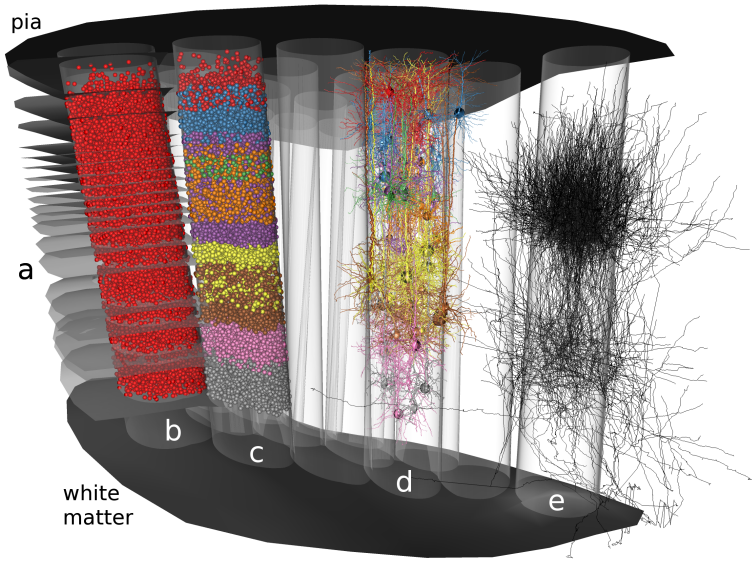

Figure 5.1: Establishing the barrel field neuron population. For illustrative purposes, each step of the population assembly is shown in a different column and for a small number of morphologies only. a) Mixture region boundary surfaces (clipped). b) Soma distribution inside a column. c) Cell type assignment. d) Dendrite reconstructions, colored by cell type. e) VPM axon reconstructions.

white matter). To retain this property when populating the neural network, the orientation of morphologies has to be adjusted. This local apical dendrite orientation is computed by interpolating the axis direction of the 3 nearest columns at each point. To speed up computation, this local axis direction is precomputed by sampling on a uniform grid A ($50^3\,\mu m^3$ voxels).

5.3.2 Establishing the neuron population

Based on this data, the model of the barrel cortex is created as follows. First, a realization of soma positions is computed that satisfies the given soma density field, see Fig. 5.1b.

Second, each soma is assigned a cell type (Fig. 5.1c). The mixture region containing a soma is determined by finding the intersection with the boundary surfaces above and below the soma along the direction of the axis of the nearest column. The soma is randomly assigned a cell type satisfying the mixture for this region.

Third, dendrite morphologies are placed at the computed soma positions (Fig. 5.1d). For each soma, a dendrite morphology, which soma position is close ($< 50\,\mu m$) to the computed soma position, of the assigned cell type and column that is closest to the

soma is picked. This morphology is transformed as follows: 1) rotation around the column axis, such that the orientation with respect to the column center is retained; 2) translation of the reconstructed soma to the new position; 3) scaling along the column axis so the dendrite remains within the column; 4) rotation around the soma, such that the orientation towards the pia surface is retained; the new orientation is looked up in the axis field A.

Finally, reconstructed axons of neuron types for which dendrites are placed are duplicated such that their number equals the number of dendritic reconstructions. VPM axons are also duplicated; their number is determined by counting the somata in their respective thalamic region (Oberlaender et al., 2012).

The result of the final step is a population P of axon and dendrite morphologies positioned in 3D space. Each morphology has a cell type. Given the definition of the column cylinders, for each dendritic morphology its nearest column is computed and whether its soma is inside or outside this column. This allows us to define groups of morphologies: a group $G^{(ct,\,col)}$ consists of all neurons of cell type ct having their soma inside column col. In addition, we define the groups $G^{(VPM,\,col)}$ consisting of VPM cells, whose somata are located in the thalamus instead of the columns, but have axons projecting mainly into their respective columns col.

5.3.3 Estimation of synaptic connectivity

Synapse numbers are estimated based on structural overlap between axons and dendrites, commonly referred to as Peters' rule (Peters, 1979). Specifically, the number of synapses is determined by dividing the local number of presynaptic contact sites (boutons) on the axons among the local number of postsynaptic contact sites (spines) on the dendrites (see Fig. 5.2). To this end, the volume V containing all neurons is partitioned into a uniform grid of volumetric elements (of grid cell size $50^3\,\mu m^3$ in our case, related to the registration error of the morphologies into barrel field reference system (R Egger et al., 2012)). Given the bouton and spine densities β, σ, the number of boutons $B(c, \mathbf{x})$ of neuron c within grid cell $\mathbf{x}$ is

$$B(c, \mathbf{x}) = \beta \cdot L_{axon}(c, \mathbf{x}), \tag{5.1}$$

where $L_{axon}(c, \mathbf{x})$ is the axon length of c in $\mathbf{x}$. The number of spines $SP(c, \mathbf{x})$ is, analogously,

$$SP(c, \mathbf{x}) = \sigma \cdot L_{dendrite}(c, \mathbf{x}). \tag{5.2}$$

The number of boutons of an entire presynaptic group is obtained by summing over all neurons in the group:

$$B(G_{pre}, \mathbf{x}) = \sum_{i \in G_{pre}} B(i, \mathbf{x}), \tag{5.3}$$

and for the number of spines of the entire population:

$$SP(P, \mathbf{x}) = \sum_{j \in P} SP(j, \mathbf{x}). \tag{5.4}$$

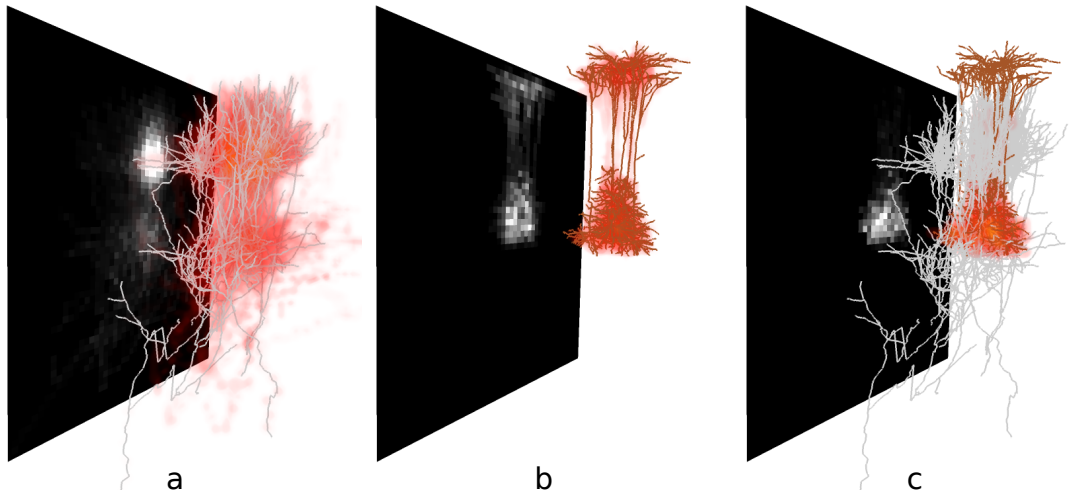

Figure 5.2: Illustrative example of the computation of synaptic connectivity for a small neuron population. (a) The 3D bouton density (red) of a presynaptic neuron groups is estimated by computing the length of the axonal branches (grey) in cubical grid cells, multiplied by the number of boutons per µm axon. The image shows a maximum intensity projection of the 3D bouton density field. (b) The spine density of the dendritic branches of a postsynaptic neuron group is computed analogously. (c) Based on these distributions, the number of synaptic contacts between the pre- and postsynaptic neuron group is estimated. The synapse density is highest in the region with most axo-dendritic overlap.

The number of synaptic contacts $S(G_{pre}, c, \mathbf{x})$ of a presynaptic neuron group G_{pre} with a postsynaptic cell c within grid cell $\mathbf{x}$ is computed as follows:

$$S(G_{pre}, c, \mathbf{x}) = SP(c, \mathbf{x}) \cdot \frac{B(G_{pre}, \mathbf{x})}{SP(P, \mathbf{x})} \tag{5.5}$$

From these single-cell synapse distributions, we can compute different statistical quantities of interest, e.g. the total number of synapses $S(G_{pre}, c)$ of c with a presynaptic group:

$$S(G_{pre}, c) = \sum_{\mathbf{x} \in V} S(G_{pre}, c, \mathbf{x}), \tag{5.6}$$

or the total number of synapses of a postsynaptic group with a presynaptic group:

$$S(G_{pre}, G_{post}) = \sum_{c \in G_{post}} S(G_{pre}, c). \tag{5.7}$$

Dividing by the number of postsynaptic cells $|G_{post}|$ results in the average number of synapses per postsynaptic cell:

$$\hat{S}(G_{pre}, G_{post}) = \frac{1}{|G_{post}|} \sum_{c \in G_{post}} S(G_{pre}, c). \tag{5.8}$$

Computing $S(G_{pre}, G_{post})$ for all combinations of pre- and postsynaptic groups results in a connectivity matrix M, representing the total number of synapses between each pair of groups. This can be done analogously for the matrix $\hat{M}$ consisting of $\hat{S}(G_{pre}, G_{post})$ entries.

5.4 Framework for visual analysis of synaptic connectivity at multiple scales

A framework has been developed that lets the user visualize and extract quantitative synapse connectivity information from a barrel cortex model described above. It can be used, e.g., to answer the neuroscientific questions (Q1–Q4) posed in the Introduction (5.1). The framework consists of multiple coordinated views and follows the Model-View-Controller paradigm. The model consists of:

- neuron population P,
- network metadata: column geometry and cell type properties (e.g., bouton/spine density β and σ, respectively),
- synapse evaluator proxy providing values of connectivity matrices M, $\hat{M}$ and synapse densities for the 3D view,
- selection of pre- and postsynaptic groups $\{G_{pre}\}$, resp. $\{G_{post}\}$,
- and a selected postsynaptic neuron $c \in \{G_{post}\}$.

To ensure interactive response times, the connectivity matrix values are precomputed. The 3D synapse distributions (see Eq. 5.5) are computed on-the-fly using cached values of frequently used fields (population spine density grid $SP(P, \mathbf{x})$ and bouton densities for all groups $B(G_{pre}, \mathbf{x})$).

The framework presents the user the following views:

- Cortical Column Connectivity View (CCCV, see Sec. 5.4.1),
- 3D view showing the synapse density as well as synapse distributions on the dendrites of individual neurons (see Sec. 5.4.2),
- heat map view of a connectivity matrix, because it is a standard visualization (see Fig. 5.4c).

Whereas the CCCV and the heat map visualize connectivity at the scale of cell populations, the 3D view shows synaptic innervation at the subcellular scale. Both the heat map and the CCCV can display either the values of M or $\hat{M}$. The user can explore the connectivity information by interactively defining selections $\{G_{pre}\}$ and $\{G_{post}\}$ for which the synapse information is to be shown in the CCCV. This can be done either in the CCCV or in the heat map; the selection is automatically propagated to the other view (in the heat map, the selection is simply highlighted, as it always shows the entire matrix). In addition, a single neuron of any of the selected postsynaptic groups can be selected, for which the synapse distribution is displayed in detail in the 3D view. Single neurons can be selected by picking from a list, sorted by neuron group. By iteratively modifying the selection in a targeted manner the user can drill down in the data, while increasing insight. The framework is implemented in ZIBAmira (`http://amira.zib.de`).

5.4.1 Cortical Column Connectivity View

To overcome the main disadvantage of the heat map (no spatial reference), we devised a visualization that presents the essential connectivity information within and between cortical columns in a semi-spatial context. In the CCCV, the columns are displayed in 2D as contours (Fig. 5.3a). Their position approximates their relative position within the cortical sheet (Fig. 1.1c), thus creating a spatial reference. A more exact mapping of the cortical sheet to the 2D plane would be conceivable, but for our application this approximation sufficed. In each contour the pre- and postsynaptic connectivity values are displayed as bars. The vertical ordering of the cell types follows the cortical layering and is therefore a rough approximation of the spatial column axes (Fig. 5.3c).

The user explores the connectivity information by interactively specifying selections $\{G_{pre}\}$ and $\{G_{post}\}$ (Fig. 5.3b, c). When a selection has been defined, the bars on the left side of each column show for each selected presynaptic neuron group the sum of synapses this group projects to. Conversely, the bars on the right side display the sum of synapses each postsynaptic group receives from the selected presynaptic groups. The sum of all values on the presynaptic side thus always equals the sum of values on the postsynapic side.

5.4.2 3D view of subcellular synapse distributions

The purpose of the 3D view is two-fold: first, it visualizes subcellular differences in synaptic density by coloring the morphology of a selected postsynaptic neuron c by the local synapse density $S(\{G_{pre}\}, c, \mathbf{x})$ (see Fig. 5.4a). Second, it shows the distribution of synapses on the dendritic branches of c, colored by presynaptic cell type, in order to identify subcellular regions of preferred synaptic innervation for different presynaptic cell types (Fig. 5.4b).

The synapse positions are determined by randomly placing $S(G_{pre}, c, \mathbf{x})$ synapses on the dendrites of c within the $50^3\,\mu m^3$ grid cell $\mathbf{x}$, for each $G_{pre} \in \{G_{pre}\}$. When more

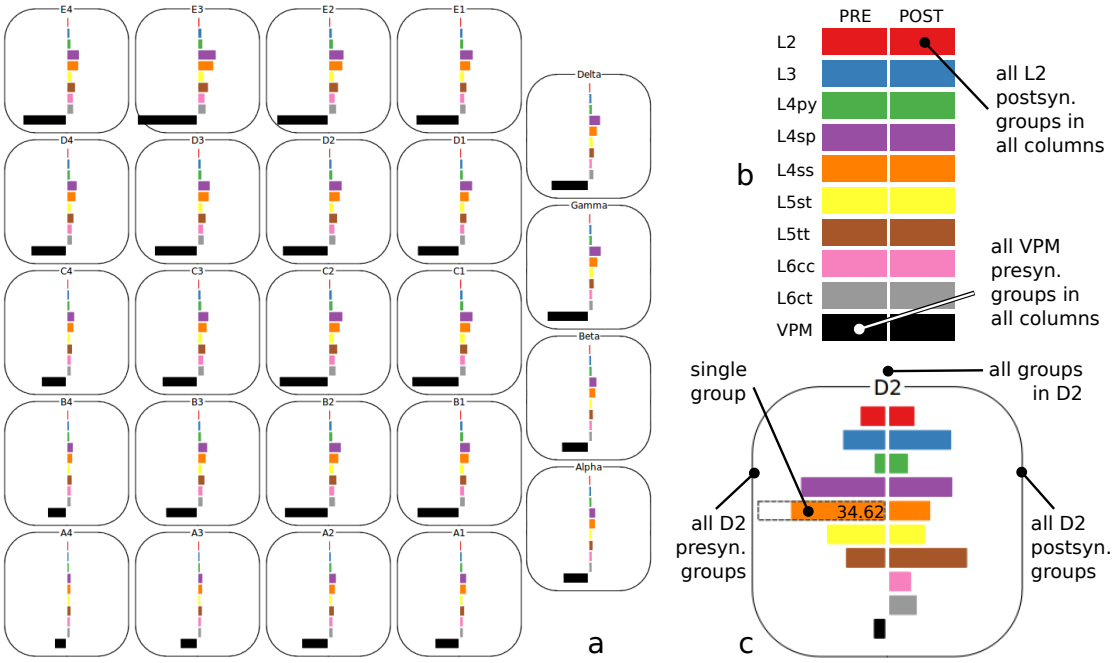

Figure 5.3: Framework for the visual analysis of synaptic connectivity: the Cortical Column Connectivity Viewer (CCCV). (a) The CCCV lays out all columns according to their position in the cortical sheet. The bars show the number of synapses selected that presynaptic groups (left, here: all VPM groups) share with selected postsynaptic groups (right, here: all other cell types). (b) Cell type legend that is displayed as part of the CCCV, and which can be used to make selections by clicking on the marked areas. (c) One example column in the CCCV. The rendered elements can be used to define selections. Here the presynaptic L4ss group of column D2 has been selected (indicated by the dashed line). Multi-selection is achieved by holding a modifier key. Values can be displayed as text (only one value shown for illustration).

detailed information concerning the true distribution along the dendrites becomes available, this can be incorporated.

