Axon Morphology Analysis: from Image Processing to Modelling

Alejandro Mottini

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES, INFORMATION TECHNOLOGIES AND COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

from the University of Nice - Sophia Antipolis

Specialty : Control systems, signal and image processing

Defended by

Alejandro Ricardo Mottini d'Oliveira

Axon Morphology Analysis: from Image Processing to Modelling

Thesis Advisor: Xavier Descombes

prepared at Morpheme Team, INRIA Sophia Antipolis/I3S/IBV

Defended on September 30, 2014

Jury:

Reviewers :	Anuj Srivastava	-	Florida State University, USA
	François Lauze	-	University of Copenhagen, Denmark
Examinators :	Charles Kervrann	-	INRIA Rennes, France
	Florence Besse	-	Institut de Biologie de Valrose, France
Advisor :	Xavier Descombes	-	INRIA Sophia Antipolis, France
President :	Michèle Studer	-	Institut de Biologie de Valrose, France

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS
ÉCOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION
ET DE LA COMMUNICATION

THÈSE

pour l'obtention du grade de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis

Mention : AUTOMATIQUE, TRAITEMENT DU SIGNAL ET DES IMAGES

présentée et soutenue par

Alejandro Ricardo MOTTINI d'OLIVEIRA

Analyse de la Morphologie Axonale: du Traitement des Images a la Modélisation

Thèse dirigée par: Xavier DESCOMBES

préparée dans l'Equipe MORPHEME, INRIA Sophia Antipolis/I3S/IBV

soutenue le 30 Septembre, 2014

Jury:

Rapporteurs :	Anuj SRIVASTAVA	-	Florida State University, USA
	François LAUZE	-	University of Copenhagen, Denmark
Examinateurs :	Charles KERVRANN	-	INRIA Rennes, France
	Florence BESSE	-	Institut de Biologie de Valrose, France
Directeur :	Xavier DESCOMBES	-	INRIA Sophia Antipolis, France
Président :	Michèle STUDER	-	Institut de Biologie de Valrose, France

Acknowledgments

I wish to thank my advisor Xavier Descombes and collaborator Florence Besse for their continuous support and direction during my thesis. Their guidance and advice were fundamental to my work. It has been a great pleasure working with you both for the past three years.

Importantly, I would like to convey my thanks and gratitude to the rest of my thesis committee : Anuj Srivastava, François Lauze, Charles Kervrann and Michèle Studer for their time spent reading my thesis, useful insights and constructive questions.

I would also like to thank all my lab-mates from the Morpheme team for sharing with my these last three years of my life. I have enjoyed meeting and working together with everyone. Thank you for all the support, help, and fun times.

During my time here I was fortunate to meet many great people from different teams. I would like to thank them all. Without them these last three years would have been a lot less interesting and enjoyable.

Finally, I would like to thank my family and Veronika for their love, support and encouragement. I truly would not have been able to succeed without having them by my side. This thesis is dedicated to you.
Thank you!

Axon Morphology Analysis: from Image Processing to Modelling

Abstract: The morphological analysis of axonal trees is an important problem in neuroscience. It has been shown that the morphological characteristics of these structures provide information on their functioning and allows the characterization of pathological states. For example, the number and size of branches have a direct impact on the number of partners that a neuron can have. Therefore, it is of great importance to develop methods to analyze their shape and to quantify differences between structures.

In this thesis we propose a method for the comparison of axonal trees that takes into account both topological and geometrical information. Using this method, which is based on the Elastic Shape Analysis Framework, we can compute the geodesic path between two axons and the mean shape of a population of trees. In addition, we derive a classification scheme based on this metric and compare it with state of the art approaches.

Finally, we propose a 2D discrete stochastic model for the simulation of axonal biogenesis. The model is defined by a third order Markov Chain and considers two main processes: the growth process that models the elongation and shape of the neurites and the bifurcation process that models the generation of branches. The growth process depends, among other variables, on an external attraction field generated by a chemoattractant molecule secreted by the target area.

Both techniques were validated on a database of real fluorescent confocal microscopy images of single neurons within intact adult *Drosophila* fly brains. Both normal neurons and neurons in which certain genes were inactivated have been considered (two mutations).

Results show that the proposed axon comparison method obtains better results that other methods found in the literature, and that the stochastic model parameter values provide information about the axon growth properties of the populations.

Keywords: morphological analysis, elastic shape analysis, axonal biogenesis, confocal microscopy

Analyse de la Morphologie Axonale: du Traitement des Images a la Modélisation

Résumé: L'analyse de la morphologie axonale est un problème important en neuroscience. Diverses études ont montré que les caractéristiques morphologiques de ces structures donnent des informations sur son fonctionnement et permettent la caractérisation d'états pathologiques. Par exemple, le nombre et la taille des branches de l'axone ont un impact direct sur le nombre de connections que le neurone peut établir. En conséquence, il est très important de développer des méthodes pour étudier leurs formes et quantifier leurs différences structurelles.

Dans cette thèse on propose une méthode pour la comparaison des arbres axonaux qui inclue des informations topologiques et géométriques sur les arbres. La méthode est fondée sur la théorie des formes élastiques. Avec cette approche, nous pouvons exhiber le chemin géodésique entre deux formes et la forme moyenne d'un ensemble d'échantillons. En outre, nous proposons un schéma de classification à partir de cette métrique que nous comparons à l'état de l'art.

Finalement, nous proposons un modèle stochastique discret pour la simulation de la croissance axonale. Le modèle est défini par une chaîne de Markov d'ordre 3. Il considère 2 processus principaux: la croissance qui modélise l'élongation et la forme de l'axone, et le processus de bifurcation qui modélise la génération des branches. Le processus de croissance dépend de différentes variables, dont un champ externe d'attraction généré par certaines molécules dans l'environnement.

Les deux techniques proposées ont été validées sur une base d'images de microscopie confocale de neurones chez la *Drosophile*. Des neurones normaux et modifiés génétiquement ont été considérés (deux mutations)

Les résultats montrent que la méthode de comparaison proposée fournit de meilleurs résultats que les méthodes décrites dans la littérature. De plus, les paramètres du modèle stochastique donnent des informations sur le processus de croissance de chaque population d'axones.

Mots Clés: analyse morphologique, théorie des formes élastiques, microscopie confocale

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Context

The brain is the most complex biological organ, both in terms of structure and function. Microscopically, it is composed of billions of neurons that interconnect, forming what are known as neural circuits. These neuronal connections are essential to the processing and storage of information.

Neurons communicate with one another by means of two types of protrusions (neurites) called dentrites and axons. Dendrites receive the information arriving to the neuron, while the axons are in charge of transmitting the information to other neighboring cells. While a neuron can have many dendrites, they always posses only one axon. This long filamentous structure is able to connect a neuron to distant parts of the brain or body targeting specific cells.

During the developmental stage of the nervous system, neurons extend their axons in order to form the necessary neural circuits. Growing axons are guided to their target by specialized structures known as growth cones. These structures are present at the axon tip, and are able to sense chemical signals present in their environment. The paths taken by the growth cones depend on these signals, which can be either attractive or repulsive.

Once an axon reaches its destination, it begins to branch in order to connect to multiple targets, all of which will receive the same signals coming from the neuron body (soma). There are different branching configurations adapted to different connection requirements. Certain axons can have all their branches at the terminal

part (the part furthest away from the neuron body), while others can have multiple branches that sprout throughout the axon filament.

It is known that certain morphological characteristics are tightly linked to the neuron ability to function properly. For example, the number of branches in an axon is related to the number of connections a neuron can make. Thus, these morphological characteristics are also able to characterize pathological states. It is known that certain neurological diseases such as Fragile X Syndrome or Spinal Muscular Atrophy are related to unusual morphological characteristics of the neurons in the brain. Therefore, the morphological analysis of neurons, and in particular, of axons, is of great importance.

To analyze these morphological characteristics, researchers depend on neuronal images that can be acquired using different modalities. One of the most relevant is confocal microscopy. This imaging technique offers several advantages over conventional optical microscopy, and when combined with the fluorescence phenomenon, it is capable of obtaining sharp 3D images of thick specimens that would otherwise be impossible to study.

During the course of this thesis we have worked in collaboration with biologists from the Institut de Biologie de Valrose. They have provided us with a large image database of axons taken from neurons of the *Drosophila* fly brain. Both normal and pathological neurons have been considered. The pathological neurons were obtained by mutating certain genes related to neurological diseases in humans. These images will be the subject of study throughout this thesis, and will be used to validate the different proposed algorithms.

Given a set of images, they need to be processed in order to extract the information in them. The first step necessary for their processing is the segmentation. Currently, there is a wide variety of segmentation methods, each one adapted for different situations. Since none of the reviewed methods obtained satisfactory results on our images, we have proposed a new one.

Once the morphological information has been extracted from the images, we need to analyze it and quantify the differences between different populations of axons. Different approaches have been proposed. Some are based on distance functions that consider topological or geometrical information (or both), while other rely on directly comparing certain morphological features such as length or number of branches. We propose a new approach for the comparison of tree-like structures based on the Elastic Shape Analysis Framework, which we apply to the comparison and classification of different populations of axons. Moreover, we propose a method for the computation of the average shape of a populations of trees. The resulting mean shapes provide biologist with a visual way of analyzing the morphological

characteristics of a population.

A different approach to the morphological analysis problem is to use the segmented neurons as input of an axonal biogenesis model. A model can provide an insight into the underlying processes and natural parameters involved in the axonal morphogenesis. We propose a new discrete stochastic model that considers several biological processes (such as growth and branching). All the parameters involved in the model are estimated from the images. This parameter estimation scheme provides another tool to characterize and classify different populations.

The entire morphological analysis process is represented in Figure 1.1.