5.5 Application example

To show how the framework is used to obtain insight into the synaptic connectivity in the barrel cortex model, we apply it to answer the questions posed in Section 5.1.

In order to quantify thalamic input into the barrel cortex (Q1, Q2), we select all presynaptic VPM groups by clicking on the PRE-VPM box (Fig. 5.3b). The total presynaptic number of synapses of these groups is indicated by the black bars in Fig. 5.3a. We observe that the VPM cells corresponding to the central columns and the E-row

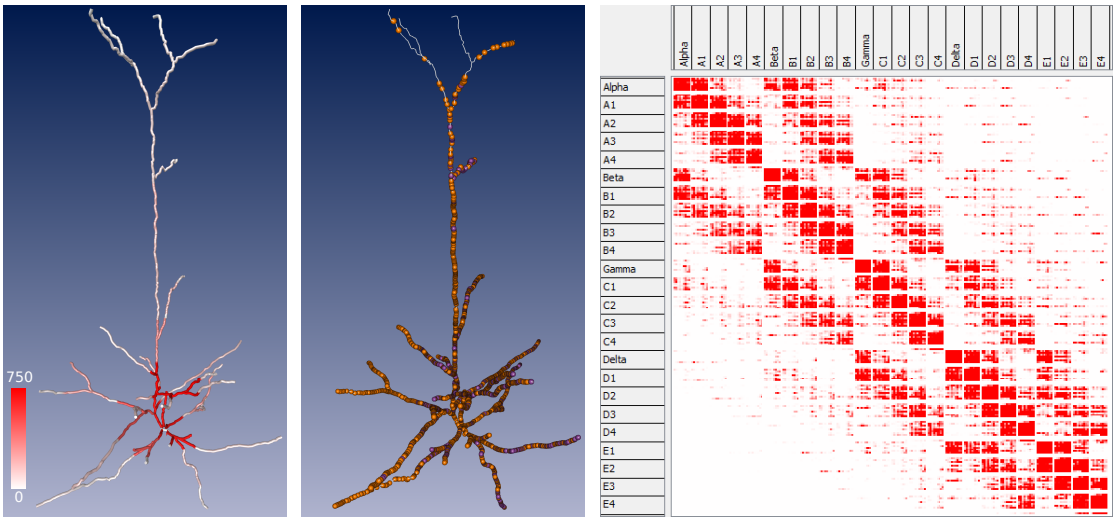

Figure 5.4: Framework for the visual analysis of synaptic connectivity: subcellular synapse distributions and heat map visualization. (a) Synapse density color-coded on L4py dendrites. Each edge point is colored by the number of synapses in the $50^3\,\mu m^3$ *grid cell that contains it (color interpolated between edge points). (b) Synapse positions of presynaptic L4ss and L4sp cells on L4py dendrites. (c) Connectivity matrix visualization by the heat map view.*

provide most input, and that the cells in these columns receive most input. Further we observe that the relative amount of synapses per postsynaptic group is similar for each column: Layer 4 star pyramids (L4sp) and spiny stellates (L4ss) receive most input, whereas L2 and L3 cells hardly receive any (as reported in (Oberlaender et al., 2012)).

To reveal how one of the central columns (D2) is innervated by its corresponding VPM axons (Q1), we select the presynaptic D2-VPM group and all postsynaptic D2 groups (Fig. 5.5a). Again we observe that L4sp and L4ss receive the largest number of synapses from VPM (indicated by the red arrows). To determine whether this is due to the large number of neurons in these two groups, we display the average number of synapses per postsynaptic neuron (Fig. 5.5b) and observe that this is the case, and that among all cell types actually Layer 4 pyramids (L4py) receive most synapses per cell from VPM.

To find out what other cell types a L4py neuron gets input from (Q1), we select all presynaptic groups and the postsynaptic L4py group in D2. Most input is due to L4ss and L4sp (Fig. 5.5c, red arrows). As pyramidal cells have distinct dendritic compartments with short, highly arborized basal dendrites around the soma and a long apical dendrite extending towards the pia, we ask whether there is cell-type-specific clustering of synapse positions on the dendrites (Q3). We therefore switch to the 3D view, select a L4py cell and observe that synapses are densest close to the soma (Fig. 5.6a). The main input types are L4ss and L4sp. Restraining the selection to these types and show-

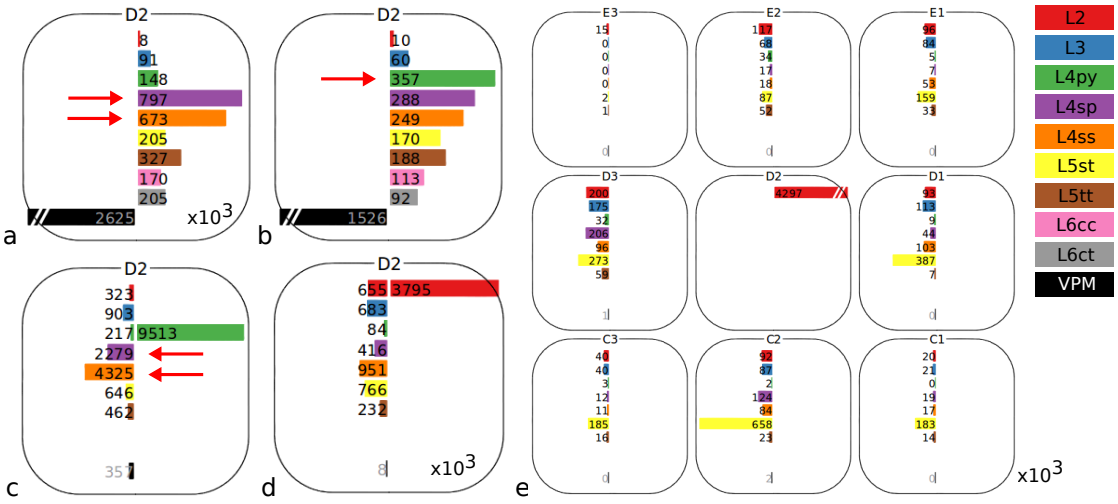

Figure 5.5: (a) Total number of VPM synapses with D2 cell types.(b) Average number of VPM synapses. (c) Average number of synapses that an L4py neuron receives from other cell types in D2. (d) Intra-column L2 innervation (total synapses). (e) L2 input from surrounding columns.

ing the synapse positions (5.6b) reveals that these cell types mainly innervate the basal dendrites. This raises the question what cell types connect to the apical dendrite. We broaden the selection to all presynaptic cell types and zoom in on the apical tuft (5.6c). The L2, L3 and L5 slender-tufted (L5st) cells seem to dominate the synaptic input to the tuft. Indeed, if we look at the L2-L4py synapses only (5.6d,e), we observe that L2 predominantly innervates the apical dendrite.

What cells does L2 receive input from, when not from VPM? Selecting the postsynaptic L2 group in D2 immediately provides the answer. To differentiate between intra-column input and input from surrounding columns (Q4), we first select all presynaptic D2 cell types and observe that L2 in D2 is innervated by ∼3.8 million synapses (Fig. 5.5d). Selecting all presynaptic cell types in the columns surrounding D2 reveals that L2 in D2 receives more input from neighboring columns (∼4.3 million synapses), mainly from L5st (5.5e).

5.6 Discussion

The barrel cortex model described here enables the study of its (sub)cellular architecture region and synaptic connections. The most important assumptions for model validity are that (i) the reconstructed morphologies are a representative sample of the true population and that (ii) the computation of the synaptic connectivity based on Peters' rule is a good estimate. See (Oberlaender et al., 2012) for a more detailed discussion.

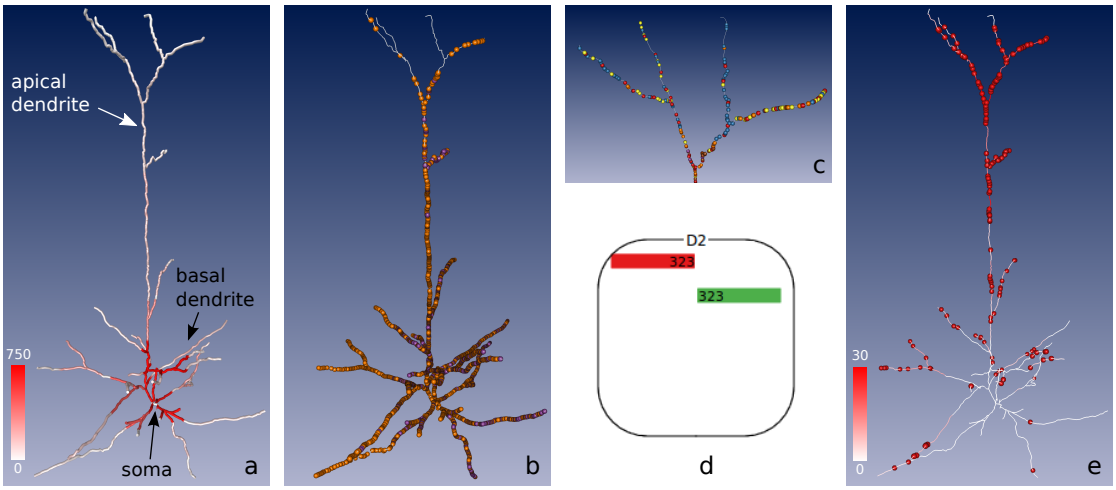

Figure 5.6: (a) Synapse density color-coded on L4py dendrites. (b) Synapse positions of presynaptic L4ss and L4sp cells. (c) Synapses on apical tuft. (d) Number of synapses for the L2-L4py connection in the CCCV. (e) L2 synapse positions.

The information displayed by the CCCV is to a large extent the same as a heat map representation. The main differences are the aggregation of connectivity matrix values when more than one pre- or postsynaptic group is selected, and the semi-spatial context, aiding localization and the investigation of the interaction between columns. Whether it can replace the heat map, which is more generally applicable and currently the familiar standard representation, remains to be seen. However, as they share the same data structure (the connectivity matrix) they can easily be used side-by-side in an integrated framework. The CCCV is not limited to the barrel cortex, but can be used for all regions located on a curved 2D surface, such as the cortical sheet. The size of the brain region that can be studied using the CCCV is, however, limited by the number of columns that fit on the screen. While the framework presented here allows studying connectivity from the level of synapses and individual neurons up to cortical areas containing several dozens of columns, such as S1, larger models spanning more scales and eventually representing entire brains will undoubtedly be developed. Their investigation requires visualization tools that extend the multiscale approach to the entire hierarchy from micro- to macro-scale.

The modeling approach and the visualization framework are considered very useful by domain experts from the Max Planck Florida Institute: “This set of tools allows scientists to investigate structural organization principles at the scale of an entire cortical area with sub-cellular resolution. In particular cell-type- and location-specific connectivity patterns are accessible for the first time, and may be extended and/or compared to more direct connectivity measurements from electron microscopy. The 2D layout of columns and cell types is a great advantage of the CCCV over the heat map representation as

it aids in localization and understanding of column interactions. The various selection options make query specification simple, resulting in short question/response iterations. It is therefore fun to work with."

6 Applications in neuroscience: reverse engineering the rat barrel cortex

The methods and tools described in the previous chapters have been applied to address neurobiological research questions regarding structure and function of neurons and neural circuits. In the following, several published studies are reviewed that benefited significantly from the presented methods and tools.

6.1 Number and spatial distribution of neuron somata in cortical columns

As stated by Meyer et al. (HS Meyer et al., 2010), the number of neurons in a cortical column has previously been estimated by extrapolating measurements of neuron density, based on Sterio's disector method (Sterio, 1984). However, the estimates for neuron density in the rat somatosensory cortex varied by almost a factor of 2: between 48 000/mm^3 (Beaulieu, 1993) and 77 000/mm^3 (Keller and GC Carlson, 1999).

To obtain more precise measurements, required for example for quantitative models of cortical function (Helmstaedter et al., 2007), Meyer et al. (HS Meyer et al., 2010) used the automatic soma detection method described in Chapter 2 to count all neurons in the D3 barrel column, in addition to the manually counted C2 and D2 columns. To this end tangential brain sections (50 μm thickness) were stained with NeuN (Mullen et al., 1992), labeling all neuron somata, and imaged using a confocal microscope. The column boundaries were determined by delineation and extrapolation of clearly discernable areas in L4 in images of stained VPM axons or GAD67 immunolabeled sections.

The number of neurons and their 3D distribution within the C2, D2, D3 and cortical columns could thus be determined with high precision (Fig. 6.1), yielding an average of approximately 19 000 neurons per column. Based on the soma density profile along the column axis, layer boundaries could be determined. Disagreement in neuron density reported in previous studies could thus be resolved by exhaustive counting: the neuron density reported by Meyer et al. ($\sim$78 000/mm^3) agrees with the numbers reported by Keller and Carlson (Keller and GC Carlson, 1999).

Recently, the neuron distribution in the entire barrel cortex and thalamus of four rats has been determined, on the basis of the automatic soma counting method (HS Meyer et al., 2013). This yielded that the number of neurons per column and thalamic barreloid varies substantially within individual animals, particularly an increase in neuron numbers from the A- to the E-row can be observed. The soma density field resulting

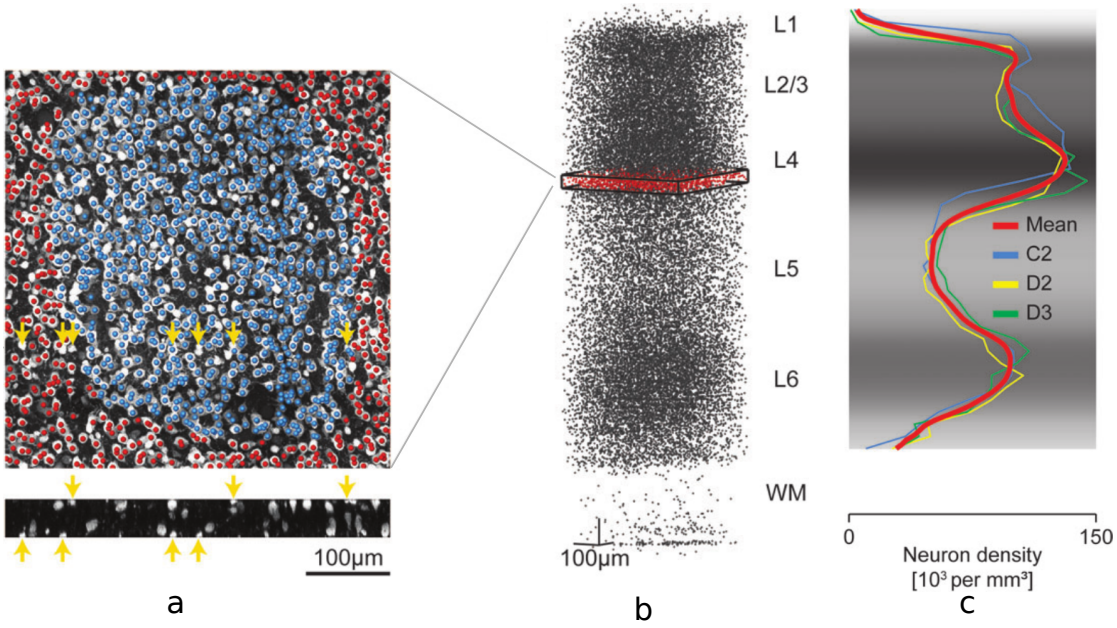

Figure 6.1: Number and distribution of NeuN-labeled cell bodies. (a) Markers representing soma midpoints within (blue) and outside the C2 barrel outline. Yellow arrows indicate cells at slice edges that were not counted as their midpoint was not contained in this section (see orthogonal view in bottom panel). (b) Soma positions in the region comprising the C2 column. (c) Soma density profile of the C2, D2, and D3 columns and their average. Image modified from (HS Meyer et al., 2010).

from this study has been used to create the barrel field neural network model, described in Chapter 5.

6.2 Reconstruction and analysis of neuron morphology

Using the pipeline described in Chapter 4 relatively large numbers of neurons can be reconstructed in great detail within relatively short time. The resulting increase in sample size often increases the amount of evidence for findings involving neuron morphology from merely anecdotal to statistically relevant. In the following, we describe published neuroscientific results that involve detailed reconstructions of dendritic and/or axonal trees, created using the Filament Editor, to illustrate this point.

6.2.1 Classification of dendritic cell types

Oberlaender et al. (Oberlaender et al., 2012) reconstructed 95 dendrite morphologies of excitatory neurons in a cortical column in the rat barrel cortex using the reconstruction

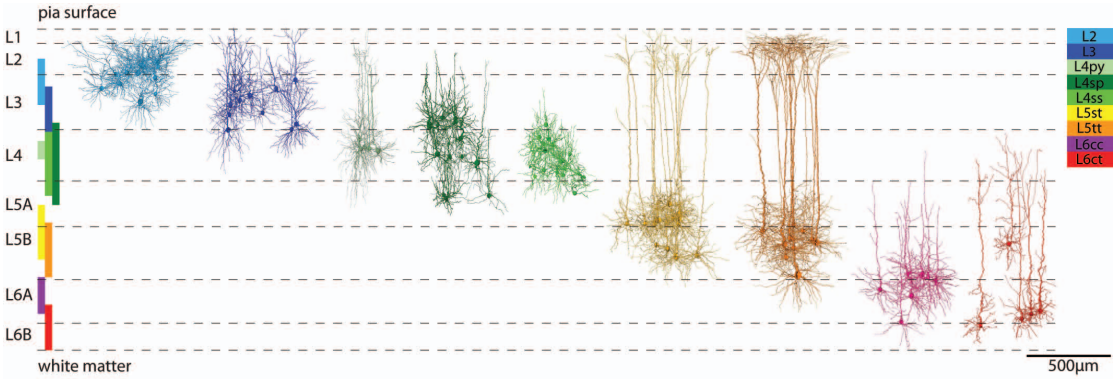

Figure 6.2: Excitatory cell types in a cortical column. Cluster analysis of morphological features identified 9 cell types. The vertical colored bars (left) represent the vertical extent of the cell-type-specific soma locations. These cell type boundaries do not always match the cytoarchitectonic definition of cortical layers, indicated by the dashed lines (adopted from (HS Meyer et al., 2010)). Some cell types intermingle within layers. Image reproduced from (Oberlaender et al., 2012).

pipeline described in Chapter 4. This relatively large sample made it feasible to perform a cluster analysis (Ankerst et al., 1999). Nine different cell types could be distinguished based on 90 morphological features (Fig. 6.2). Most discriminating features are: soma depth from the pia, maximal vertical span of the apical dendrite, and the box volume around all terminal nodes.

6.2.2 Behavioral state-specific pathways involving layer 5 neurons

Cortical layer 5 contains two excitatory cell types, slender-tufted (L5st) and thick-tufted neurons (L5tt). These neuron types respond differently in terms of spiking activity, depending on behavioral state. L5st neurons convey motion and phase information during active whisking, but do not respond to passive whisker touch. In contrast, L5tt neurons increase spiking after passive touch (De Kock et al., 2007; De Kock and Sakmann, 2009).

Oberlaender et al. (Oberlaender et al., 2011) used the morphology reconstruction pipeline and analysis tools described in Chapter 4 to create 3D axonal and dendritic morphologies of L5st and L5tt neurons from *in vivo* preparations, and to analyze their length characteristics and innervation patterns. They found that L5st neurons densely innervate supragranular layers (L1-L3) of large portions of the vibrissal area and beyond (Fig. 6.3a–d). In contrast, the projections of L5tt neurons are less dense and primarily confined to infragranular layers (L5-L6) (Fig. 6.3e–h).

These results give rise to the hypothesis that thick-tufted neurons in the rat vibrissal cortex facilitate object location by active whisking, based on near-simultaneous input from two pathways (Oberlaender et al., 2011). First, input from L5st neurons on the

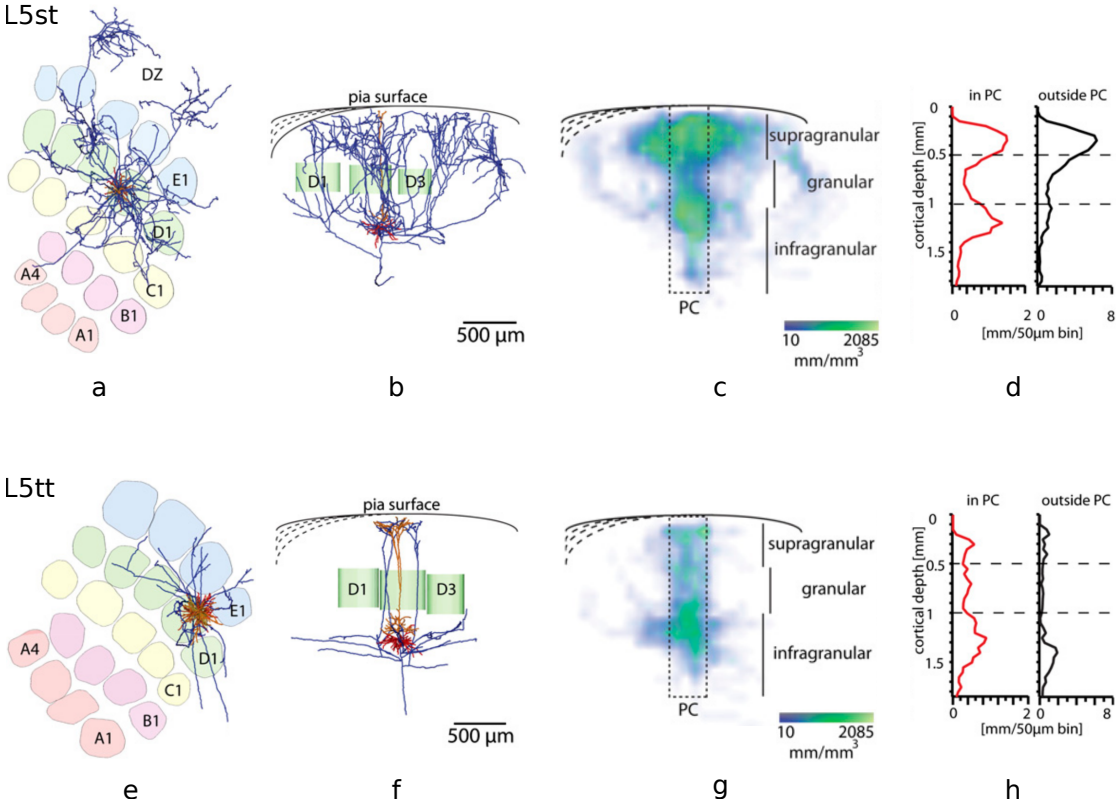

Figure 6.3: Morphological reconstruction and analysis of layer 5 slender-tufted (a-d) and thick-tufted (e-h) neurons reveals cell-type-specific innervation patterns. (a) View on the 3D morphological reconstruction of a L5st neuron. The axonal tree (blue), apical (orange) and basal (red) dendrites are shown with reference to the barrel field. This neuron displays wide lateral spread into multiple barrel columns, surrounding the principal column (PC) and outside the vibrissal area (dysgranular cortex, DZ). (b) Semicoronal view along row D in (a). The lateral projections mainly innervate the supragranular layers. This is quantified by the axon density profile (axon length per mm^3) of all five reconstructed L5st neurons (c). (d) Axon length profile along the vertical column axis reveals two innervation zones. The L5st infragranular innervation zone is restricted to the principal column, in contrast to the supragranular innervation zone, which extends to surrounding columns. Morphological reconstruction of an individual L5tt neuron (e, f) as well as quantified axon density and length profiles of five L5tt neurons (g-h) reveals that innervation remains confined to the principal column and to the infragranular layers of a limited number of surrounding columns. Image modified from (Oberlaender et al., 2011).

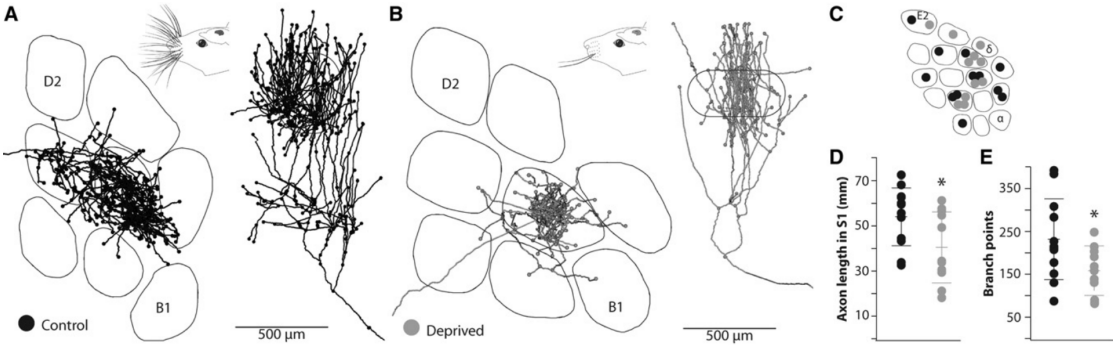

Figure 6.4: (A,B) Reconstructions of thalamocortical axons in control (A) and deprived (B) animals. Axons in deprived animals correspond to trimmed whiskers. Deprived animals had all but two whiskers trimmed. Left: tangential view. Right: radial view. Lines: barrel borders. Dots: branch and end points. (C) Columns targeted by control (black) and deprived (gray) axons. (D) Distributions of total axon lengths within the cortex. Lines represents means ± SD. (E) Distributions of branch points within the cortex. Reproduced from (Oberlaender et al., 2012), with permission from Elsevier.

L5tt apical tuft, carrying whisker motion and phase information, may lock the L5tt dendrite membrane potential to the whisking cycle. Second, the L5tt basal dendrites receive VPM input carrying whisker touch information. Near-coincident input from these two pathways may result in increased spiking activity of thick-tufted neurons and thus enhanced signaling to their subcortical targets.

6.2.3 Thalamocortical plasticity in adult rats

The brain's capacity to rewire is thought to diminish with age. It is widely believed that development stabilizes the synapses from thalamus to cortex and that adult experience alters only synaptic connections between cortical neurons (Oberlaender et al., 2012).

To investigate sensory experience-induced plasticity, Oberlaender et al. (Oberlaender et al., 2012) used the reconstruction pipeline described in Chapter 4 to reconstruct the axonal trees of 23 thalamic VPM neurons projecting into the barrel cortex (Fig. 6.4). Of these neurons 11 belonged to input-deprived adult rats, whose whiskers had been trimmed for the preceding 12–27 days, and 12 to the control group. Quantitative analysis of axon length revealed that the average axon length decreased by 25% and the number of branching points by 32%. Innervation of L4 was decreased by 37% in the targeted column. This rewiring indicates that thalamo-cortical axons remain plastic in adulthood.

In vivo recordings of L4 neurons of the deprived group revealed that the number of action potentials from individual cells after sensory stimuli does not significantly differ from the control group. These results suggest that other elements of the thalamocortical

circuit must be plastic to compensate for the loss of synapses due to rewiring, potentially homeostatic strengthening of corticocortical and/or unpruned thalamocortical synapses.

6.3 3D neural network assembly and analysis

In this section, two studies are reviewed that used the reverse engineering approach to generate models of neural networks in the rat barrel cortex, and to make quantitative predictions about the synaptic connectivity between (populations of) neurons. First, the approach is used to predict cell-type-specific connectivity of thalamical VPM neurons in a model of a single cortical column (Oberlaender et al., 2012). Second, cell-to-cell connection probability is computed in a model of the entire barrel cortex and compared to experimental measurements reported in literature (R Egger et al., 2014).

6.3.1 3D cortical column: analysis of cytoarchitecture and synaptic connectivity

The reverse engineering approach to modeling anatomically realistic neural networks, results in an explicit 3D geometrical description, which enables quantitative 3D analysis of cytoarchitectonic properties of the modeled neuron population, and prediction of synaptic wiring. Such investigations are of particular interest to understand the structure of a cortical column, being a basic anatomical unit.

Oberlaender et al. (Oberlaender et al., 2012) constructed a 3D model representing the excitatory neuron population of a cortical column in the rat vibrissal cortex and its thalamic (VPM) input using the modeling approach described in Section 5.3 (Fig. 6.5) and quantified anatomical properties of interest, for example cell-type-specific dendrite-spine and VPM synapse densities.

The cell-type-specific dendrite-spine density (Fig. 6.6A, B) density indicates the spatial extent within the column where a specific cell type population receives its input. The spine density for each cell type population (i.e., each postsynaptic group, see Eq. 5.2, 5.3) was computed in $50^3\,\mu m^3$ grid cells for 3D visualization (Fig. 6.6A), and in $50\,\mu m$ thick xy-slabs for the depth profile (6.6B). Clearly observable from these analyses is that individual cell types have different innervation domains within the column, which overlap to differents extents. Some cell types have two major innervation domains, particularly L5tt and to a lesser extent L5st and L4py neurons.