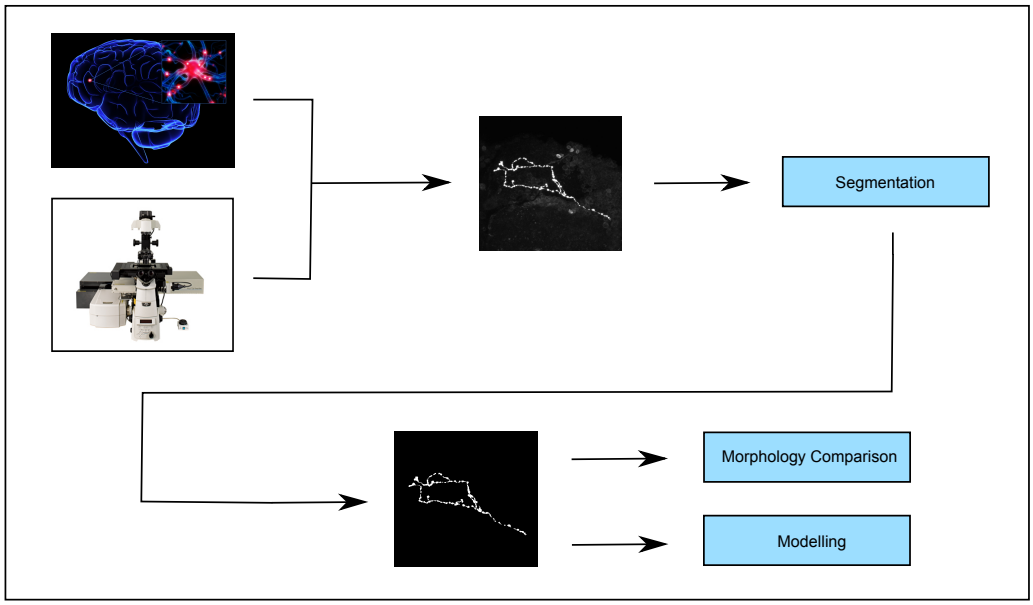


Figure 1.1: Schematic diagram of the morphological analysis process (images from http://scientopia.org/, http://www.nikoninstruments.com/ and http://www.wadsworth.org/).

1.2 Thesis Organization

1.2.1 Main Contributions

To summarize, four main contributions were proposed:

- A segmentation algorithm for confocal microscopy images of axonal trees. The method is based on a multi-step approach that combines filament enhancement, binarization, skeletonization and gap filling algorithms in a pipeline capable of extracting the axons from the images.

- A new distance for comparing tree-like shapes based on the Elastic Shape Analysis Framework. The distance takes into account both topological and geometrical information.

- An iterative algorithm for the computation of the mean shape of a population of trees based on our Elastic Shape Analysis distance.

- A discrete stochastic model for axonal tree biogenesis. The model is defined by a third order Markov Random Chain and considers several biological processes. In particular, the chemoattractant field to which the growth cones are exposed is modeled.

1.2.2 Manuscript Organization

This manuscript is organized along the published and submitted work on which it is based.

Chapter 2 presents the motivation behind this thesis and briefly describes the biological processes involved in the morphology of axonal trees.

Chapter 3 describes the image databases used throughout this thesis to evaluate the different proposed algorithms. We also briefly present the acquisition process and the subject of study (*Drosophilia*, commonly called fruit flies).

We continue on Chapter 4 with a description of the necessary steps to perform the morphological analysis of the axonal trees: segmentation, comparison and modeling. We briefly describe different methods found in the literature for each of these three steps. Some of these methods will letter be used as a benchmark for comparing our algorithms.

In Chapter 5 we detail our first contribution that consists on an automatic multi-step algorithm for the segmentation of axonal confocal microscopy images. The method was first proposed in [Mottini 2012].

Chapter 6 presents the two main contributions of this thesis that we have proposed in [Mottini 2013, Mottini 2014a]. We start by presenting our tree comparison method. Finally, we present a method capable of computing the mean shape of a population of trees. Although the method could potentially be applied to different types of trees, we concentrate on the comparison of axonal trees.

Chapter 7 details a discrete stochastic model for the simulation of axonal trees that we have proposed in [Mottini 2014b]. It considers two main processes: the growth process that models the elongation and shape of the neurites and the bifurcation process that models the generation of branches. The model is defined by a third order Markov Chain, whose parameters are estimated from confocal mi-

croscopy images. It can be used as an alternative way of analyzing the morphological properties of the different populations.

Finally, we conclude this thesis by summarizing the key contributions of this work and discussing perspectives in Chapter 8.

1.3 List of Publications

- **A. Mottini**, X. Descombes and F. Besse. From Curves to Trees: A Tree-like Shapes Distance Using the Elastic Shape Analysis Framework. *Neuroinformatics*, under review, 2014.
- **A. Mottini**, X. Descombes, F. Besse and E. Pechersky. Discrete Stochastic Model for the Generation of Axonal Trees. *In Proc. IEEE EMBC*, Chicago, USA, 2014.
- **A. Mottini**, X. Descombes and F. Besse. Axonal Tree Classification Using an Elastic Shape Analysis Based Distance. *In Proc. IEEE ISBI*, Beijing, China, 2014.
- **A. Mottini**, X. Descombes and F. Besse. Tree-like Shapes Distance Using the Elastic Shape Analysis Framework. *In Proc. British Machine Vision Conference*, Bristol, UK, 2013.
- **A. Mottini**, X. Descombes and F. Besse. Axon Extraction from Fluorescent Confocal Microscopy Images. *In Proc. IEEE ISBI*, Barcelona, Spain, 2012.

Chapter 2

Axonal Biogenesis

Contents

In this chapter we introduce some of the key biological concepts underlying the axonal growth process.

We start by describing the structure of neurons, with an emphasis on neuronal extensions. The extensions (axons and dendrites) are usually referred to as neurites, and present a specialized structure at their tips called growth cones. These structures are responsible for guiding developing neurites to their targets by sensing chemical signals present in the cellular environment.

An important step in the development of neuronal extensions is the branching step. Branching most frequently starts once the neurites reach their targets, and consists in the bifurcation of the main process into multiple branches. It promotes neuron interconnection, a process mediated by specialized cell junctions called synapses.

Given that the morphology of neuronal trees, and axonal trees in particular, is tightly linked with the number and type of connections a neuron can have, their morphological analysis provides an insight into their functioning and a way of characterizing possible pathological states.

2.1 Introduction

The nervous system is composed of hundreds of billions of interconnected neurons, a type of cell specialized in information transmission. Macroscopically, a neuron is

composed of its cell body (soma) and of its cellular extensions referred to as neurites. There are two types of extensions: dendrites and axons. In mammals, axons are typically long and thin, with an uniform width, On the other hand, dendrites are short and their width decrease with increased distance from the soma. While there is only one axon per neuron, many dendrites sprout from the soma (see Figure 2.1).

These cellular extensions allow neurons to connect to each other, thus establishing neuronal circuits that play an essential role in transmitting and storing information [Bassell 2000]. Dendrites receive the information arriving to the neuron, while the axons are in charge of transmitting the information to other neighboring cells.

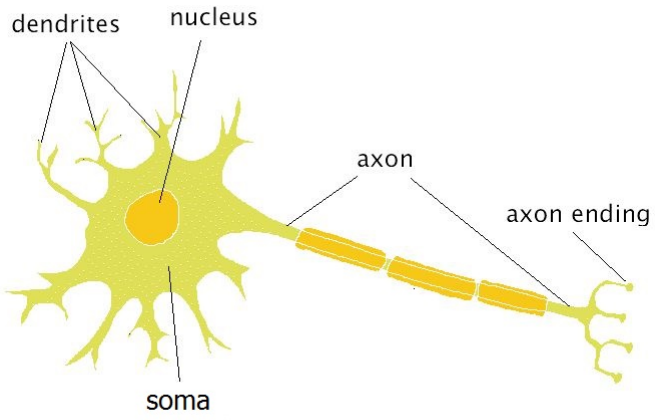


Figure 2.1: Schematic diagram of a neuron (from http://www.webspace.ship.edu).

The formation of functional neural networks relies on several processes, including axon growth and guidance, arborization of the neurites, recognition of the neuron connection targets and establishment of synaptic connections between them [Weiner 2013].

2.2 Axonal Growth

During the development of the nervous system, specialized structures (referred to as growth cones) guide growing axons to their targets.

Growth cones are specialized hand-shaped sensing structures present at the tips of the neurites, which are composed of three parts: the central and peripheral domains and the transition zone [Lowery 2009] (see Figure 2.2). The central domain corresponds to the proximal part of the growth cone (the part closest to the soma). This region has a dense network of microtubules (MT). Besides their structural role, MT are in charge of transmitting the necessary proteins to the cone [Hirokawa 2010].

The peripheral domain lies at the periphery of the growth cone and presents two types of cytoplasmic extensions: the filopodia and the lamellipodia. Filopodia are the guidance sensors, which give growth cones their characteristic hand shape (they are the fingers). In contrast, lamellipodia are located in between filopodia and are rich in F-acting networks.

Finally, the transition zone connects the central and peripheral domains. It presents an arc-shaped actin (a filamentous proteins) network that holds the MT in place [Burnette 2008].

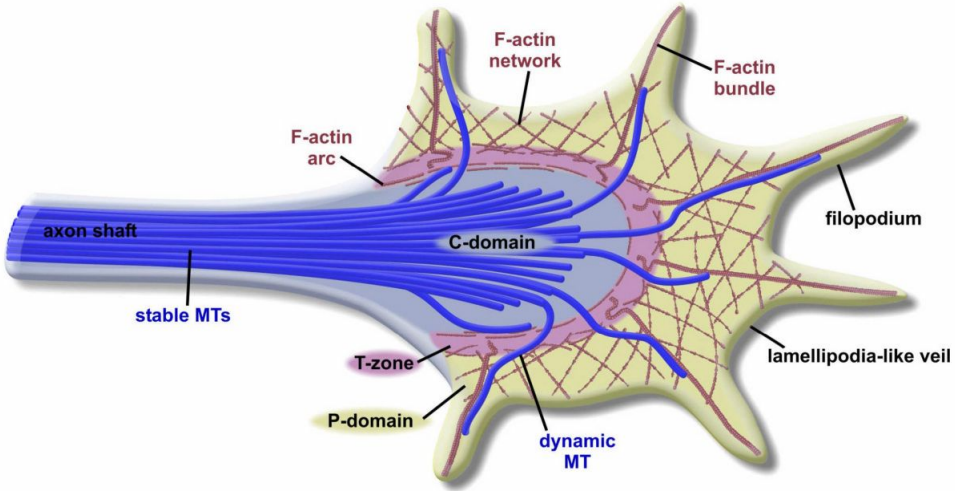


Figure 2.2: Schematic diagram of an growth cone (diagram taken and modified from Lowery and Van Vactor, 2009).