After computing the 3D bouton density based on the VPM axons, the synapse estimation method described in Section 5.3.3 was applied to compute the number of synaptic contacts between the VPM axons and all cell types in the column (Fig. 6.6C, D) in $50^3\,\mu m^3$ grid cells. Cell-type-specific differences in synaptic innervation are clearly visible in the 3D density visualization and depth profile. The VPM innervation is largest in layer 4, with L4sp and L4ss neurons receiving most putative synapses, with a less pronounced innervation zone between Layer 5 and 6. The VPM synapse distributions displayed in general only a single innervation peak for each cell type, except for L6ct

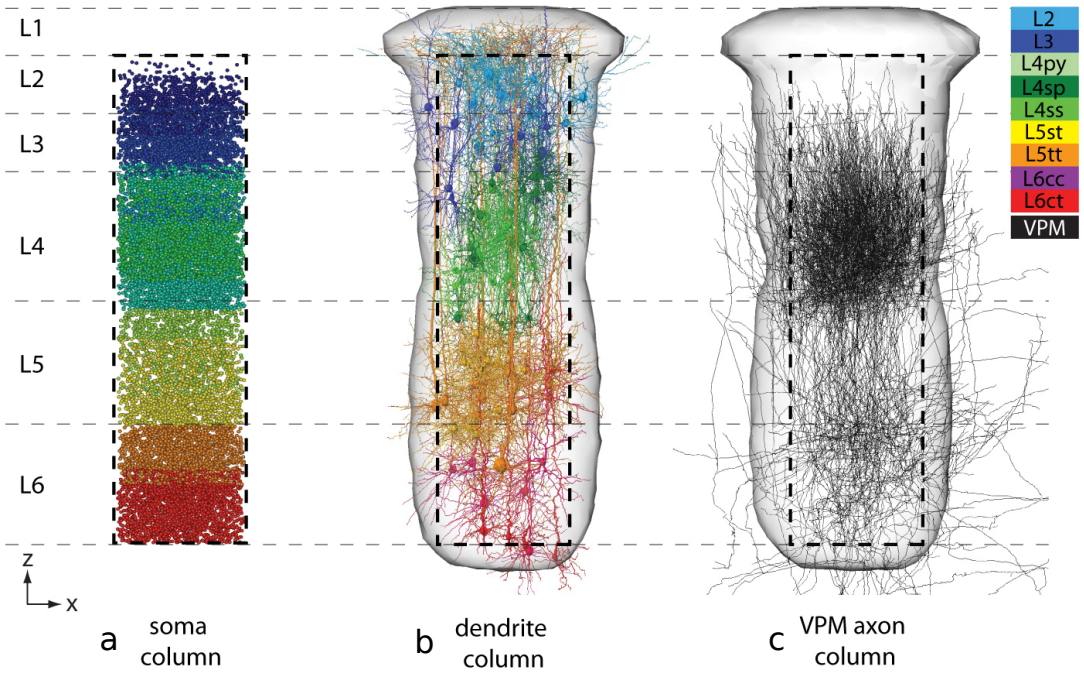

Figure 6.5: Reconstruction of thalamocortical circuits between VPM and excitatory neurons in a cortical column. (a) Excitatory neuron somata, colored by cell type. (b) 3D network of dendrites of excitatory neurons (only a small subset displayed for visualization purposes). (c) VPM axon reconstructions innervating the cortical column. Image reproduced from (Oberlaender et al., 2012).

and, to a lesser extent L5tt. Although the spine density for L5tt showed two major innervation domains, only the lower one is targeted by the VPM axons.

Although the synaptic connectivity is an estimation, which uses a heuristic based on structural overlap (for a discussion, see (Oberlaender et al., 2012)), the predicted bouton and VPM densities are in agreement with experimental studies (Bruno and Sakmann, 2006; Da Costa and Martin, 2011; HS Meyer et al., 2010).

6.3.2 Quantitative analysis of neuron-to-neuron connectivity in a model of the rat barrel cortex

In (R Egger et al., 2014) the barrel cortex model described in Ch. 5 was extended to include inhibitory neurons (interneurons). In addition, an explicit mathematical expression quantifying the *innervation* of one neuron by another, based on axon-dendritic overlap, is presented. From the innervation the probability that a connection by one or more synapses exists between any pair of neurons can be computed, resulting in a dense

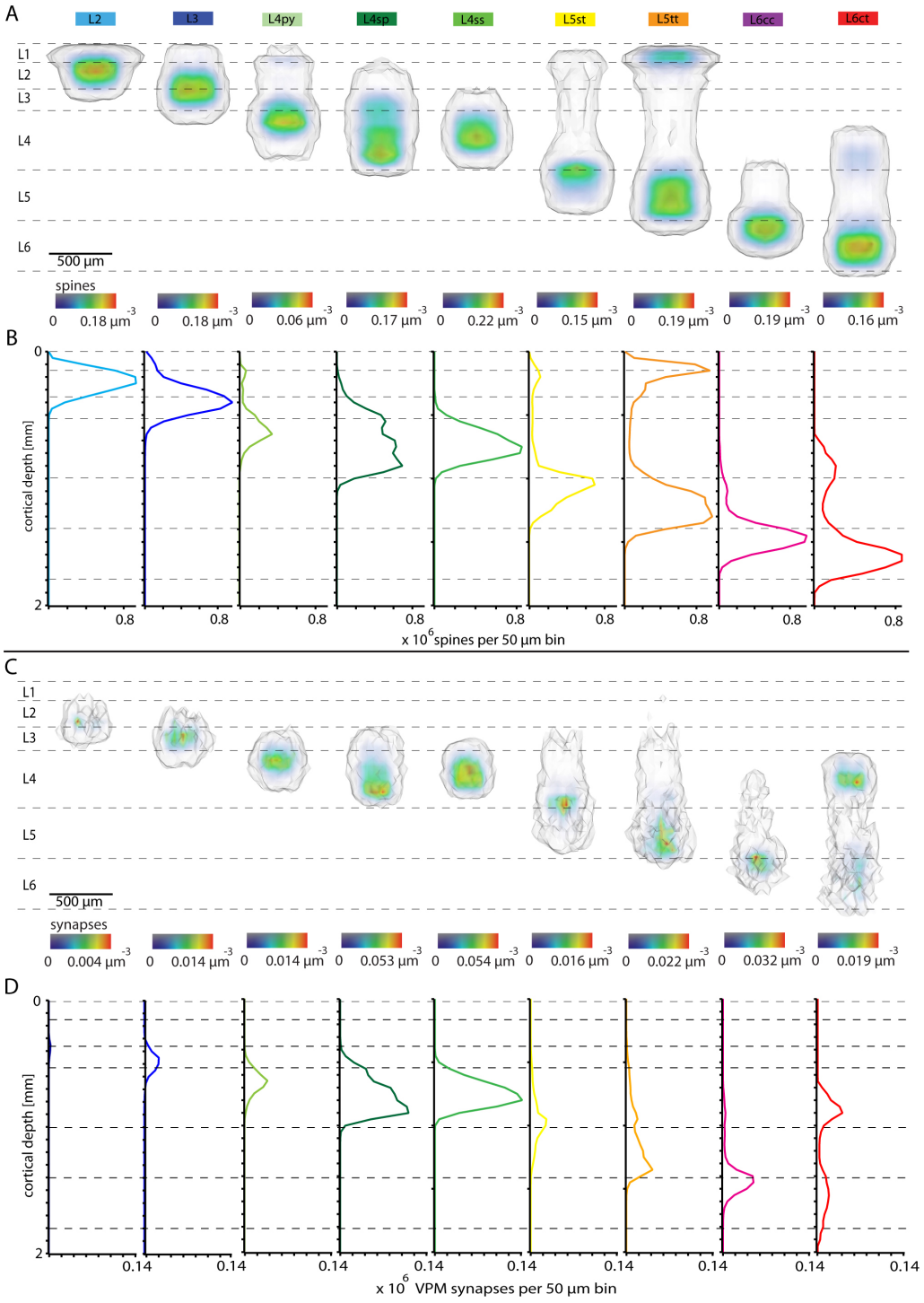

Figure 6.6: Quantitative analysis of cell-type-specific dendrite-spine and synapse densities. (A) 3D spine densities. (B) Spine density profile along column axis. (C) 3D synapse densities. (D) Synapse density profile. Image reproduced from (Oberlaender et al., 2012).

(i.e., cell-to-cell), *statistical connectome*. Based on innervation and connection probability, derived quantities, such as divergence (average number of postsynaptic neurons that connect to a single presynaptic neuron) and convergence (average number of postsynaptic neurons that connect to a single presynaptic neuron), can be computed, as well as the putative number of synapses. In the following this quantitative version of Peters' rule and comparisons of these predicted quantities with *in vivo*/*in vitro* measurements are briefly reviewed.

Interneurons form synaptic contacts not only at the spines, but also on the surface of soma and dendrites. Therefore, the computation of the synaptic connectivity (see Sec. 5.3.3) was updated to accommodate this. In the following, spines and post-synaptic contact sites on the surface are collectively called post-synaptic target sites (PSTs). Assuming that all PSTs in a voxel centered on $\mathbf{x}$ are equally likely to connect to any bouton in the same voxel, the probability that neuron j is targeted by a bouton of a neuron i with cell type $T(i)$ within $\mathbf{x}$ is given by:

$$p_j(\mathbf{x}, T(i)) = \frac{PST_j(\mathbf{x}, T(i))}{\sum_j PST_j(\mathbf{x}, T(i))} \tag{6.1}$$

If neuron i has B_i boutons in the voxel at $\mathbf{x}$, the average innervation $\tilde{I}_{ij}$ is defined as:

$$\tilde{I}_{ij} = B_i(\mathbf{x}) \cdot p_j(\mathbf{x}) \tag{6.2}$$

The probability that neuron j connects to n out of B_i voxels can be approximated by a Poisson distribution:

$$P(n; \tilde{I}_{ij}(\mathbf{x})) = \frac{\tilde{I}_{ij}^n(\mathbf{x})}{n!} e^{-\tilde{I}_{ij}(\mathbf{x})} \tag{6.3}$$

The probability of finding a connection between any two neurons i and j within a specific voxel at $\mathbf{x}$ equals 1 minus the probability that neuron j does not connect to any bouton in any voxel:

$$p_{ij}(\mathbf{x}) = 1 - P(n = 0; \tilde{I}_{ij}(\mathbf{x})) = 1 - e^{-\tilde{I}_{ij}} \tag{6.4}$$

Assuming that synapses in different voxels are formed independently of another, the total probability of finding a connection between neurons i and j is:

$$p_{ij} = 1 - \prod_{\mathbf{x}} P(n = 0; \tilde{I}_{ij}(\mathbf{x})) = 1 - e^{-\sum_{\mathbf{x}} \tilde{I}_{ij}(\mathbf{x})} = 1 - e^{-I_{ij}} \tag{6.5}$$

Here, I_{ij} is the total (i.e., summed over all voxels) average innervation from neuron i to neuron j. Intuitively, I_{ij} is the expected number of synapses connecting i to j. Thus, for a given innervation value I_{ij}, we can compute the probability that neuron i and j share n_{ij} synapses as: $n_{ij} = Poisson(n; I_{ij})$.

This approach is illustrated for the connection between a VPM axon and a L4 spiny stellate (L4ss) neuron in Fig. 6.7. Although there is substantial overlap between the

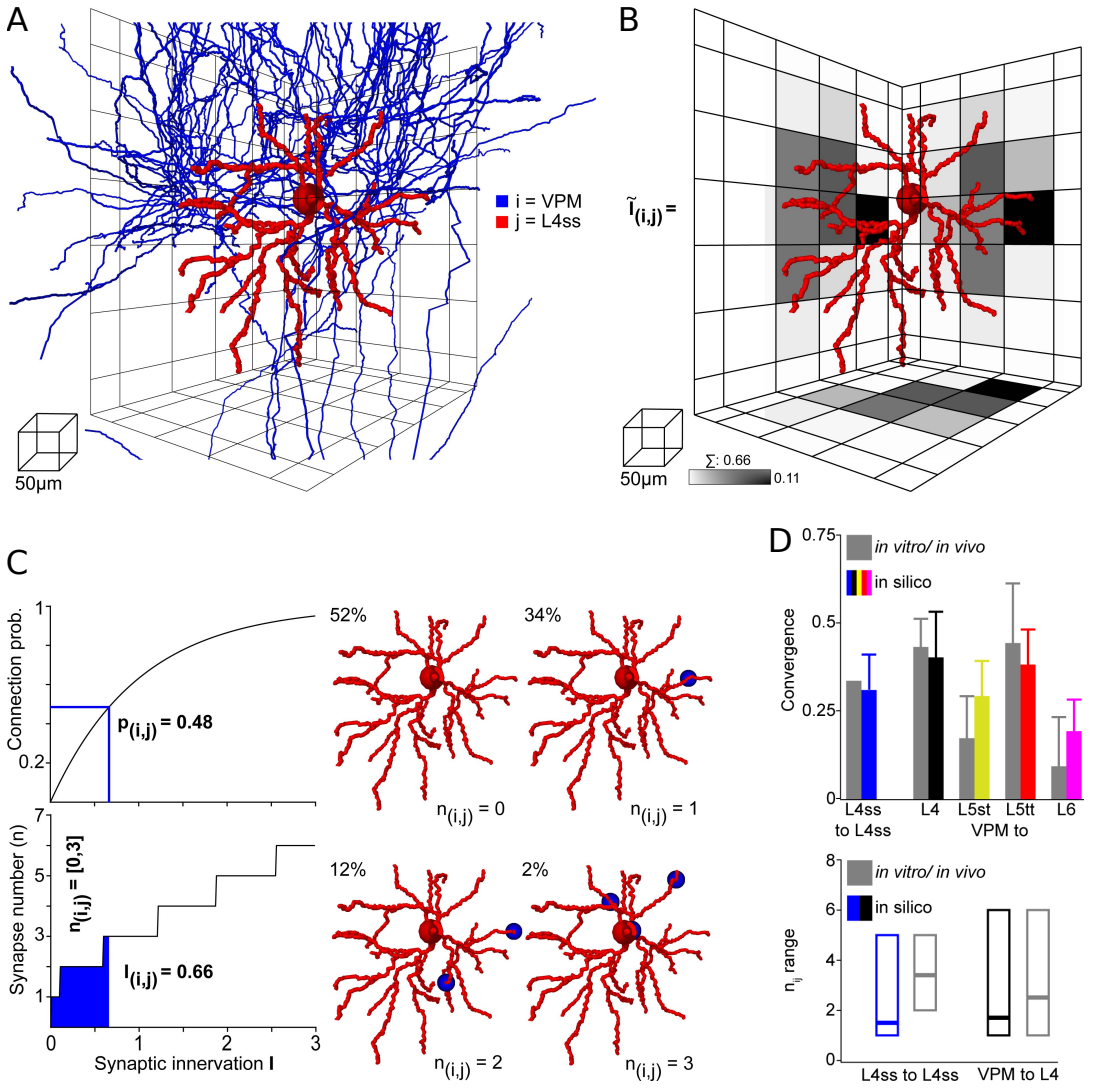

Figure 6.7: Computation of the statistical innervation, connection probability, and putative number of synapses between neurons in dense networks. (A) VPM axon (blue), L4 spiny stellate dendrite (red), and the grid used for computing bouton and PST densities. (B) The computed innervation $\tilde{I}_{ij}(\mathbf{x})$ from the VPM axon to the L4ss dendrite. The gray-colored squares in the grid represent the maximum projection of $\tilde{I}_{ij}(\mathbf{x})$. The total $I_{ij} = 0.66$. (C) Left top: connection probability from neuron i to j as a function of I_{ij}. Bottom: Possible range of number of synapses as a function of I_{ij}. Right: four possible synapse distributions and their probability of occurrence, computed from the 3D innervation density in (B). (D) Comparison of pair-wise connectivity statistics in the D2 column of the in silico model and experimental results from physiological and anatomical measurements in vitro and in vivo. Top: convergence of intra-barrel connectivity (L4ss-L4ss) and thalamocortical connectivity (VPM-L4/L5st/L5tt /L6). Bottom: observed and calculated range of number of synapses per connection. Image modified from (R Egger et al., 2014).

axon and dendrites (6.7A), the resulting innervation value I_{ij} equals merely 0.66 (6.7B), due to on the order of 1000 other potential target neurons in the same region competing for the same boutons. The probability that the two neurons are connected is therefore only 48% (6.7C), and the probability of the two cells sharing 1, 2 or 3 synapses is 34%, 12% and 2%, respectively.