The growth cones constantly explore their environment to search for their targets by extending and retracting their protrusions. During the growth phase, a growth cone extends its filopodia. Then, the MT network extends and the central region established itself around the MT, thus allowing the axonal growth.

The path taken by the growth cones depends on chemical signals (guidance cues) present in their environment. These signals can be attractive or repulsive, and control in a non-autonomous way the navigation of the growth cones until they reach their targets [Campbell 2001].

2.2.1 Axonal guidance signals

The growth cones rely on the proteins present on its surface to detect and interpret the extracellular signals produced in the environment. These signals include guidance molecules, which bind to specific receptors in the growth cones and activate specific signaling pathways.

Guidance molecules can be attractive or repulsive, and long ranged (diffusible) or short-ranged (surface-bound). Attractive molecules stimulate the growth and reorientation of the cones, while repulsive molecules inhibit growth and can even induce the cone retraction. The cones can be simultaneously stimulated by a combination of both attractive and repulsive molecules. The signals detected at each filopidal tip are integrated to determine the growth cone behavior [Bruckert 2012] and guide it through the developing nervous system until it reaches its target.

Once the axons reach their target, the extension phase stops and the arborization phase begins. This phase produces the axonal branches that will later favor synaptic connections with other neurons.

2.3 Axonal Branching

Axons generate branches in order to connect to multiple targets and establish neuronal circuits. The branch formation process can be determined genetically, stimulated by neuron activity or induced by external factors [Bruckert 2012].

There are several branching processes mentioned in the literature, which differ in terms of shape, size and location [Gobson 2011]. The first type we will consider is arborization. It consists on the formation of secondary branches at the extremity of the axon. These secondary branches can also produce third level branches, thus creating a structure that resembles a tree. Examples of this type of branching are found in retinal ganglion cells [Gobson 2011]. A second type of morphology is generated by simple bifurcation. In this case, two daughter branches that grow away from each other in opposite direction are generated. Some examples are present in dorsal ganglion sensory neurons [Ozaki 1997]. These two branching types involve the division of the growth cone [Gallo 2011]. This division starts with the loss of cytoplasmic extensions (lamellipodia and filopodia) situated at the extremity of the cone and is characterized by cytoskeletal reorganization. This process ends with the formation of independent growth cones that will generate independent branches.

The final bifurcation case is referred to as collateral formation. This case involves the formation of daughter branches that sprout from the the central part of the axon shaft. These new branches extend orthogonally or obliquely from the main axon and project to targets that are different from that of the main axon. Examples of this branching type are present in sensory neurons in the spinal cord [Gobson 2011]. The different bifurcation types are schematized in Figure 2.3.

The strategies involved in the guidance and growth of the daughter branches are the same that for the main axon.

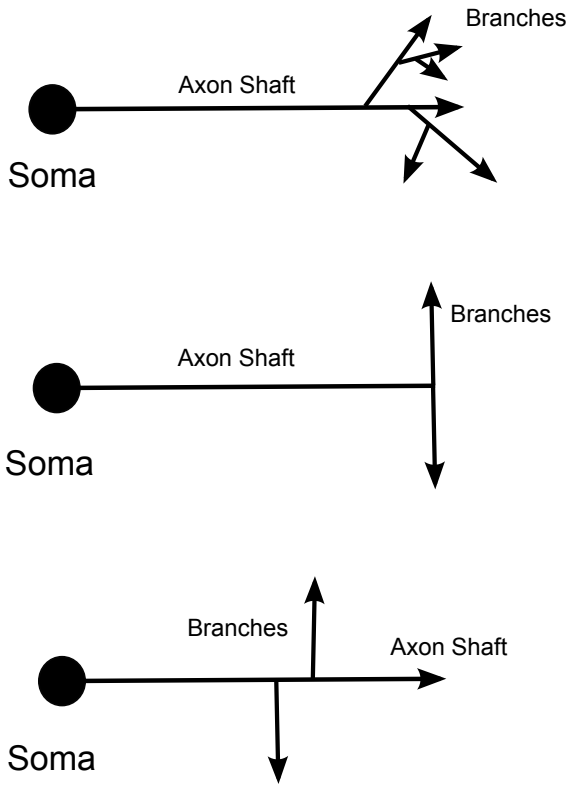


Figure 2.3: Schematic diagram of the different axonal branching processes: arborization (top), bifurcation (middle) and collateral formation (bottom).

2.4 Synaptogenesis

The branching and target recognition processes are followed by the synapse formation stage, which involves the interconnection of neuronal partners at specific locations [Colon-Ramos 2009].

Synapses are specialized cell junctions that control the information flow between connected neurons. Therefore, their study is critical to understand neuronal circuit activity and regulation.

There are two types of synapses: electrical and chemical synapses. Both types differ considerably in terms of structure and way of functioning. In addition, they both relay information using very different mechanisms (see Figure 2.4).

Electrical synapses are specialized intercellular connections between neurons (and other types of cells) that connect the cytoplasm of two neighboring cells. These connections, which are referred to as gap junctions, allow the bidirectional propagation of signals (including electrical stimuli). They provide the fastest method for information propagation between cells, and are known to be important for the synchronization of the electrical activity across networks.

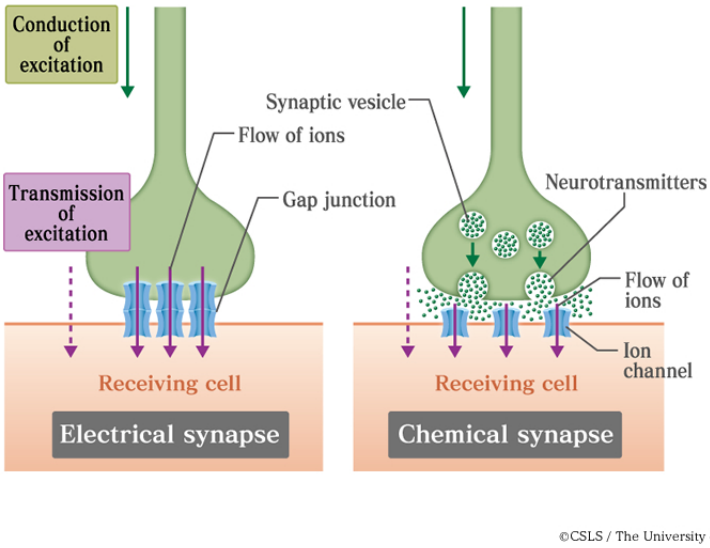


Figure 2.4: Electrical synapses transmit signals directly through the gap junctions, while chemical synapses transmit information indirectly using chemical signals (taken from http://csls-text.c.u-tokyo.ac.jp).

Chemical synapses are far more common. They use chemical signals (neurotransmitters) to transmit information. They are slower than electrical ones, and allow the flow of information in only one direction. Furthermore, they activity can be finely tuned in response to stimulation and experience, a process called synaptic plasticity.

2.5 Neuronal Circuits

When neurons interconnect with one-another, they generate complex networks that process information. These networks are commonly referred to as Neuronal circuits (see Figure 2.5). The correct establishment of the circuits depends on the successful completion of all the developmental steps explained in the previous sections [Lu 2009a].

Different morphological characteristics have a direct impact on the way neurons grow and interconnect, and are therefore tightly linked to the circuit anatomical and functional properties [Brown 2008].

For example, the length and number of axon branching points (see Figure 2.6) are critical parameters as they determine the territory covered by the axonal tree and are thus closely related to the number of partners a given neuron can have. A

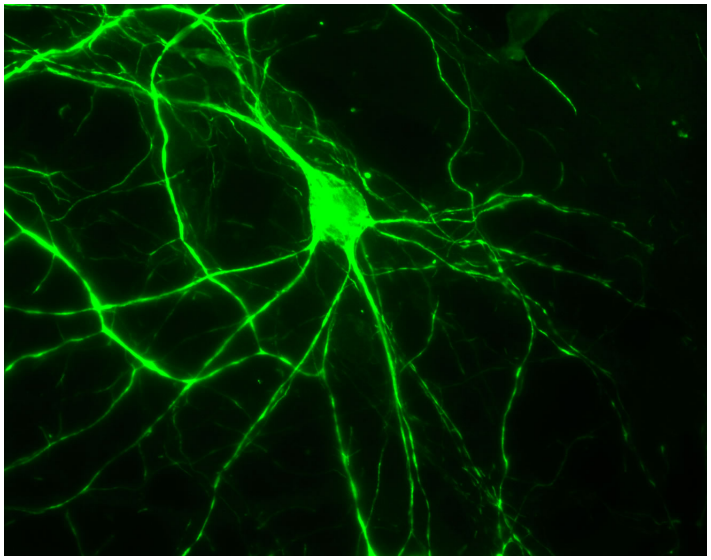


Figure 2.5: Population of neurons creating a neural circuit (from http://www.encorbio.com).

greater length also implies that the neuron can reach far away targets in the nervous system.

On the other hand, features such as maximum extent among paths in a tree have a functional significance since the greater the distance along a path, the greater the attenuation of the signal. Moreover, other morphological characteristics such as neurite diameter affects the way action potentials are generated and thus the way the information is processed.

Understanding how these parameters are controlled is a key step for understanding the mechanisms driving neuronal circuit establishment, both in the embryo and in adults during the regeneration process. For example, whereas the number of neurons does not significantly change in normal aging, some subtle morphological changes do appear in dendrites and axons [Kabaso 2009].

The study of axonal morphology is critical to our understanding of the nervous system and of neurological disorders. This is why throughout this thesis we will concentrate on understanding and characterizing the morphological differences between normal and pathological tissues.