To validate the model, the predicted convergence and putative number of synapses was compared to experimental measurements for connections between cell types for which results have been reported (Fig. 6.7D). The *in silico* predictions of convergence matched the previously reported values (Bruno and Sakmann, 2006; Constantinople and Bruno, 2013; Feldmeyer et al., 1999; CC Petersen and Sakmann, 2000) within one standard deviation. The range of putative synapses per connection also matched: for L4ss-to-L4ss connections this range was 1–5 (*in silico*), compared to 2–5 (*in vitro* (Feldmeyer et al., 1999)). For VPM-to-L4 connections, the range was 1–6 (*in silico*), compared to 1–6 (*in vivo* (Schoonover et al., 2014)).

6.4 Numerical simulation of neural signal processing

The neural network modeling approach described in Chapter 5 predicts the number and spatial distribution of synaptic connections between (groups of) neurons. This is essential information for investigating signal propagation through a network by numerical simulation. Here, two studies are presented illustrating that the proposed anatomical modeling method in combination with simulation is a feasible approach to investigate structure-function relationships in individual neurons (R Egger et al., 2014) and neuron populations (Lang et al., 2011).

6.4.1 Stimulus-dependent activity of a layer 5 thick-tufted neuron

Egger et al. (R Egger et al., 2014) illustrate how the anatomical modeling approach can be combined with simulation experiments to investigate structure-function relationships at the single neuron level. On the example of one specific L5 thick-tufted (L5tt) pyramidal neuron it is shown how neuron location and 3D morphology, as well as the behavioral state (passive whisker deflection vs. active touch), can largely influence a neuron's function in response to sensory stimulation.

In particular, Oberlaender et al. (Oberlaender et al., 2011) hypothesize that the interplay of spiking input of VPM, encoding touch, and L5st neurons, encoding whisking cycle phase, may cause L5tt serve as a coincidence detector encoding object localization (see also Sec. 6.2.2).

To look into this, Egger et al.(R Egger et al., 2014) first use the modeling approach described in Section 5.3 (Fig. 6.8A) to estimate the number of synapses, particularly from VPM and L5st from the principal whisker (PW) and the eight surrounding whiskers (Su8W), and their position on the dendrites of an exemplary L5tt neuron in the D2 column (Figure 6.8B,C). The estimated number of contacts of this neuron with presynaptic neurons is 20658. Roughly 2% (460) of these synapses stem from the VPM. Interestingly,

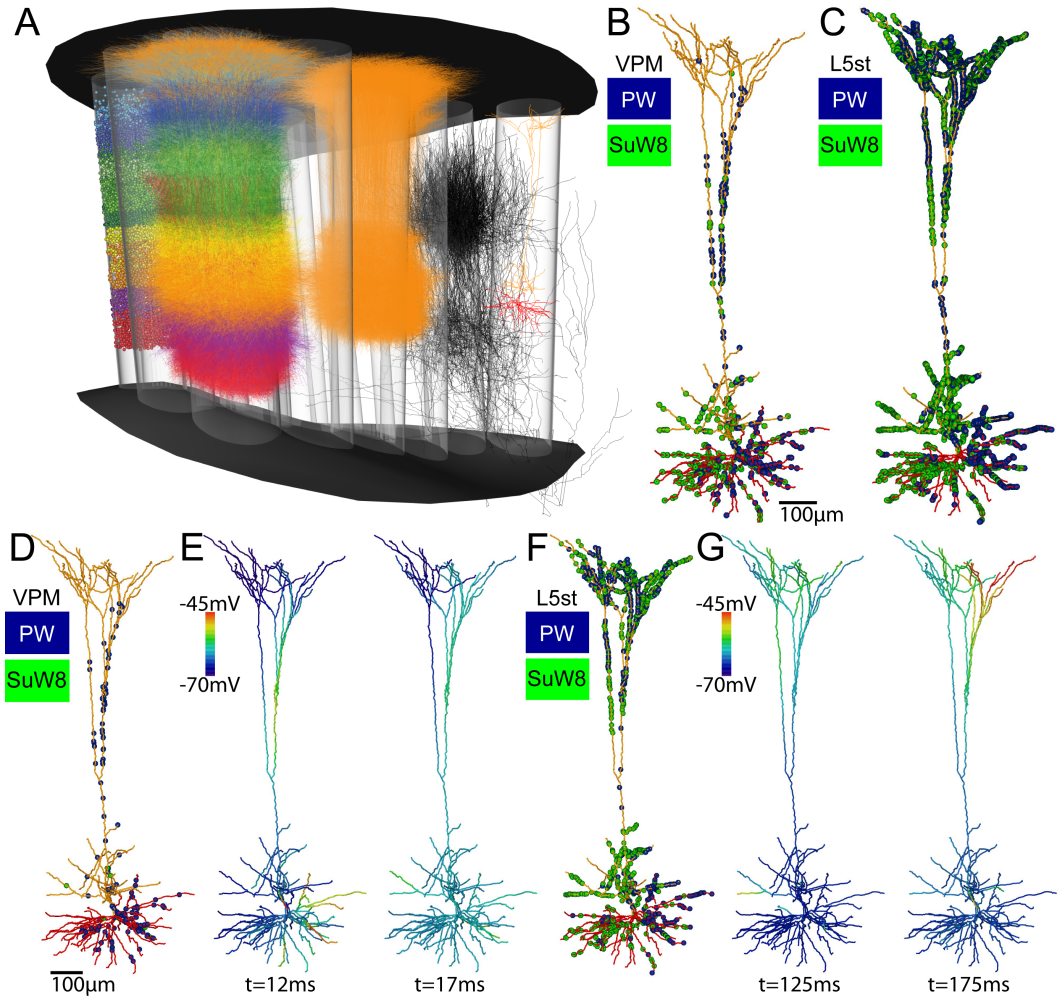

Figure 6.8: (A) The reverse engineering pipeline. From left to right: assignment of neuron somata to different cell types, replacement of each neuron soma with a dendrite tracing of the same cell type, all L5tt dendrites of one column, all VPM axons of one column, an individual L5tt neuron registered to its location in the average network model. (B) Subcellular distribution of VPM synapses from the principal whisker (PW) and the eight surrounding whiskers (Su8W) on the individual L5tt neuron from panel A. (C) Subcellular distribution of L5st synapses from the PW and Su8W on the individual L5tt neuron from A. (D) Active VPM synapses after passive touch. Scalebar applies to D–G. (E) Spread of depolarization along the neuron 12 ms and 17 ms after passive touch, respectively. (F) L5st synapses with increased activity during whisking. (G) Spread of depolarization along the neuron 125 ms and 175 ms after beginning of the whisking cycle, respectively. Image reproduced from (R Egger et al., 2014).

only 245 of the VPM synapses originate from the corresponding D2 barreloid in the thalamus, nearly the same amount as from the Su8W (215). This estimate of the structural thalamocortical connectivity may in part explain why L5tt neurons usually respond well to deflections of the PW and Su8W, in contrast to VPM neurons, which responses are largely confined to the PW. Approximately 16% (3 368) of the cell's synapses stem from L5st neurons, the majority (2 055) originating in Su8W.

Figure 6.8B and C show a possible distribution of the predicted synapses on the dendritic branches. Quantification of the path length from each of these synapse positions to the soma reveals a striking difference in subcellular location of VPM and L5st synapses. Approximately 79% of the VPM synapses are located on the dendrites around the soma (proximal contacts, see also Fig. 1.1a); the remaining 21% are located on the apical trunk and tuft (distal contacts). In contrast, ∼60% of the L5st synapses are located on the apical tuft, while 40% are proximal contacts.

Second, using a simplified full-compartmental simulation model (see (R Egger et al., 2014) for details), the potential spread of depolarization caused by VPM input after passive whisker touch (Brecht and Sakmann, 2002; Bruno and Sakmann, 2006) and by L5st input during periods of active whisking (De Kock and Sakmann, 2009; Oberlaender et al., 2011) was investigated.

After VPM input 145 of 245 PW and 8 of 215 Su8W synapses were active (Fig. 6.8D). Figure 6.8E shows the resulting depolarization 12 ms (left panel) and 17 ms (right panel) after the stimulus. The depolarization by VPM input after passive touch remains largely restricted to the basal and apical oblique dendrites and does not spread into the apical tuft.

L5st input was simulated during one whisking cycle, effectively activating 444 of 1313 PW and 736 of 2055 Su8W synapses (Fig. 6.8F). Figure 6.8G shows the resulting depolarization 125 ms (left panel) and 175 ms (right panel) after the onset of the whisker movement. In contrast to the VPM activation after passive touch, the depolarization by L5st input remains largely confined to the apical tuft dendrite, leaving the proximal dendrites largely at rest.

Egger et al. (R Egger et al., 2014) conclude: "These simplified simulation experiments demonstrate that a typical L5tt neuron may have two structurally and functionally distinct domains, the proximal and apical tuft dendrites, respectively. The two compartments receive very different synaptic input and may thus process and encode different aspects of whisker-evoked sensory stimuli. Depending on the spatial and temporal superposition of the two inputs, the output of a L5tt neuron may change, encoding information of the environment, such as the location of an object".

6.4.2 Thalamocortical activation of a L4 spiny stellate neuron population

Lang et al. (Lang et al., 2011) illustrate how the reverse engineering approach can be used to investigate structure-function relationships at the network, cellular, and subcellular scale.

The network modeling approach described in Chapter 5 was employed to create a network formed between the thalamus (VPM) and spiny stellate neurons in layer 4 (L4ss) of a cortical barrel column in rat vibrissal cortex. After (passive) whisker deflection, the VPM cells activate synaptically connected neurons of the L4ss population by emitting action potentials (APs, spikes), resulting in electrical signals on the postsynaptic side. These travel along the dendrites towards the soma, resulting in a excitatory postsynaptic potential (EPSP). Superposition of unitary EPSPs (uEPSPs, i.e. EPSPs originating from different synapses) results in an action potential of the L4ss cell when the accumulated potential exceeds a threshold.

The ensemble of L4ss neurons within an average barrel comprises 2752 ± 46 dendrite morphologies. This ensemble is innervated by 285 ± 13 thalamocortical axons from the respective VPM barreloid, which results in an average number of 246 ± 123 VPM synapses per L4ss neuron. These numbers resemble results from previous studies that investigated connectivity between thalamus and L4ss neurons in the cortex (Bruno and Sakmann, 2006; Da Costa and Martin, 2011).

The activation of the L4ss population by the VPM cells is numerically simulated using *NeuroDUNE*, a software framework for simulating (networks of) full-compartmental neurons. The electrical membrane parameters of the L4ss neurons were chosen in agreement with previously reported experimental data (see (Lang et al., 2011) for details). The VPM spike probability and timing is based on *in vivo* measurements (Brecht and Sakmann, 2002; Bruno and Sakmann, 2006). In the simulation experiment, the number and location of L4ss neurons was investigated that showed a subthreshold or spiking response after VPM activation.

Lang et al. (Lang et al., 2011) report findings both at the network and the subcellular level. At the network level, reconstruction of the network structure and synaptic wiring predicted that the number of VPM synapses on a postsynaptic cell strongly depends on the soma location of the L4ss neurons. Specifically, the number of VPM synapses decreased with increasing distance from the barrel column center (BCC) toward the barrel borders (Fig. 6.9D). Activating the L4ss population by VPM input results in similar location-specific spiking responses (Fig. 6.9A, C). While the relative number of neurons producing a subthreshold response displays only a weak radial decay (∼19%) toward the barrel borders, spiking responses decay dramatically (∼60% lower than at the BCC). A similar radial decay in spiking probability has been observed for L2/3 neurons in a barrel column of mouse vibrissal cortex using 2-photon Ca^{2+} imaging (Kerr et al., 2007).

In addition to this structure-function relationship at the network level level, the simulation results predict a second mechanism at the subcellular scale. A bimodal histogram of uEPSP amplitude and peak time suggests that VPM synapses may be pooled into proximal (close to the soma, therefore shorter peak times and larger amplitudes) and distal (further from soma, longer peak times, smaller amplitude) synapses (Fig. 6.10B, C). In addition to the total number of synapses, the relative numbers of proximal and distal VPM synapses per L4ss neuron may also be location-specific. Toward the barrel borders, the number of proximal contacts decreases more than the number of distal con-

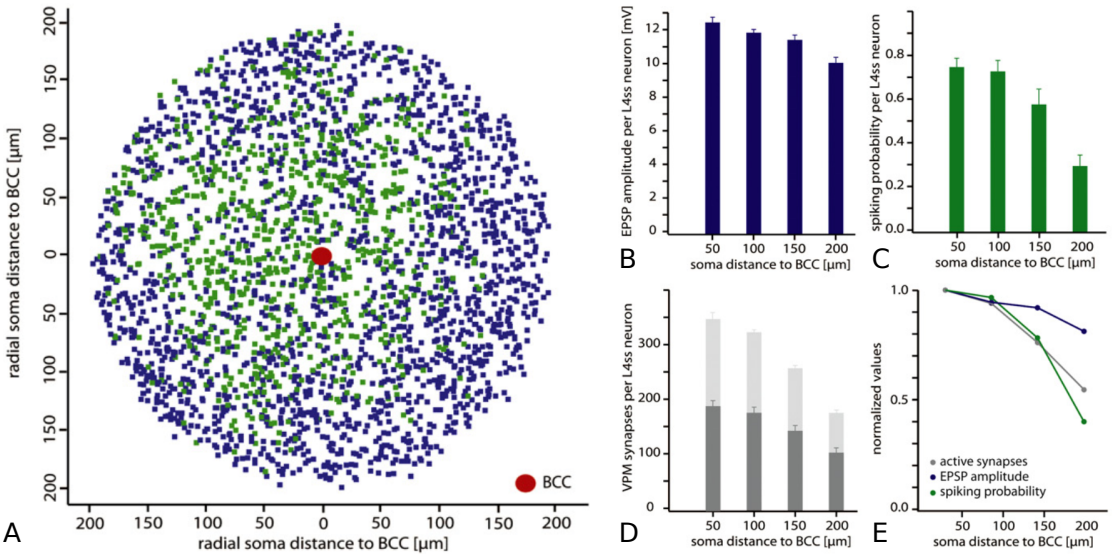

Figure 6.9: Radial decay in L4ss spiking. (A) Top view onto the distribution of L4ss somata in a barrel column. Spiking neurons are shown in green and are preferentially located around the barrel column center (BCC). (B) Quantification of the radial dependence of the subthreshold (i.e. EPSP amplitude at the soma) and (C) spiking responses for 5 simulation trials. (D) The radial decay in spiking follows the radial decay of (active) VPM synapses (light gray: all synapses, dark gray: active synapses). The decay in spiking at the barrel borders even exceeds the decay of active synapses (E). This may reflect a stronger reduction of proximal VPM synapses toward the barrel borders, when compared to distal contacts. Image modified from (Lang et al., 2011).

tacts (with 14% and 5% respectively). This could explain the spiking probability at the barrel borders being even lower than predicted by the radial decay in VPM synapses per cell (Fig. 6.9E).