2.6 Conclusions

The nervous system is organized in neural circuits, networks of neurons that interconnect in order to exchange information. The formation of these circuits depends on multiple developmental processes including axon growth and guidance, neurite

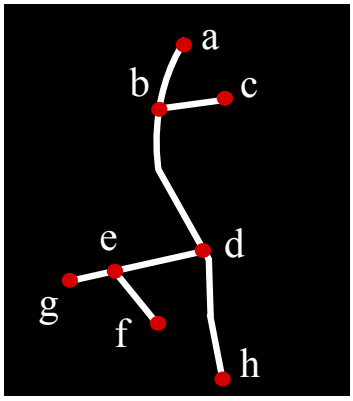


Figure 2.6: Axonal tree diagram. The main axon is the curve that passes through nodes (a,b,d,h). Branches are defined by nodes (b,c),(d,e,g) and the sub-branch by nodes (e,f). Both nodes b and d are referred to as first level branching points, while node e is a second level branching point.

branching and synaptogenesis.

Growing axons are guided to their target by specialized sensing structures present at their tips referred to as growth cones. The path taken by the growth cones depends on chemical signals present in their environment. These signals can be attractive or repulsive.

Once the axons reach their target, they begin to branch in order to connect to multiple targets. There are different branching configurations adapted to different connection requirements. The actual connection and communication between neurons is done though specialized cell junctions called Synapses.

The morphological analysis of axonal trees is an important problem in neuroscience since it is known that their morphology provides information on their functioning and allows the characterization of pathological states. In particular, the analysis of neuronal axonal topologies allows biologists to study the causes of certain neurological diseases such as Fragile X Syndrome or Spinal Muscular Atrophy.

Chapter 3

Image Database Description

Contents

In this chapter we discuss the neuronal images used throughout this thesis.

We start by introducing general concepts regarding the image acquisition process. Confocal microscopy is a technique capable of generating 3D image stacks of specimens at very high resolutions. When combined with Fluorescence imaging, these techniques provide an efficient way of analyzing cellular processes in living systems that would otherwise be impossible to study.

Then, we describe the two databases that will be used to validate the different methods proposed in Chapters 5,6 and 7. A special emphasis is put on the image database provided by our collaborators at the Institut de Biologie de Valrose. We describe the used study model (neurons from the the mushroom bodies of *Drosophilia*) and the general characteristics of the images.

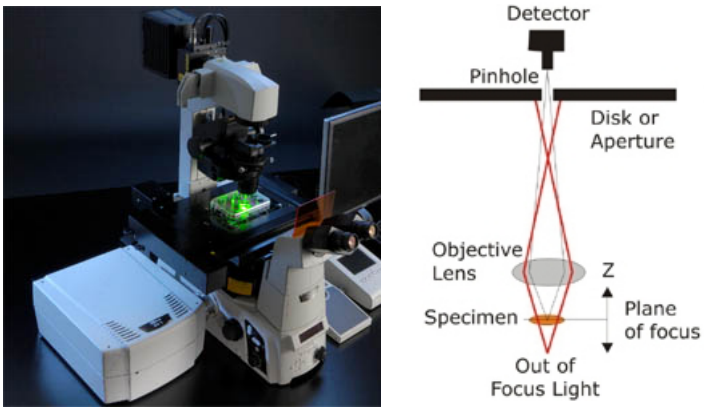
3.1 Introduction

Biologists currently possess different imaging techniques capable of producing images of biological systems at a cellular or subcellular level. Among them is confocal microscopy [Semwogerere 2014], a technique capable of generating sharp 3D images of specimens at very high resolutions.

3.1.1 Confocal Microscopy

Confocal microscopy is an imaging technique which was pioneered by Marvin Minsky (Harvard University) in 1955. This method creates an image by focusing a light ray sequentially across a specimen and then collecting the returning rays. Unlike classical microscopes, the specimen is illuminated one point at a time, thus avoiding most of the unwanted scattered light that would decrease the quality of the image. Moreover, the returning light passes through a second aperture (pinhole) that filters rays that are not coming directly from the focal point. The remaining light rays are collected and the image is gradually reconstructed. Finally, a three-dimensional (3D) image volume of the specimen is obtained by assembling different slices taken along the vertical axis [Semwogerere 2014].

This imaging technique offers several advantages over conventional optical microscopy, the most important being the ability to collect serial optical sections from thick specimens and the elimination of out-of-focus glare. In addition, confocal microscopy outperforms two-photon microscopy in imaging thin isolated structures (such as individual cells or axons) since it uses a shorter excitation wavelength which produces higher optical resolution images. In the biomedical sciences, confocal microscopy is usually used to image fixed cells or tissues that have been previously labeled with fluorescent dyes [Paddock 2014].



(a) (b)

Figure 3.1: Confocal microscope picture (left) and diagram (right) (from: http://www.nikoninstruments.eu).

3.1.2 Fluorescence Imaging

The phenomenon of fluorescence was first observed over a century ago. However, the realization that fluorescence could be applied for imaging purposes was recognized much later [Andersson-Engels 1997]. When combined with different imaging techniques, fluorescence provides an excellent technique to study cellular processes in living systems that would otherwise be impossible to study.

Fluorescence is the physical process by which a special molecule (a fluorescent molecule) is exited with photons, and some part of this absorbed energy is re-emitted afterwards in the form of light. More precisely, a light source (usually a laser) is used to excite a fluorescent molecule. This molecule absorbs the energy, which is used to increase its energy level to an excited state. Since this excited state is unstable, the molecule then relaxes to a lower energy level (see Figure 3.2). The difference in energy between the two levels is emitted in the form of light. In the context of biological imaging, this emitted light is captures by a microscope and an image is created.

There currently exist many fluorescent molecules, each one emitting light in a particular characteristic wavelength that is determined by its physical properties (and not by the wavelength of the absorbed light). The most commonly used molecule is the Green fluorescent protein (GFP), which was isolated in 1961 from the jellyfish Aequorea Victoria (see Figure 3.2) 3.3.

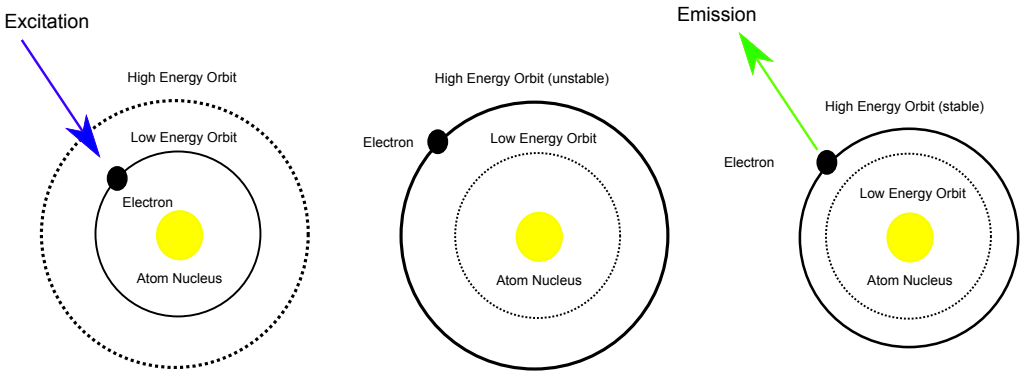


Figure 3.2: Diagram summaryzing the Fluorescence phenomenon.

Since the objects of study usually do not fluoresce by themselves, the fluorescent molecules need to be introduced in the biological structures of interest artificially. There are different techniques for accomplishing this. The gene responsible for the florescence can be artificially introduced in the cells of interest, either transiently by injection of transfection, or stably via trasgenesis.. Once the gene has been introduced, the cells will synthesize the florescent protein by themselves.

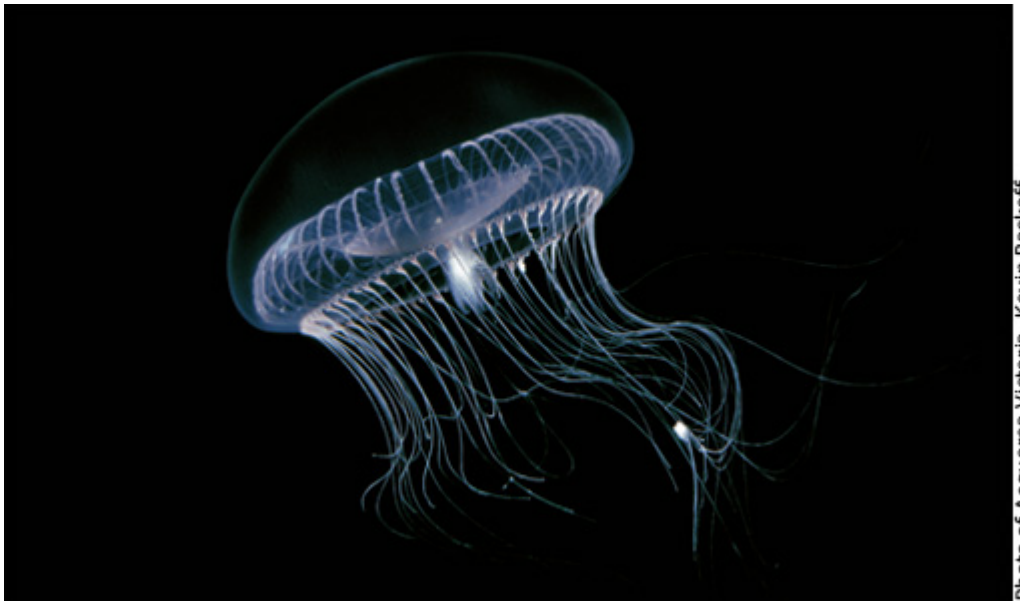


Figure 3.3: Picture of the Aequorea Victoria jellyfish (from: http://www.nobelprize.org/).

3.2 Image Databases

During this PhD we have worked in collaboration with biologists from the Institut de Biologie de Valrose (IBV). They have provided us with a set of images that we have used to validate the different methods described in the following chapters.

In addition, we have also used the public database Neuromorpho for the validation of the methods presented in Chapter 6.

3.2.1 IBV Database

3.2.1.1 Model of Study

Due to their short life cycle and the simplicity of their genome, *Drosophilia* flies are ideal subjects for investigating the basic mechanisms underlying neuronal morphology. Mutant flies, with defects on one or several genes, can be use to unveil the causes of different pathologies.

In particular, we work with neurons from the mushroom bodies (also called Kenyon cells). The neurons present in this region of the brain have been thoroughly studied before. Many techniques exist to modify their genes, making them a good model [Bruckert 2012]. The mushroom bodies are composed of two symmetrical stalks referred to as peduncles, which branch into district lobes [Crittenden 1998] (see Figure 3.4).

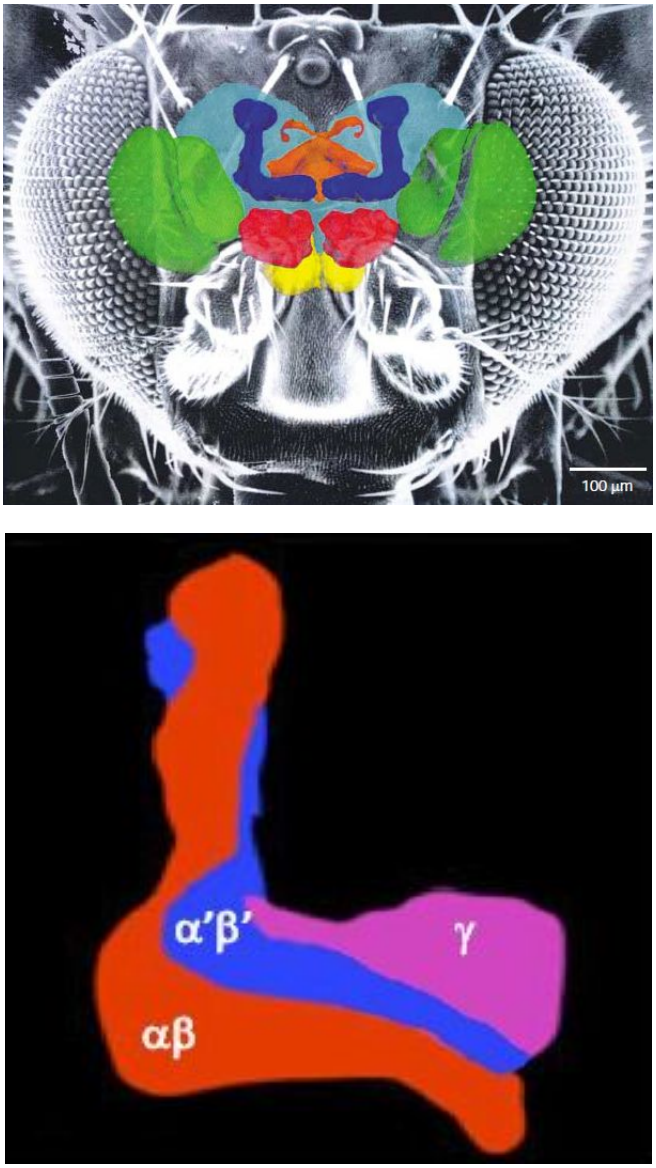


Figure 3.4: *Drosophilia* brain diagram (top, mushroom bodies in blue) and diagram of the mushroom body lobes (bottom) (from: [Heisenberg 2003] and [Bruckert 2012]). γ, $\alpha'\beta'$ and $\alpha\beta$ correspond to three sub populations with specific axonal projection patters.

Three populations of neurons have been defined depending on their specific axonal patterns: γ, $\alpha\beta$ and $\alpha'\beta'$ [Heisenberg 2003] (see Figure 3.5). In particular, we study the γ neurons, which are composed of a main axon that branches at different locations.

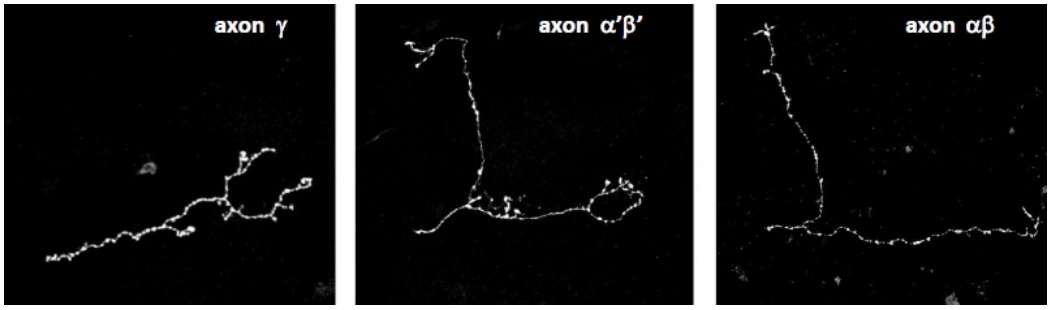


Figure 3.5: Characteristic morphology of each type of axons (γ, $\alpha\beta$ and $\alpha'\beta'$) (from [Bruckert 2012]).

3.2.1.2 Images

The biologist have fluorescently labeled single γ neurons within intact adult *Drosophila* fly brains, and have acquired 3D fluorescent confocal microscopy images of their axonal trees. Both normal neurons and neurons in which the function of the *imp* (mutant type 1) or *profilin* (mutant type 2) genes was inactivated were imaged. *imp* encodes a conserved RNA binding protein controlling subcellular mRNA transport and local protein synthesis, and is essential for axonal remodeling [Medioni 2014]. *profilin* encodes a regulator of the actin cytoskeleton involved in axonal pathfinding [Wills 1999]. Mutations in these two conserved genes have been linked to neurological pathologies [Donnelly 2011, C.H. Wu 2012].

The database is composed of 61 images: 20 normal, 24 type 1 mutant and 17 type 2 mutant. Each 3D image stack has a resolution of $0.093967 \times 0.093967 \times 0.814067$ μm and two channels. The morphology of single axonal trees is visible in the first channel, while the morphology of the overall neuronal structure in which axons are developing is visible in the second channel (see Figure 3.6).

All the axonal trees were manually segmented by an experienced biologist using Neuromantic 4.2.5 (see Figure 3.7). This is a very time consuming task. Therefore, we will propose an automatic segmentation method that will be presented in Chapter 5.

As we can appreciate in Figure 3.8, each population of axons sometimes present a distinctive morphological signature that differentiates them from the others. Normal axons are long and present several long branches and sub branches. On the other hand, mutant type 1 neurons are short and have few branches, and mutant type 2 axons are short but present many short branches and sub branches.

However, as we can appreciate in Figure 3.9, this is not always the case. There is a considerable variability in each population, and it is sometimes difficult to tell to which population an axonal tree belongs to. Understanding if and how the

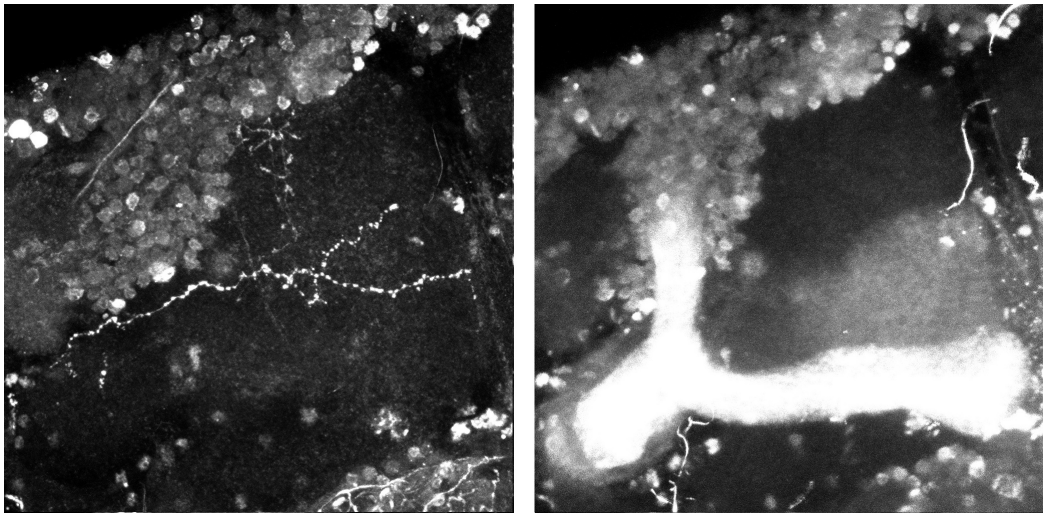


Figure 3.6: Channel 1 (left; GFP staining) and 2 (right; Fasciclin II staining) for a given image (maximum intensity projections).

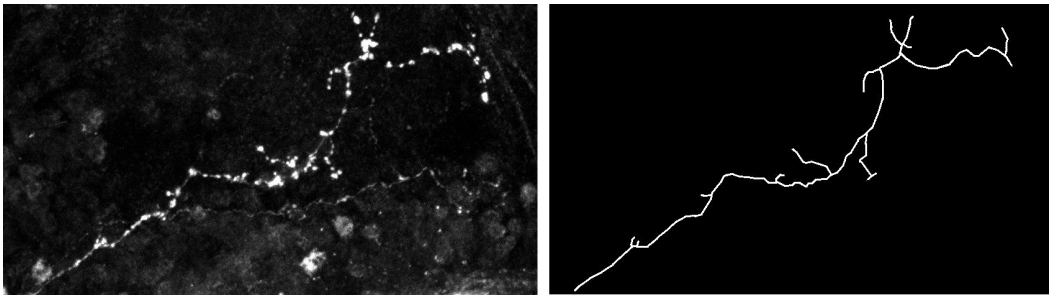


Figure 3.7: Original confocal microscopy image of an axonal tree (left) and its tracing (right) (maximum intensity projections).

populations are morphologically different will be studied in Chapters 6 and 7.

3.2.2 Neuromorpho Database

Neuromorpho [Ascoli 2007] is an open neuron reconstruction database available online at neuromorpho.org. It contains contributions from over 100 laboratories worldwide and is composed of over 10.000 reconstructions. Moreover, it consists on reconstructions from over 12 different species (from salamander to human), 16 main brain regions, and 30 cell types. This makes is the largest open database of neuron reconstructions.