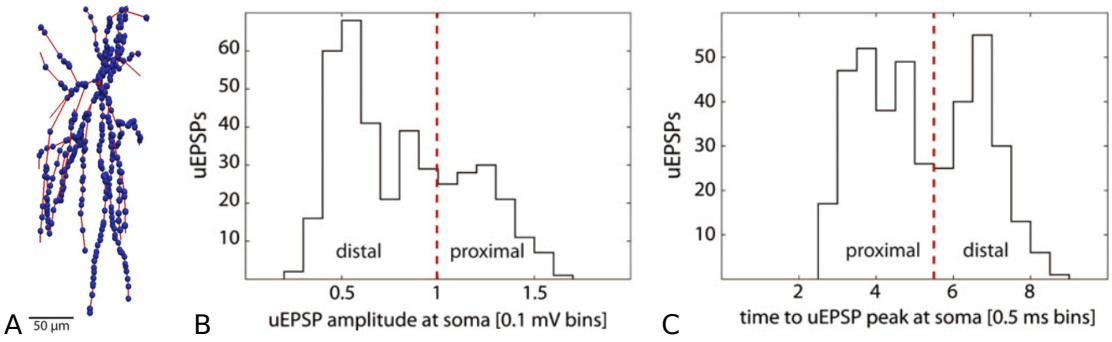

Figure 6.10: (A) Realization of synapse locations on dendrites of a L4 spiny stellate neuron. (B,C) The histograms of the uEPSP amplitudes and peak times display bimodal distributions for this example neuron, suggesting a pooling into proximal (close to the soma) and distal (further from soma) synapses. Image modified from (Lang et al., 2011).

7 Conclusion

This thesis presented a complete set of methods and tools to create and analyze anatomically realistic 3D models of neural networks in the rat barrel cortex. These networks consist of populations of neurons and their synaptic connectivity. They are created using anatomical data reconstructed from 3D microscopic images. The particular contributions are:

- an automatic algorithm for the automatic detection and counting of neuron somata in 3D images,
- an interactive proof-editor for validating, correcting, alignment, merging, annotating, and analyzing tracings of filamentous structures obtained from multiple brain sections,
- an automatic method for the 2D rigid alignment of 3D sections containing filamentous structures, such as neuronal arborizations,
- a tool to create 3D a model of a neural network representing the rat barrel cortex, based on reconstructed anatomical data, and
- a framework for the interactive visual analysis of synaptic connectivity at multiple scales.

We have seen how these visual computing techniques support neuroscientists in addressing biological questions regarding anatomy and structure-function relationships of neural circuits. Particular aspects where these techniques significantly contribute to advancing the field of neuroscience are:

- Mastering scale and complexity. The presented tools address the challenges induced by the size of the brain area under investigation, and/or the size of the data to a large extent by automation of labor-intensive tasks. For example, the automatic cell counting algorithm (Ch. 2) makes the exhaustive counting of all neurons in a large brain area (such as the rat barrel cortex) possible (Sec. 6.1). The presented neuron reconstruction workflow (Ch. 4) significantly reduces the amount of manual labor, and the dependency on neuroscience experts, effectively increasing the number of reconstructions that can be created within a particular time frame.

- Quality assurance. Visual computing techniques are essential to assess and improve the quality of data extracted from 3D images for the following reasons. First, a common approach to validate image analysis methods is to compare their results to an expert-generated ground truth. This requires either a quantitative comparison algorithm and/or an effective visualization (e.g., see Sec. 2.5.3, 3.4, 4.3.10). Interactive tools are often required to create a ground truth in the first place (e.g., see Sec.4.3.7). Second, visual proof-editing tools enable the user to interactively make corrections (e.g., see Sec. 4.3.7). In the case of the Filament Editor, the user is intentionally part of the reconstruction workflow: the interactive editing/ correcting employs human pattern recognition abilities to resolve ambiguities in often unwieldy data, thus complementing automatic methods in an efficient way.

- Quantitative analysis. Visual computing methods are a prerequisite to increase sample sizes of the investigated objects. Shifting from anecdotal to statistically relevant evidence increases confidence in the reported quantities of interest. For example, the presented automated method allows to exhaustively count the number of neurons in large brain volumes (see Ch. 2), thereby avoiding errors due to extrapolation of manually counted smaller samples. Also, classification of morphological cell types (6.2.1) is only meaningful, when the sample size is large enough to be considered representative for the entire population.

- Model building. The 3D neural network models are concrete representations of accumulated anatomical knowledge of the brain region at hand. As such they may serve as an integrated view on the current state of knowledge, and allow to put individual findings into a broader (network) perspective. Additionally, from the 3D neural network models and the associated algorithm to estimate the synaptic connectivity (Ch. 5) quantitative predictions involving anatomy, and, through numerical simulations (Sec. 6.4) also physiology, can be generated, which can be compared to experimental measurements (Sec. 6.3) for model validation, or used as testable hypotheses of experimentally inaccessible network properties.

- Interactive, visual data exploration. Interactive, visual data exploration aids in acquainting oneself with the data, data set comparison, anomaly detection, generating hypotheses, etc. when a priori knowledge about the data is limited. In this neuroscientific context, this is useful from the relatively trivial scenario of comparing the 3D morphology of individual neurons (Dercksen et al., 2012) to the complex case of multi-scale exploration of synaptic connectivity from the subcellular to the population level (Sec. 5.4).

Altogether, visual computing techniques are key prerequisites to gain understanding of structure and function in neural networks, for example the mechanistic principles underlying sensory information processing in the mammalian brain, and have therefore become essential in the daily work of neuroscientists.

References

Abercrombie, M (1946). "Estimation of nuclear population from microtome sections". *The Anatomical Record*, 94(2), pp. 239–247.

Akemann, W, YM Zhong, N Ichinohe, KS Rockland, and T Knöpfel (2004). "Transgenic mice expressing a fluorescent in vivo label in a distinct subpopulation of neocortical layer 5 pyramidal cells". *Journal of Comparative Neurology*, 480(1), pp. 72–88.

Akutsu, T (1996). "Protein structure alignment using dynamic programing and iterative improvement". *IEICE Transactions on Information and Systems*, E79-D(12), pp. 1629–1636.

Andres, B, U Köthe, M Helmstaedter, W Denk, and FA Hamprecht (2008). "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification". In: *Pattern Recognition*. Ed. by G Rigoll. Vol. 5096. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 142–152.

Ankerst, M, MM Breunig, HP Kriegel, and J Sander (1999). "OPTICS: ordering points to identify the clustering structure". In: *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*. Philadelphia: ACM Press, pp. 49–60.

Ascoli, GA (2006). "Mobilizing the base of neuroscience data: the case of neuronal morphologies". *Nature Reviews Neuroscience*, 7(4), pp. 318–324.

Ascoli, GA, DE Donohue, and M Halavi (2007). "NeuroMorpho.Org: a central resource for neuronal morphologies". *The Journal of Neuroscience*, 27(35), pp. 9247–9251.

Bajcsy, P, SC Lee, A Lin, and R Folberg (2006). "Three-dimensional volume reconstruction of extracellular matrix proteins in uveal melanoma from fluorescent confocal laser scanning microscope images". *Journal of Microscopy*, 221(Pt 1), pp. 30–45.

Bassett, DS, JA Brown, V Deshpande, JM Carlson, and ST Grafton (2011). "Conserved and variable architecture of human white matter connectivity". *NeuroImage*, 54(2), pp. 1262–1279.

Baum, D (2007). "A point-based algorithm for multiple 3D surface alignment of drug-sized molecules". PhD thesis. Freie Universität Berlin.

Beaulieu, C (1993). "Numerical data on neocortical neurons in adult rat, with special reference to the GABA population". *Brain Research*, 609(1–2), pp. 284–292.

Binzegger, T, RJ Douglas, and KAC Martin (2004). "A quantitative map of the circuit of cat primary visual cortex". *The Journal of Neuroscience*, 24(39), pp. 8441–8453.

Bock, DD, WCA Lee, AM Kerlin, ML Andermann, G Hood, AW Wetzel, S Yurgenson, ER Soucy, HS Kim, and RC Reid (2011). "Network anatomy and in vivo physiology of visual cortical neurons". *Nature*, 471(7337), pp. 177–182.

Borisyuk, R, AK Al Azad, D Conte, A Roberts, and SR Soffe (2011). "Modeling the connectome of a simple spinal cord". *Frontiers in Neuroinformatics*, 5(20).

Brecht, M and B Sakmann (2002). "Whisker maps of neuronal subclasses of the rat ventral posterior medial thalamus, identified by whole-cell voltage recording and morphological reconstruction". *The Journal of Physiology*, 538(Pt 2), pp. 495–515.

Briggman, KL, M Helmstaedter, and W Denk (2011). "Wiring specificity in the direction-selectivity circuit of the retina". *Nature*, 471(7337), pp. 183–188.

Bron, C and J Kerbosch (1973). "Algorithm 457: finding all cliques of an undirected graph". *Communications of the ACM*, 16(9), pp. 575–577.

Broser, PJ, V Grinevich, P Osten, B Sakmann, and DJ Wallace (2008). "Critical period plasticity of axonal arbors of layer 2/3 pyramidal neurons in rat somatosensory cortex: layer-specific reduction of projections into deprived cortical columns". *Cerebral Cortex*, 18(7), pp. 1588–1603.

Broser, PJ, R Schulte, S Lang, A Roth, F Helmchen, J Waters, B Sakmann, and G Wittum (2004). "Nonlinear anisotropic diffusion filtering of three-dimensional image data from two-photon microscopy". *Journal of Biomedical Optics*, 9(6), pp. 1253–64.

Bruno, RM, TTG Hahn, DJ Wallace, CPJ De Kock, and B Sakmann (2009). "Sensory experience alters specific branches of individual corticocortical axons during development". *Journal of Neuroscience*, 29(10), pp. 3172–3181.

Bruno, RM and B Sakmann (2006). "Cortex is driven by weak but synchronously active thalamocortical synapses". *Science*, 312(5780), pp. 1622–1627.

Byun, J, MR Verardo, B Sumengen, GP Lewis, BS Manjunath, and SK Fisher (2006). "Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images". *Molecular Vision*, 12, pp. 949–960.

Cannon, RC, DA Turner, GK Pyapali, and HV Wheal (1998). "An on-line archive of reconstructed hippocampal neurons". *Journal of Neuroscience Methods*, 84(1–2), pp. 49–54.

Carnevale, NT and ML Hines (2006). *The NEURON book*. New York: Cambridge University Press.

Celikel, T and B Sakmann (2007). "Sensory integration across space and in time for decision making in the somatosensory system of rodents". *Proceedings of the National Academy of Sciences of the United States of America*, 104(4), pp. 1395–1400.

Chang, H and B Parvin (2006). "Segmentation of three dimensional cell culture models from a single focal plane". In: *Advances in Visual Computing*. Ed. by G Bebis, R Boyle, B Parvin, D Koracin, P Remagnino, A Nefian, G Meenakshisundaram, V Pascucci, J Zara, J Molineros, H Theisel, and T Malzbender. Vol. 4292. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 586–595.

Cheng, J and JC Rajapakse (2009). "Segmentation of clustered nuclei with shape markers and marking function". *IEEE Transactions on Biomedical Engineering*, 56(3), pp. 741–748.

Chui, H and A Rangarajan (2003). "A new point matching algorithm for non-rigid registration". *Computer Vision and Image Understanding*, 89(2–3), pp. 114–141.

Constantinople, CM and RM Bruno (2013). "Deep cortical layers are activated directly by thalamus". *Science*, 340(6140), pp. 1591–1594.

Cragg, BG (1967). "The density of synapses and neurones in the motor and visual areas of the cerebral cortex". *Journal of Anatomy*, 101(Pt 4), pp. 639–654.

Crook, S, P Gleeson, F Howell, J Svitak, and RA Silver (2007). "MorphML: level 1 of the NeuroML standards for neuronal morphology data and model specification". *Neuroinformatics*, 5(2), pp. 96–104.

Da Costa, NM and KAC Martin (2011). "How thalamus connects to spiny stellate cells in the cat's visual cortex". *The Journal of Neuroscience*, 31(8), pp. 2925–2937.

Daněk, O, P Matula, C Ortiz-de-Solórzano, A Muñoz-Barrutia, M Maška, and M Kozubek (2009). "Segmentation of touching cell nuclei using a two-stage graph cut model". In: *Proceedings of the Scandinavian Conference on Image Analysis*. Ed. by AB Salberg, JY Hardeberg, and R Jenssen. Springer-Verlag Berlin Heidelberg, pp. 410–419.

De Bruin, PW, VJ Dercksen, FH Post, AM Vossepoel, GJ Streekstra, and FM Vos (2005). "Interactive 3D segmentation using connected orthogonal contours". *Computers in Biology and Medicine*, 35(4), pp. 329–346.

De Kock, CPJ, RM Bruno, H Spors, and B Sakmann (2007). "Layer- and cell-type-specific suprathreshold stimulus representation in rat primary somatosensory cortex". *The Journal of Physiology*, 581(1), pp. 139–154.

De Kock, CPJ and B Sakmann (2009). "Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific". *Proceedings of the National Academy of Sciences of the United States of America*, 106(38), pp. 16446–16450.

Dercksen, VJ, C Brüß, D Stalling, S Gubatz, U Seiffert, and HC Hege (2008). "Towards automatic generation of 3D models of biological objects based on serial sections". In: *Visualization in Medicine and Life Sciences*. Ed. by L. Linsen, H. Hagen, and B. Hamann. Springer-Verlag Berlin Heidelberg, pp. 3–25.