This database provides the scientific community with a mutual benchmark on which to evaluate different algorithms.

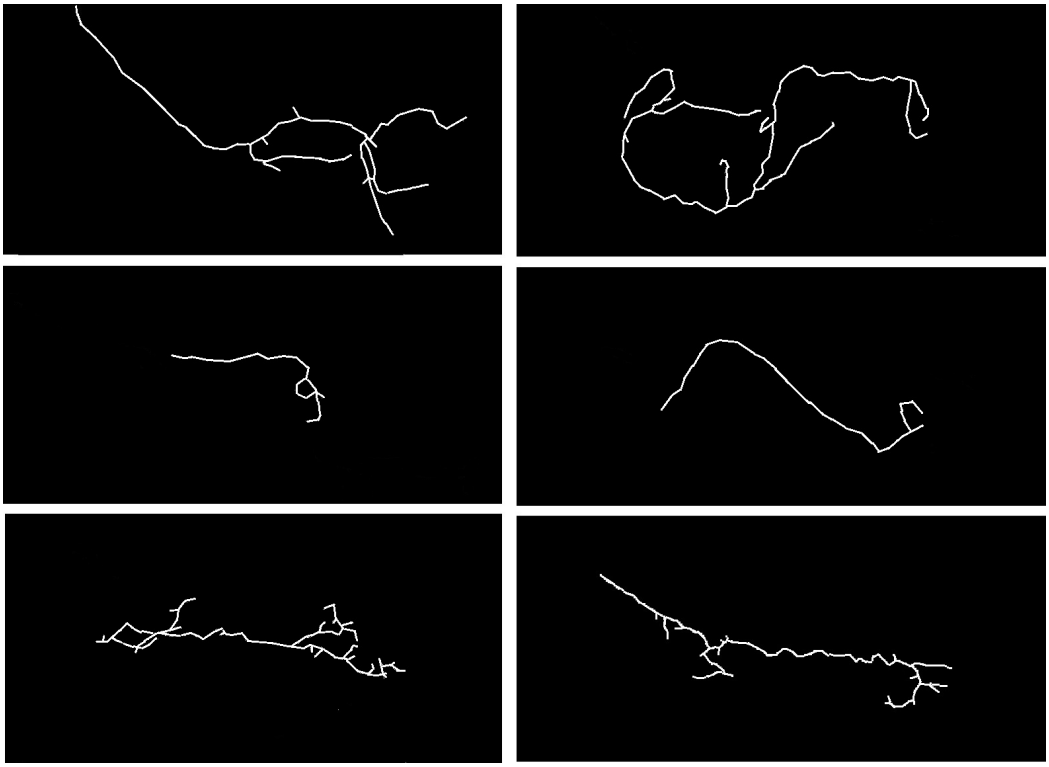


Figure 3.8: Some examples of normal (top), mutant type 1 (middle) and mutant type 2 (bottom) axonal trees. The examples present distinct morphological characteristics depending on the type of neuron (2D projections).

3.3 Conclusions

Confocal microscopy is a technique capable of generating sharp high resolution 3D images. When combined with fluorescent techniques, it provides a way to study cellular processes in living systems that would otherwise be impossible to study with conventional imaging techniques. During the course of this PhD we have worked with two sets of images.

The first set of images was provided by biologists of the Institut de Biologie de Valrose (Nice, France). They have used fluorophores to label single neurons from a specific region of the *Drosophila* fly brain. This particular region has been widely studied in the past. Then, a confocal microscope was used to acquired 3D images of them. Each image stack consists of two channels. The first channel presents the morphology of single axonal trees, while the second channel presents the morphology of the brain structure where the neurons develop. Both normal neurons and neurons in which certain genes were inactivated have been imaged. The mutated genes have been linked to neurological diseases in humans.

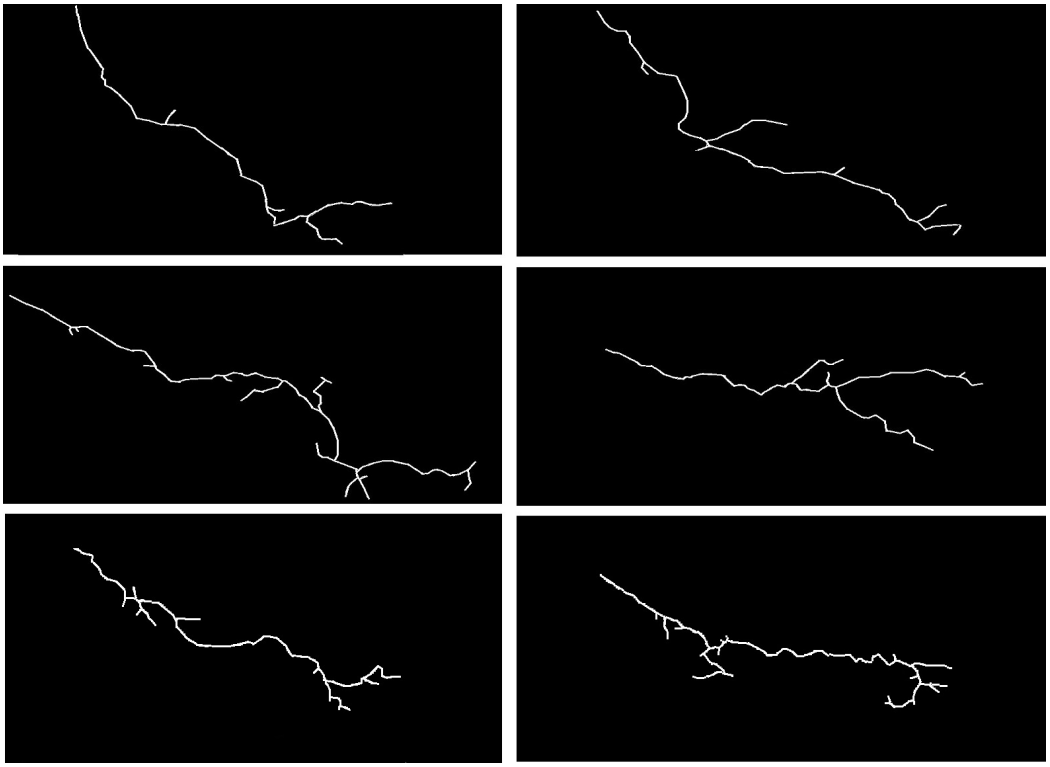


Figure 3.9: Some examples of normal (top), mutant type 1 (middle) and mutant type 2 (bottom) axonal trees that are similar to each other (2D projections).

We have also considered reconstructions taken from the database Neuromorpho, the world's largest open neuron reconstruction database.

Biologists are interested in understanding the morphological differences between the different populations of neurons. In the following chapters we will propose methods to analyze and quantify these differences.

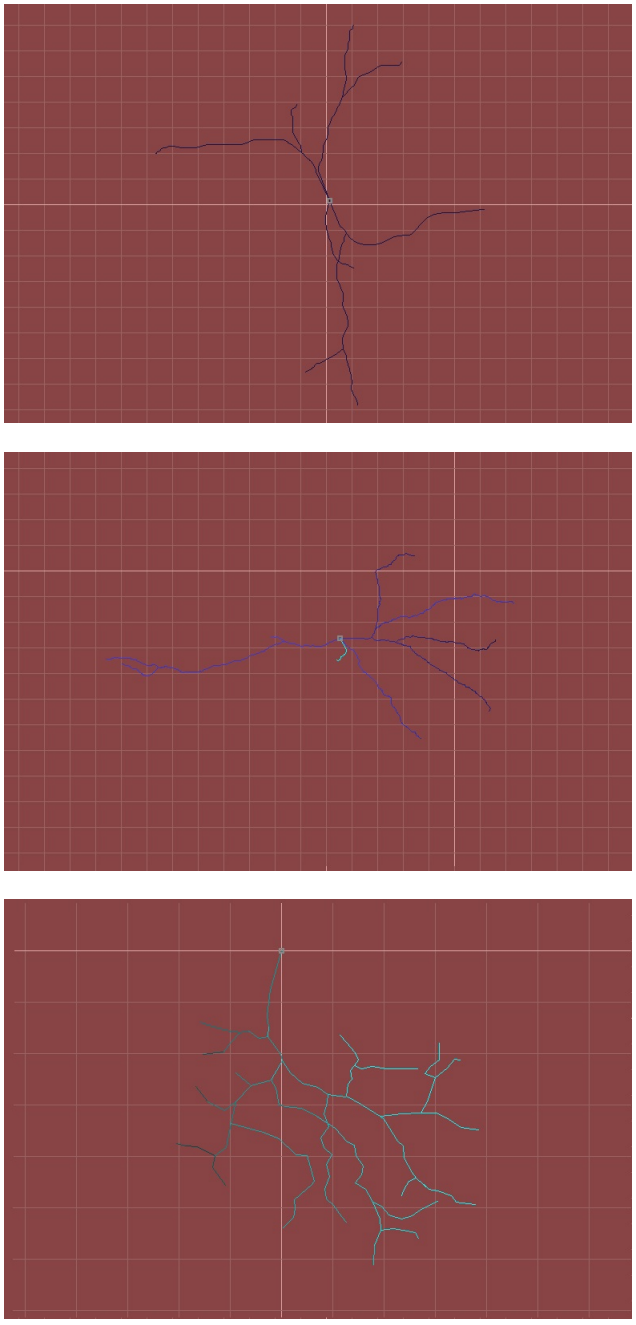


Figure 3.10: Examples of reconstructions of different types of neurons from Neuromorpho: Interneuron (top), Medium spiny cell (middle) and Retinotectal (bottom) (visualized with Neuromantic).

Chapter 4

Axon Morphology Analysis: from Image Processing to Modelling

Contents

This chapter provides an overview of the different image processing problems that we have worked on during this thesis: segmentation, morphology comparison and axonal growth modeling.

Segmentation is the first step necessary for the morphological analysis of axonal topologies. Given the nature of the neuronal images (large size, low signal to noise ratio, etc), this step presents many difficulties.