Dercksen, VJ, R Egger, HC Hege, and M Oberlaender (2012). "Synaptic connectivity in anatomically realistic neural networks: modeling and visual analysis". In: *Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)*. Norrköping, Sweden, pp. 17–24.

Dercksen, VJ, HC Hege, and M Oberlaender (2014). "The Filament Editor: an interactive software environment for visualization, proof-editing and analysis of 3D neuron morphology". *Neuroinformatics*, 12(2), pp. 325–339.

Dercksen, VJ, M Oberlaender, B Sakmann, and HC Hege (2012). "Interactive visualization – a key prerequisite for reconstruction of anatomically realistic neural networks". In: *Visualization in Medicine and Life Sciences II*. Ed. by L Linsen, H Hagen, B Hamann, and HC Hege. Springer-Verlag Berlin, pp. 27–44.

Dercksen, VJ, S Prohaska, and HC Hege (2005). "Fast Cross-sectional Display of Large Data Sets". In: *IAPR Conference on Machine Vision Applications*. Tsukuba, Japan, pp. 336–339.

Dercksen, VJ, B Weber, D Günther, M Oberlaender, S Prohaska, and HC Hege (2009). "Automatic alignment of stacks of filament data". In: *Proceedings of the IEEE In-*

ternational Symposium on Biomedical Imaging: From Nano to Macro. Boston, MA, pp. 971–974.

Diamond, ME, M von Heimendahl, PM Knutsen, D Kleinfeld, and E Ahissar (2008). "'Where' and 'what' in the whisker sensorimotor system". *Nature Reviews Neuroscience*, 9(8), pp. 601–612.

Dodt, HU, U Leischner, A Schierloh, N Jährling, CP Mauch, K Deininger, JM Deussing, M Eder, W Zieglgänsberger, and K Becker (2007). "Ultramicroscopy: three-dimensional visualization of neuronal networks in the whole mouse brain". *Nature Methods*, 4(4), pp. 331–336.

Donaldson, HH (1895). *The growth of the brain: a study of the nervous system in relation to education*. London: W. Scott.

Donohue, DE and GA Ascoli (2011). "Automated reconstruction of neuronal morphology: an overview". *Brain Research Reviews*, 67(1–2), pp. 94–102.

Egger, R, VJ Dercksen, CPJ De Kock, and M Oberlaender (2014). "Reverse engineering the 3D structure and sensory-evoked signal flow of the rat vibrissal cortex". In: *The Computing Dendrite*. Ed. by H Cuntz, MWH Remme, and B Torben-Nielsen. Springer Series in Computational Neuroscience. Springer, pp. 127–145.

Egger, R, VJ Dercksen, D Udvary, HC Hege, and M Oberlaender (2014). "Generation of dense statistical connectomes from sparse morphological data". *Frontiers in Neuroanatomy*, 8(129).

Egger, R, RT Narayanan, M Helmstaedter, CPJ De Kock, and M Oberlaender (2012). "3D Reconstruction and standardization of the rat vibrissal cortex for precise registration of single neuron morphology". *PLoS Computational Biology*, 8(12), e1002837.

Ertürk, A, CP Mauch, F Hellal, F Förstner, T Keck, K Becker, N Jährling, H Steffens, M Richter, M Hübener, E Kramer, F Kirchhoff, HU Dodt, and F Bradke (2012). "Three-dimensional imaging of the unsectioned adult spinal cord to assess axon regeneration and glial responses after injury". *Nature Medicine*, 18(1), pp. 166–71.

Fatakdawala, H, J Xu, A Basavanhally, G Bhanot, S Ganesan, M Feldman, JE Tomaszewski, and A Madabhushi (2010). "Expectation-maximization-driven geodesic active contour with overlap resolution (EMaGACOR): application to lymphocyte segmentation on breast cancer histopathology". *IEEE Transactions on Biomedical Engineering*, 57(7), pp. 1676–1689.

FEI Visualization Sciences Group (2013a). *Amira 5.5*.

— (2013b). *Amira 5.5 Reference Guide*.

— (2013c). *Open Inventor 8.1*.

Feldman, DE and M Brecht (2005). "Map plasticity in somatosensory cortex". *Science*, 310(5749), pp. 810–815.

Feldmeyer, D, V Egger, J Lubke, and B Sakmann (1999). "Reliable synaptic connections between pairs of excitatory layer 4 neurones within a single 'barrel' of developing rat somatosensory cortex". *The Journal of Physiology*, 521(Pt 1), pp. 169–90.

Feldmeyer, D and B Sakmann (2000). "Synaptic efficacy and reliability of excitatory connections between the principal neurones of the input (layer 4) and output layer (layer 5) of the neocortex". *The Journal of Physiology*, 525(Pt 1), pp. 31–39.

Fischler, MA and RC Bolles (1981). “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. *Communications of the ACM*, 24(6). Ed. by MA Fischler and O Firschein, pp. 381–395.

Frick, A, D Feldmeyer, M Helmstaedter, and B Sakmann (2008). “Monosynaptic connections between pairs of L5A pyramidal neurons in columns of juvenile rat somatosensory cortex”. *Cerebral Cortex*, 18(2), pp. 397–406.

Fuchs, H, ZM Kedem, and SP Uselton (1977). “Optimal surface reconstruction from planar contours”. *Communications of the ACM*, 20(10), pp. 693–702.

Gamma, E, R Helm, R Johnson, and J Vlissides (1994). *Design patterns: elements of reusable object-oriented software*. Reading, MA: Addison-Wesley Longman, Inc.

Gerhard, S, A Daducci, A Lemkaddem, R Meuli, JP Thiran, and P Hagmann (2011). “The Connectome Viewer Toolkit: an open source framework to manage, analyze, and visualize connectomes”. *Frontiers in Neuroinformatics*, 5(3), pp. 1–15.

Gillette, Ta, KM Brown, and Ga Ascoli (2011). “The DIADEM metric: comparing multiple reconstructions of the same neuron”. *Neuroinformatics*, 9(2–3), pp. 233–245.

Gonzalez, RC and RE Woods (2002). *Digital Image Processing*. 2nd Ed. Upper Saddle River, NJ: Prentice Hall.

Groh, A, HS Meyer, EF Schmidt, N Heintz, B Sakmann, and P Krieger (2010). “Cell-type specific properties of pyramidal neurons in neocortex underlying a layout that is modifiable depending on the cortical area”. *Cerebral Cortex*, 20(4), pp. 826–836.

Guo, Y, Y Wang, S Fang, H Chao, AJ Saykin, and L Shen (2012). “Pattern Visualization of Human Connectome Data”. In: *EuroVis - Short Papers*. Ed. by T Weinkauf and M Meyer. Vienna, Austria, pp. 78–83.

Hagmann, P, L Cammoun, X Gigandet, S Gerhard, P Ellen Grant, V Wedeen, R Meuli, JP Thiran, CJ Honey, and O Sporns (2010). “MR connectomics: principles and challenges”. *Journal of Neuroscience Methods*, 194(1), pp. 34–45.

Hagmann, P, L Cammoun, X Gigandet, R Meuli, CJ Honey, VJ Wedeen, and O Sporns (2008). “Mapping the structural core of human cerebral cortex”. *PLoS Biology*, 6(7), e159.

Halavi, M, Ka Hamilton, R Parekh, and Ga Ascoli (2012). “Digital reconstructions of neuronal morphology: three decades of research trends”. *Frontiers in Neuroscience*, 6(49).

Hege, HC, M Seebaß, D Stalling, and M Zöckler (1997). *A generalized marching cubes algorithm based on non-binary classifications*. Tech. rep. SC-97-05. Zuse Institute Berlin.

Helmstaedter, M (2013). “Cellular-resolution connectomics: challenges of dense neural circuit reconstruction”. *Nature Methods*, 10(6), pp. 501–507.

Helmstaedter, M, KL Briggman, and W Denk (2011). “High-accuracy neurite reconstruction for high-throughput neuroanatomy”. *Nature Neuroscience*, 14(8), pp. 1081–1088.

Helmstaedter, M, CPJ de Kock, D Feldmeyer, RM Bruno, and B Sakmann (2007). “Reconstruction of an average cortical column in silico”. *Brain Research Reviews*, 55(2), pp. 193–203.

Hill, SL, Y Wang, I Riachi, F Schürmann, and H Markram (2012). "Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits". *Proceedings of the National Academy of Sciences of the United States of America*, 109(42), E2885–94.

Hogrebe, L, ARC Paiva, E Jurrus, C Christensen, M Bridge, L Dai, RL Pfeiffer, PR Hof, B Roysam, JR Korenberg, and T Tasdizen (2012). "Serial section registration of axonal confocal microscopy datasets for long-range neural circuit reconstruction". *Journal of Neuroscience Methods*, 207(2), pp. 200–210.

Homberg, U, R Binner, S Prohaska, VJ Dercksen, A Kuß, and U Kalbe (2009). "Determining Geometric Grain Structure from X-Ray Micro-Tomograms of Gradated Soil". In: *Workshop Internal Erosion*. Vol. 21, pp. 37–52.

Horikawa, K and WE Armstrong (1988). "A versatile means of intracellular labeling: injection of biocytin and its detection with avidin conjugates". *Journal of Neuroscience Methods*, 25(1), pp. 1–11.

Ibanez, L, W Schroeder, L Ng, and J Cates (2005). *The ITK Software Guide*. Kitware Inc.

Irimia, A, MC Chambers, CM Torgerson, M Filippou, DA Hovda, JR Alger, G Gerig, AW Toga, PM Vespa, R Kikinis, and JD Van Horn (2012). "Patient-tailored connectomics visualization for the assessment of white matter atrophy in traumatic brain injury". *Frontiers in Neurology*, 3(10), pp. 1–21.

Jeong, WK, J Beyer, M Hadwiger, R Blue, C Law, A Vázquez-Reina, RC Reid, J Lichtman, and H Pfister (2010). "Ssecrett and NeuroTrace: interactive visualization and analysis tools for large-scale neuroscience data sets". *IEEE Computer Graphics and Applications*, 30(3), pp. 58–70.

Jianu, R, C Demiralp, and DH Laidlaw (2012). "Exploring brain connectivity with two-dimensional neural maps". *IEEE Transactions on Visualization and Computer Graphics*, 18(6), pp. 978–987.

Jung, C and C Kim (2010). "Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization". *IEEE Transactions on Biomedical Engineering*, 57(10), pp. 2600–2604.

Jung, C, C Kim, SW Chae, and S Oh (2010). "Unsupervised segmentation of overlapped nuclei using bayesian classification". *IEEE Transactions on Biomedical Engineering*, 57(12), pp. 2825–2832.

Kabsch, W (1978). "A discussion of the solution for the best rotation to relate two sets of vectors". *Acta Crystallographica Section A*, 34(5), pp. 827–828.

Kaiser, M (2011). "A tutorial in connectome analysis: topological and spatial features of brain networks". *NeuroImage*, 57(3), pp. 892–907.

Kandel, ER, JH Schwarz, and TM Jessel (2000). *Principles of neural science*. 4th Ed. McGraw-Hill.

Kaufman, DL, JF McGinnis, NR Krieger, and AJ Tobin (1986). "Brain glutamate decarboxylase cloned in lambda gt-11: fusion protein produces gamma-aminobutyric acid". *Science*, 232(4754), pp. 1138–1140.

Kaynig, V, A Vazquez-Reina, S Knowles-Barley, M Roberts, TR Jones, N Kasthuri, E Miller, J Lichtman, and H Pfister (2013). *Large-scale automatic reconstruction of neuronal processes from electron microscopy Images*. http://arxiv.org/pdf/1303.7186v1.pdf.

Keller, A and GC Carlson (1999). "Neonatal whisker clipping alters intracortical, but not thalamocortical projections, in rat barrel cortex". *Journal of Comparative Neurology*, 412(1), pp. 83–94.

Kerr, JND, CPJ De Kock, DS Greenberg, RM Bruno, B Sakmann, and F Helmchen (2007). "Spatial organization of neuronal population responses in layer 2/3 of rat barrel cortex". *Journal of Neuroscience*, 27(48), pp. 13316–13328.

Kharma, N, H Moghnieh, J Yao, YP Guo, A Abu-Baker, J Laganiere, G Rouleau, and M Cheriet (2007). "Automatic segmentation of cells from microscopic imagery using ellipse detection". *Image Processing, IET*, 1(1), pp. 39–47.

Kirchner, S (2003). "Ein Approximationsalgorithmus zur Berechnung der Ähnlichkeit dreidimensionaler Punktmengen". Diplomarbeit. Humboldt-Universität zu Berlin.

— (2007). "An FPTAS for computing the similarity of three-dimensional point sets". *International Journal of Computational Geometry & Applications*, 17(2), pp. 161–174.

Kleinfeld, D, A Bharioke, P Blinder, DD Bock, KL Briggman, DB Chklovskii, W Denk, M Helmstaedter, JP Kaufhold, WCA Lee, HS Meyer, KD Micheva, M Oberlaender, S Prohaska, RC Reid, SJ Smith, S Takemura, PS Tsai, and B Sakmann (2011). "Large-scale automated histology in the pursuit of connectomes". *The Journal of Neuroscience*, 31(45), pp. 16125–16138.

Al-Kofahi, Y, W Lassoued, W Lee, and B Roysam (2010). "Improved automatic detection and segmentation of cell nuclei in histopathology images". *IEEE Transactions on Biomedical Engineering*, 57(4), pp. 841–52.

Kozloski, J, K Sfyrakis, S Hill, F Schürmann, C Peck, and H Markram (2008). "Identifying, tabulating, and analyzing contacts between branched neuron morphologies". *IBM Journal of Research and Development*, 52(1/2), pp. 43–55.

Kremer, JR, DN Mastronarde, and JR McIntosh (1996). "Computer visualization of three-dimensional image data using IMOD". *Journal of Structural Biology*, 116(1), pp. 71–76.

Kuß, A, M Gensel, B Meyer, VJ Dercksen, and S Prohaska (2010). "Effective techniques to visualize filament-surface relationships". *Computer Graphics Forum*, 29(3), pp. 1003–1012.

Lang, S, VJ Dercksen, B Sakmann, and M Oberlaender (2011). "Simulation of signal flow in three-dimensional reconstructions of an anatomically realistic neuronal network in rat vibrissal cortex". *Neural Networks*, 24(9), pp. 998–1011.

Lee, SC and P Bajcsy (2008). "Trajectory fusion for three-dimensional volume reconstruction". *Computer Vision and Image Understanding*, 110(1), pp. 19–31.

Lefort, S, C Tomm, JC Floyd Sarria, and CCH Petersen (2009). "The excitatory neuronal network of the C2 barrel column in mouse primary somatosensory cortex". *Neuron*, 61(2), pp. 301–316.

Liang, YD, BA Barsky, and M Slater (1992). *Some improvements to a parametric line clipping algorithm.* Tech. rep. UCB/CSD-92-688. EECS Department, University of California, Berkeley.

Lin, CY, KL Tsai, SC Wang, CH Hsieh, HM Chang, and AS Chiang (2011). "The Neuron Navigator: exploring the information pathway through the neural maze". In: *IEEE Pacific Visualization Symposium.* Hong Kong, pp. 35–42.