The tracings obtained in the segmentation stage can be compared to one another using a variety of methods. This allows us to identify morphological characteristics of different populations of neurons. In addition, the tracings can be used as input to simulation methods that estimate their parameters from them. These simulation methods can help us, among other things, to understand the biological processes involved in neuronal formation.

We describe different methods found in the literature for each of these three topics, and discuss their advantages and disadvantages.

4.1 Introduction

The properties of neuronal cells are controlled in part by their intrinsic gene expression program and in part by neuron-neuron interactions. It is manifested by a precise morphology as well as adapted distribution and number of synapses. In this work we concentrate on the most visible of these factors: neuronal morphology. Morphological properties can include general characteristics such as length and number of bifurcations, and more complex characteristics such as shape. Each of these features has an impact on the behavior of individual neurons and on their ability to inter connect and create neural networks.

Given a set of neuronal images, the first step necessary for the morphological analysis of the neurons is the segmentation of the images. Thus, we obtain a simpler representation of the neuronal topology that synthesizes the most relevant characteristics. Over the past decades, much progress has been done in this field. Many different algorithms and software packages can be found in the literature, and new methods appear each year. However, for the moment, we do not have techniques that can be applied to any type of neuronal images. In this chapter we briefly review some of the currently available techniques.

Once we have obtained the tracings, one needs to analyze their shape and to quantitatively compare their morphology. Several approaches have been proposed over the years. Different methods will be discussed in this chapter, with a special emphasis put on TED and Path2Path. These two methods will be used in Chapter 6 to evaluate the performance of our method.

Finally, a different yet related problem is the modeling and simulation of neuronal development. These techniques provide a way of investigating the underlying processes and natural parameters (i.e. axonal elasticity) involved in the neuronal

formation. The models can work on different scales (from sub to macro cellular) and consider different biological processes depending on their complexity and focus. Several solutions exist for the generation of realistic individual neuronal morphologies, as well as methods for the generation of full neuronal circuits. Three of such models will be briefly explained in this chapter.

4.2 Segmentation

The automated extraction of axons from biological images is a key problem in the field of neuron image analysis. This is put into evidence by the proposal of the DIADEM (DIgital reconstruction of Axonal and DEndritic Morphology) challenge in 2010 [K.M. Brown 2011]. DIADEM is an international competition whose purpose was to promote advances in the field of automatic neuron reconstruction. The challenge provided 6 image databases that provide a representative sample of the different challenges faced by segmentation algorithms. The images correspond to different animal species, brain regions, neuron types and image modalities. They provide a benchmark on which to evaluate tracing methods.

At present, the most common segmentation approach is manual tracing. The process of manually segmenting neuronal structures (even in the case when the user is aided by a computer application) can take, depending on the size of the images and complexity of the structures, up to several days. Moreover, given the recent advancements in the acquisition process, biologists nowadays posses large image databases. The manual processing of these images is thus unfeasible.

To tackle this challenge, many different methods for the automatic or semi-automatic segmentation of neuronal structures have been proposed. These algorithms provide different levels of automation, and are based on different techniques. Some are designed to work on 3D image stacks, while others are restricted to 2D projections.

In this section we will describe three algorithms that exemplify different classes of algorithms. We start with V3D-Neuron [Peng 2010], a semi automatic graph-based algorithm. We then describe the algorithm proposed by Zhang et al. [Zhang 2008]. This method exemplifies a family of algorithms specifically designed to work on images containing multiple non branching axons. Finally, we describe Tree2Tree [Basu 2010] and Wearne et al. [Wearne 2005]. Like many others, these two methods propose a multi-step approach that consists on the binarization of the image, the skeletonization of the binary objects, the gap filling of the skeleton segments and a final post processing step. We have also chosen this segmentation approach for the method proposed in Chapter 5.

In addition, several computer applications have been developed for the processing of such images. These applications, which include different extraction algorithms, provide manual or semi-automatic modes of operation along with friendly graphical user interfaces (GUI). Two of these applications will be discussed in this section.

A more comprehensive review of different segmentation methods and tools can be found in [Donohue 2011].

4.2.1 V3D-Neuron

Peng et al. [Peng 2010] developed a graph-augmented deformable model to reconstruct the 3D structure of neurons. The method requires the user to mark the root and end nodes. Then, the algorithm searches for the optimum path connecting the markers.

The first step of the algorithm is to create a graph $G = (V, E)$, where V is the set of all nodes and E the collection of edges between them. Each image voxel v corresponds to a node of the graph. The nodes associated to neighboring voxels are connected with undirected edges e, whose weights are defined as the product of the euclidean distance between them and a second term that depends on the image intensity at the points:

$$\begin{aligned} e(v_0, v_1) &= \|v_o - v_1\| \left(\frac{g_I(v_0) + g_I(v_1)}{2} \right) \\ g_I(v) &= \exp(\lambda(1 - I(v)/I_{\max})^2) \end{aligned} \tag{4.1}$$

where $I(v)$ corresponds to the intensity of the image at point v, $I_{\max}$ is the maximum intensity of the image and λ is a parameter of the algorithm.

The authors then use Dijkstra's algorithm to find the optimum shortest path P between each end node and the root node (all previously selected by the user). Given the way that the $g_I(v)$ was defined, the least cost path between a pair of nodes will the one that passes through the brightest voxels with the least overall distance.

The final step consists on using the optimum paths to initialize the control points of a deformable curve C, which is further refined and smoothed using a local search to better fit the optimum tracing. The deformation step is formalized as an energy minimization problem. The energy function for a discrete deformable curve C includes several terms that consider the length and smoothness of the curve and the intensity of the image on the points of the curve. Moreover, this energy formulation can easily incorporate additional prior knowledge such as a prior user

selected C curve.

At the end of the process, the final neuron tracing will be generated by connecting the nodes of each optimum path (one per end point).

This method presents promising results, but the need for manual selection of end points is an important shortcoming, specially when analyzing large databases of images.

4.2.2 Dynamic Programming - Watershed

In [Zhang 2008], the authors propose a semi-automatic extraction algorithm for microscopy images of axons. The authors combine Dynamic Programming (DP) searching (an optimization procedure designed to efficiently search for the global optimum of an energy function) with marker controlled watershed segmentation to solve the segmentation problem.

The process starts with the user manually selecting a set of seed points, one point for each axon in the first slice. The method then behaves as a tracking algorithm. For each seed point, the algorithm dynamically searches for optimum paths for the candidate point from its neighboring pixels. To improve the computational efficiency of the algorithm, only the neighbors within a small search region on next slice are considered. Then, a cost is assigned to each of the voxels in the search region. This cost is calculated using a function that depends on the derivatives of the image at the given points.

The points with the minimum cost value are used as markers for the watershed method, which is applied to segment the different axons in the current slice. The centroids of the segmented regions are used as the detected centerline points (the seeds of the following slice) and a new searching process is applied starting from these points. This process continues until the last slice is reached.

The method is based of certain assumptions, the most relevant for us being that all the axons start at the first slice and no new axons appear during the rest of the slices (no branching or merging of axons). As we saw in Chapter 3, our data is composed of images containing a single axonal trees that usually present several branches and sub branches. Thus, this method is not appropriate for the processing of our images.

4.2.3 Tree2Tree

Tree2Tree [Basu 2010] is a fully automatic (does not require the manual selection of a starting seed point) neuron segmentation algorithm for 2D images. It consists on several consecutive steps, and its output is the centerline of the tree with nodes

placed at a user defined resolution.

The first step is the binarization of the image, which is performed using a non global method to adapt to brightness changes in the image. The resulting binary components are then skeletonized using a classical algorithm.

These skeleton segments are usually disconnected. Thus, the authors propose a graph based gap filling method. First, a distance between neighboring segments is defined, which depends on the angle and euclidean distance between them. Then, a graph G is constructed, where each segment is a node and the weight of each edge is the distance between them. Two nodes are connected only if they are one of the k (a parameter of the method) nearest neighbors of each other. Finally, an optimum connectivity tree M is constructed from the global graph G such that the sum of the edges over the entire tree is minimum.

To obtain the final segmentation, the nodes of M are connected using cubic splines.

The fact that the method is restricted to 2D data limits its usefulness since a 2D analysis does not give crucial information about the axon morphometry and topology. In particular, the lengths and bifurcation points in the 2D projection might be different to the original 3D image.

4.2.4 NeuronStudio

This algorithm [Wearne 2005] is another example that follows the multi-step segmentation approach. Unlike Tree2Tree, this algorithm requires the user to select an initial seed point that is used in the binarization step to indicate to the algorithm which of the (potentially) found binary objects is the neuronal structure.

The binarization step is followed by a topology and geometry-preserving 3D skeletonization step [Rodriguez 2003]. Finally, the skeleton is processed to remove loops and eliminate unwanted spurious branches (prunning).

This algorithm is integrated in the NeuronStudio software package, a free program designed for the manual, and semi automatic reconstruction of neuronal structures from confocal and multi-photon images.

4.2.5 Neuromantic and FeatureJ

Neuromatic is a free application for the manual or semi-automatic reconstruction of neurons from single images or image stacks. It is being developed primarily by Darren Myatt at the University of Reading. The software has two modes of operation, manual and semi-automatic.

In the manual mode, nodes are added to the reconstruction one at a time by the user. Consecutive nodes are automatically connected by a straight line. Carrying on this procedure results in the entire neurite being segmented.

In the semi-automatic mode, the software assists the user in the segmentation procedure. The user needs to select an initial point, and then trace the axon with the mouse. The software will automatically try to detect the filament and add nodes to it (similar to the Magnetic Lasso tool of Adobe's Photoshop). The detection is based on a particle filter algorithm that tracks small segments of filament along space.

Because the software requires heavy user interaction (even in the semi-automatic mode), it would not be practical to use on a large database of images.