Lin, G, U Adiga, K Olson, JF Guzowski, CA Barnes, and B Roysam (2003). "A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks". *Cytometry Part A*, 56(1), pp. 23–36.

Lin, G, MK Chawla, K Olson, CA Barnes, JF Guzowski, C Bjornsson, W Shain, and B Roysam (2007). "A multi-model approach to simultaneous segmentation and classification of heterogeneous populations of cell nuclei in 3D confocal microscope images". *Cytometry Part A*, 71(9), pp. 724–736.

Lin, G, MK Chawla, K Olson, JF Guzowski, CA Barnes, and B Roysam (2005). "Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei". *Cytometry Part A*, 63(1), pp. 20–33.

Liu, T, G Li, J Nie, A Tarokh, X Zhou, L Guo, J Malicki, W Xia, and STC Wong (2008). "An automated method for cell detection in zebrafish". *Neuroinformatics*, 6(1), pp. 5–21.

Long, F, H Peng, and E Myers (2007). "Automatic segmentation of nuclei in 3D microscopy images of C. Elegans". In: *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro.* Arlington, VA, pp. 536–539.

Luisi, J, A Narayanaswamy, Z Galbreath, and B Roysam (2011). "The FARSIGHT Trace Editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions". *Neuroinformatics*, 9(2–3), pp. 305–315.

Luo, B (2001). "Statistical methods for point pattern matching". PhD thesis. University of York.

Luo, L, EM Callaway, and K Svoboda (2008). "Genetic dissection of neural circuits". *Neuron*, 57(5), pp. 634–660.

Maintz, JBA and MA Viergever (1998). "A survey of medical image registration". *Medical Image Analysis*, 2(1), pp. 1–36.

Malpica, N, CO De Solórzano, JJ Vaquero, A Santos, I Vallcorba, JM García-Sagredo, and F Del Pozo (1997). "Applying watershed algorithms to the segmentation of clustered nuclei". *Cytometry*, 28(4), pp. 289–297.

Margrie, TW, M Brecht, and B Sakmann (2002). "In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain". *Pflügers Archiv : European Journal of Physiology*, 444(4), pp. 491–498.

Markram, H (2006). "The Blue Brain Project". *Nature Reviews Neuroscience*, 7(2), pp. 153–160.

Mayerich, D, C Bjornsson, J Taylor, and B Roysam (2011). "Metrics for comparing explicit representations of interconnected biological networks". In: *IEEE Symposium on Biological Data Visualization*. Providence, RI, pp. 79–86.

McGuffin, MJ and I Jurisica (2009). "Interaction techniques for selecting and manipulating subgraphs in network visualizations". *IEEE Transactions on Visualization and Computer Graphics*, 15(6), pp. 937–944.

McIntosh, JR, ed. (2007). *Cellular electron microscopy (Methods in Cell Biology, vol. 79)*. Academic Press, San Diego, CA.

Meijering, E (2010). "Neuron tracing in perspective". *Cytometry Part A*, 77(7), pp. 693–704.

Meyer, HS, R Egger, JM Guest, R Foerster, S Reissl, and M Oberlaender (2013). "Cellular organization of cortical barrel columns is whisker-specific". *Proceedings of the National Academy of Sciences of the United States of America*, 110(47), pp. 19113–19118.

Meyer, HS, VC Wimmer, M Hemberger, RM Bruno, CPJ De Kock, A Frick, B Sakmann, and M Helmstaedter (2010). "Cell type-specific thalamic innervation in a column of rat vibrissal cortex". *Cerebral Cortex*, 20(10), pp. 2287–2303.

Meyer, HS, VC Wimmer, M Oberlaender, CPJ De Kock, B Sakmann, and M Helmstaedter (2010). "Number and laminar distribution of neurons in a thalamocortical projection column of rat vibrissal cortex". *Cerebral Cortex*, 20(10), pp. 2277–2286.

Mikolajczyk, K and C Schmid (2005). "A performance evaluation of local descriptors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), pp. 1615–1630.

Mishchenko, Y (2009). "Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs". *Journal of Neuroscience Methods*, 176(2), pp. 276–289.

Mullen, RJ, CR Buck, and AM Smith (1992). "NEUN, a neuronal specific nuclear-protein in vertebrates". *Development*, 116(1), pp. 201–211.

Myatt, DR, T Hadlington, GA Ascoli, and SJ Nasuto (2012). "Neuromantic – from semi-manual to semi-automatic reconstruction of neuron morphology". *Frontiers in Neuroinformatics*, 6(4).

Myronenko, A and X Song (2010). "Point set registration: Coherent point drift". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), pp. 2262–2275.

National Academy of Engineering. *Grand challenges for engineering*. http://www.engineeringchallenges.org. Accessed Sept. 13, 2012.

Nielsen, J and RL Mack (1994). *Usability inspection methods*. New York: John Wiley & Sons, Inc.

Oberlaender, M (2009). "Three-dimensional reengineering of neuronal microcircuits. The cortical column in silico." PhD thesis. University of Heidelberg, Germany.

Oberlaender, M, ZSRM Boudewijns, T Kleele, HD Mansvelder, B Sakmann, and CPJ De Kock (2011). "Three-dimensional axon morphologies of individual layer 5 neurons indicate cell type-specific intracortical pathways for whisker motion and touch".

Proceedings of the National Academy of Sciences of the United States of America, 108(10), pp. 4188–4193.

Oberlaender, M, PJ Broser, B Sakmann, and S Hippler (2009). "Shack-Hartmann wave front measurements in cortical tissue for deconvolution of large three-dimensional mosaic transmitted light brightfield micrographs". *Journal of Microscopy*, 233(2), pp. 275–289.

Oberlaender, M, RM Bruno, B Sakmann, and PJ Broser (2007). "Transmitted light brightfield mosaic microscopy for three-dimensional tracing of single neuron morphology". *Journal of Biomedical Optics*, 12(6), pp. 1–19.

Oberlaender, M, CPJ De Kock, RM Bruno, A Ramirez, HS Meyer, VJ Dercksen, M Helmstaedter, and B Sakmann (2012). "Cell type-specific three-dimensional structure of thalamocortical circuits in a column of rat vibrissal cortex". *Cerebral Cortex*, 22(10), pp. 2375–2391.

Oberlaender, M, VJ Dercksen, R Egger, M Gensel, B Sakmann, and HC Hege (2009). "Automated three-dimensional detection and counting of neuron somata". *Journal of Neuroscience Methods*, 180(1), pp. 147–160.

Oberlaender, M, A Ramirez, and RM Bruno (2012). "Sensory experience restructures thalamocortical axons during adulthood". *Neuron*, 74(4), pp. 648–655.

Oliveira, FPM and JMRS Tavares (2014). "Medical image registration: a review." *Computer Methods in Biomechanics and Biomedical Engineering*, 17(2), pp. 73–93.

Papazov, C, VJ Dercksen, H Lamecker, and HC Hege (2008). "Visualizing morphogenesis and growth by temporal interpolation of surface-based 3D atlases". In: *Proceedings of the 2008 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Paris, France, pp. 824–827.

Parekh, R and GA Ascoli (2013). "Neuronal morphology goes digital: a research hub for cellular and system neuroscience". *Neuron*, 77(6), pp. 1017–1038.

Parvin, B, Q Yang, J Han, H Chang, B Rydberg, and MH Barcellos-Hoff (2007). "Iterative voting for inference of structural saliency and characterization of subcellular events". *IEEE Transactions on Image Processing*, 16(3), pp. 615–623.

Peng, H, F Long, T Zhao, and E Myers (2011). "Proof-editing is the bottleneck of 3D neuron reconstruction: the problem and solutions". *Neuroinformatics*, 9(2–3), pp. 103–105.

Peng, H, Z Ruan, F Long, JH Simpson, and EW Myers (2010). "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets". *Nature Biotechnology*, 28(4), pp. 348–353.

Peters, A (1979). "Thalamic input to the cerebral cortex". *Trends in Neurosciences*, 2, pp. 183–185.

Petersen, CC and B Sakmann (2000). "The excitatory neuronal network of rat layer 4 barrel cortex". *The Journal of Neuroscience*, 20(20), pp. 7579–7586.

Petersen, CCH (2007). "The functional organization of the barrel cortex". *Neuron*, 56(2), pp. 339–355.

Pfister, H, V Kaynig, CP Botha, S Bruckner, VJ Dercksen, HC Hege, and JBTM Roerdink (2014). "Visualization in connectomics". In: *Scientific Visualization - Un-*

certainty, Multifield, Biomedical, and Scalable Visualization. Ed. by CD Hansen, M Chen, CR Johnson, AE Kaufman, and H Hagen. Springer, pp. 221–245.

Pinault, D (1996). "A novel single-cell staining procedure performed in vivo under electrophysiological control: morpho-functional features of juxtacellularly labeled thalamic cells and other central neurons with biocytin or Neurobiotin". *Journal of Neuroscience Methods*, 65(2), pp. 113–136.

Qi, X, F Xing, DJ Foran, and L Yang (2012). "Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set". *IEEE Transactions on Biomedical Engineering*, 59(3), pp. 754–765.

Ragan, T, LR Kadiri, KU Venkataraju, K Bahlmann, J Sutin, J Taranda, I Arganda-Carreras, Y Kim, HS Seung, and P Osten (2012). "Serial two-photon tomography for automated ex vivo mouse brain imaging." *Nature Methods*, 9(3), pp. 255–258.

Raman, S, CA Maxwell, MH Barcellos-Hoff, and B Parvin (2007). "Geometric approach to segmentation and protein localization in cell culture assays". *Journal of Microscopy*, 225(Pt 1), pp. 22–30.

Rangarajan, A, H Chui, and F Bookstein (1997). "The Softassign Procrustes matching algorithm". In: *Information Processing in Medical Imaging*. Ed. by G Duncan, James and Gindi. Vol. 1230. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 29–42.

Rockel, AJ, RW Hiorns, and TP Powell (1980). "The basic uniformity in structure of the neocortex". *Brain: A Journal of Neurology*, 103(2), pp. 221–244.

Ropireddy, D, SE Bachus, and GA Ascoli (2012). "Non-homogeneous stereological properties of the rat hippocampus from high-resolution 3D serial reconstruction of thin histological sections". *Journal of Neuroscience*, 205, pp. 91–111.

Schoonover, CE, JC Tapia, VC Schilling, V Wimmer, R Blazeski, W Zhang, Ca Mason, and RM Bruno (2014). "Comparative strength and dendritic organization of thalamocortical and corticocortical synapses onto excitatory layer 4 neurons". *The Journal of Neuroscience*, 34(20), pp. 6746–58.

Schubert, D, R Kötter, HJ Luhmann, and JF Staiger (2006). "Morphology, electrophysiology and functional input connectivity of pyramidal neurons characterizes a genuine layer Va in the primary somatosensory cortex". *Cerebral Cortex*, 16(2), pp. 223–236.

Sigg, C, T Weyrich, M Botsch, and M Gross (2006). "GPU-based ray-casting of quadratic surfaces". In: *Proceedings IEEE/Eurographics Symposium on Point-Based Graphics*. Boston, MA, pp. 59–65.

Skiena, SS (1998). *The algorithm design manual*. Springer Verlag New York, Inc.

Soille, P (1999). *Morphological image analysis: principles and applications*. Berlin Heidelberg: Springer, p. 316.

Sommer, C, C Straehle, U Koethe, and FA Hamprecht (2011). "Ilastik: interactive learning and segmentation toolkit". In: *8th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. Chicago, IL, pp. 230–233.

Sporns, O, G Tononi, and R Kötter (2005). "The Human Connectome: a structural description of the human brain". *PLoS Computational Biology*, 1(4), e42.

Stalling, D, M Westerhoff, and HC Hege (2005). "Amira: a highly interactive system for visual data analysis". In: *The Visualization Handbook*. Ed. by CD Hansen and CR Johnson. Elsevier. Chap. 38, pp. 749–767.

Sterio, DC (1984). "The unbiased estimation of number and sizes of arbitrary particles using the disector". *Journal of Microscopy*, 134(Pt 2), pp. 127–136.

Svoboda, K (2011). "The past, present, and future of single neuron reconstruction". *Neuroinformatics*, 9(2–3), pp. 97–98.

Szeliski, R (2006). "Image alignment and stitching: a tutorial". *Foundations and Trends in Computer Graphics and Vision*, 2(1), pp. 1–104.

Varga, Z, H Jia, B Sakmann, and A Konnerth (2011). "Dendritic coding of multiple sensory inputs in single cortical neurons in vivo". *Proceedings of the National Academy of Sciences of the United States of America*, 108(37), pp. 15420–15425.

Vincent, L (1993). "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms". *IEEE Transactions on Image Processing*, 2(2), pp. 176–201.

Vincent, L and ER Dougherty (1994). "Morphological segmentation for textures and particles". In: *Digital Image Processing Methods*. New York: Marcel Dekker. Chap. 2, pp. 43–102.

Wearne, SL, A Rodriguez, DB Ehlenberger, AB Rocher, SC Henderson, and PR Hof (2005). "New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales". *Neuroscience*, 136(3), pp. 661–680.

Weber, B, G Greenan, S Prohaska, D Baum, HC Hege, T Müller-Reichert, AA Hyman, and JM Verbavatz (2012). "Automated tracing of microtubules in electron tomograms of plastic embedded samples of Caenorhabditis elegans embryos". *Journal of Structural Biology*, 178(2), pp. 129–138.

Williams, RW and K Herrup (1988). "The control of neuron number". *Annual Review of Neuroscience*, 11(1), pp. 423–453.

Williams, RW and P Rakic (1988). "Three-dimensional counting: an accurate and direct method to estimate numbers of cells in sectioned material". *Journal of Comparative Neurology*, 278(3), pp. 344–352.

Wills, G (1996). "524,288 Ways to Say 'This is Interesting'". In: *Proceedings IEEE Symposium on Information Visualization*. San Francisco, CA, pp. 54–61.

Wimmer, VC, RM Bruno, CPJ De Kock, T Kuner, and B Sakmann (2010). "Dimensions of a projection column and architecture of VPM and POm axons in rat vibrissal cortex". *Cerebral Cortex*, 20(10), pp. 2265–2276.

Wong-Riley, M (1979). "Changes in the visual system of monocularly sutured or enucleated cats demonstrable with cytochrome oxidase histochemistry". *Brain research*, 171(2), pp. 11–28.

Woolsey, TA and H Van Der Loos (1970). "The structural organization of layer IV in the somatosensory region (SI) of mouse cerebral cortex. The description of a cortical field composed of discrete cytoarchitectonic units". *Brain Research*, 17(2), pp. 205–242.

Wu, HS, J Barba, and J Gil (2000). "Iterative thresholding for segmentation of cells from noisy images". *Journal of Microscopy*, 197(Pt 3), pp. 296–304.

Zitová, B and J Flusser (2003). "Image registration methods: a survey". *Image and Vision Computing*, 21(11), pp. 977–1000.

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Berlin 2015,

Vincent Jasper Dercksen