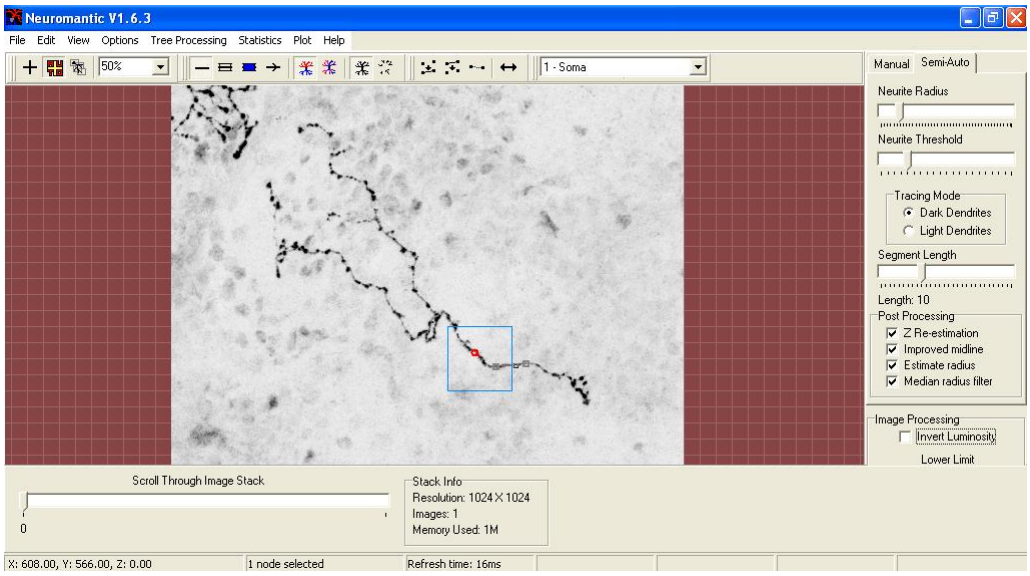


Figure 4.1: Screenshot of Neuromatic.

On the other hand, FeatureJ is a open source plugins of ImageJ [1] for the extraction of image features. It was designed by Erik Meijering at the Biomedical Imaging Group of EPFL, Switzerland.

The software supports gray-scale images of up to five-dimensions (5D), and includes several functions such as the computation of the Hessian, the derivative and the Laplacian. In particular, in [Grider 2006], the authors apply the Hessian function for the quantification of axons in microscopy images. This function enables one to compute, for all image elements (pixels/voxels), the eigenvalues of the Hessian, which can be used to discriminate locally between plate-like, line-like, and blob-like

[1] http://rsbweb.nih.gov/ij/

image structures.

After testing this application, we determined that it was not suited to our needs. It is not appropriate for high-throughput analysis of images since the user has to select manually which function to apply (Laplacian, Hessian, derivative, etc) and the values of the involved parameters. This has to be done for each individual image, and one image at a time. Moreover, when the Hessian plug-in is applied to our images (see 3), the results are hollow disconnected axons (and noise). These resulting images cannot be used for finding the branching points. Although this appears to contradict the results presented in [Grider 2006], this is due to the fact that the authors apply the software to a different type of images (where there are hundreds of thin axon segments).

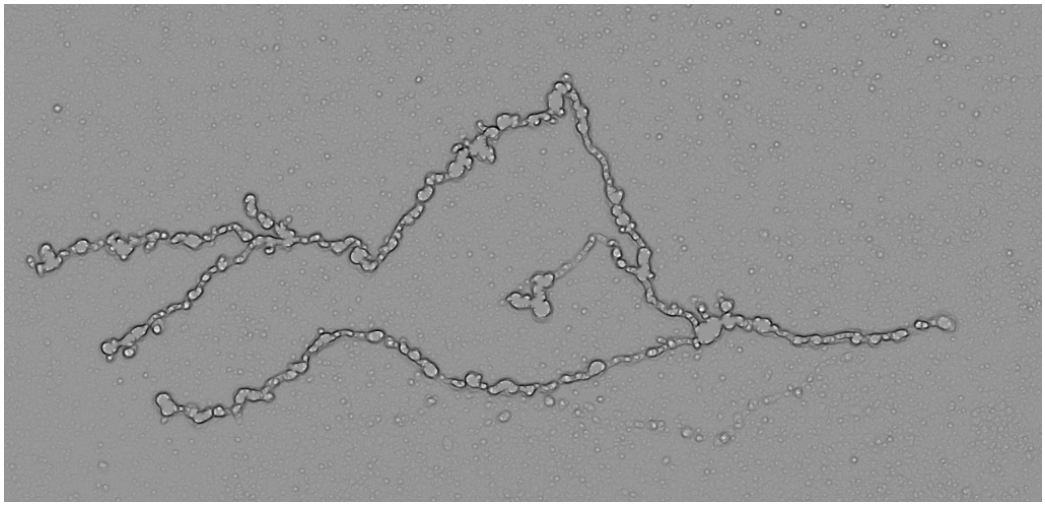


Figure 4.2: Result obtained by FeatureJ on one of our axonal images.

4.2.6 Conclusions

The segmentation of neuronal images is a very challenging problem. Given the diverse nature of the neuronal images (different imaging modalities, different structures, etc), multiple reconstruction strategies can be found in the literature. Each of these techniques is adapted to a particular type of images. In this section we have reviewed several techniques. Only two of them are applicable to our type of images (V3D and NeuronStudio), but their performance is not optimal. Therefore, in Chapter 5 we will propose a new segmentation algorithm better adapted to our problem. Both V3D and NeuronStudio will serve as benchmark for comparison with our method.

4.3 Classification

The comparison and classification of neuronal morpholigies is an important problem in neuroscience. Tracings computed with different segmentation algorithms can be compared to evaluate their performance. Additionally, by comparing the tracings of different populations of neuronal structures, biologists can gain an insight into the functional properties of different populations.

A variety of methods can be found in the literature, each one with its own advantages and disadvantages. In this section we will describe five approaches, and discuss their advantages and disadvantages.

4.3.1 TED

TED [Tai 1979, Selkow 1977, Zhang 1996] is a metric between labeled trees based on the tree-edit distance. Under this approach, the difference between two trees is measured by counting the number of nodes that must be added/removed/relabeled from one tree in order to transform it into the other. Adding node n to node m will make it its child node. Removing a node n will delete n and make its children become the children of the parent node. Finally, relabeling a node simply involves changing its label (for example, from n to m).

A cost $\gamma(o) \geq 0$ is assigned to each operation o. Moreover, we can define sequences of operations $S = \{o_i\}$. The total cost of a sequence S is the sum of the costs of each of its elements:

$$\gamma(S) = \sum_i \gamma(o_i) \tag{4.2}$$

Given two trees T_1 and T_2 with nodes $T_1 = \{t_1(1), ..., t_1(n)\}$ and $T_2 = \{t_2(1), ..., t_2(m)\}$, the distance between the two is defined as the minimum cost of all the operation sequences that transform one tree into the other:

$$D_{TED}(T_1, T_2) = \min_{S_i : S_i(T_1) = T_2} (\gamma(S_i)) \tag{4.3}$$

This metric has already been successfully applied to neuronal morphology analysis [Heumann 2009] as well as to the analysis of other branching structures.

One of the main disadvantages of this purely topological method is that double mistakes (an extra branch and a missing one) extending from the same node are ignored (see Figure 4.3). Moreover, the geometry of the curves between nodes is not taken into account.

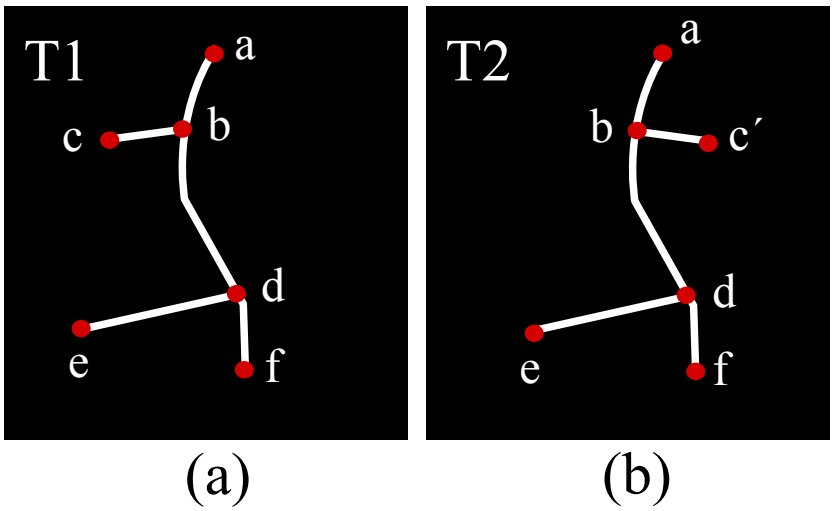


Figure 4.3: Double mistakes example for TED. Although both trees are different, TED will give a distance equal to zero.

4.3.2 Path2Path

Path2Path (P2P) was proposed in [Basu 2011] as a quantitative neuron comparison method which takes into account both topological and geometrical information. It is based on finding the minimum deformation energy between two trees, which depends on the L_1 norm between the curves that make up the trees and on two other functions that encode the topological information of the tree.

Let us consider a neuronal tree N with n terminals (end points). N can be decomposed into n paths that start in the root node (which is centered at the origin) and end in each terminal. Paths are constructed such that they do not have multiple crossings with each other. Thus, $N = \{f_i\}$, where each path is a continuous function $f_i : [0,1] \rightarrow \mathbb{R}^3$ with $f_i(0) = [0,0,0]$ (see Figure 4.4). The authors then define the path concurrence C_{f_i} of a path f_i as the piecewise constant integer function $C_{f_i} : [0,1] \rightarrow \mathbb{N}$. The path concurrence of a point in a path is defined as the number of paths that contain this point.

To take into consideration the number of levels a particular point is separated from the root, the authors introduce the concept of hierarchy. The hierarchy H_{f_i} of a point in a path f_i is defined as the number of times other concurrent paths leave f_i as one travels down from the origin to this point following path f_i. Formally, $H_{f_i} : [0,1] \rightarrow \mathbb{N}$ is a piecewise constant integer function.

These three functions are combined into a triplet (f_i, C_{f_i}, H_{f_i}) that assigns an unique structural signature to each path f_i in N. Moreover, the path deformation

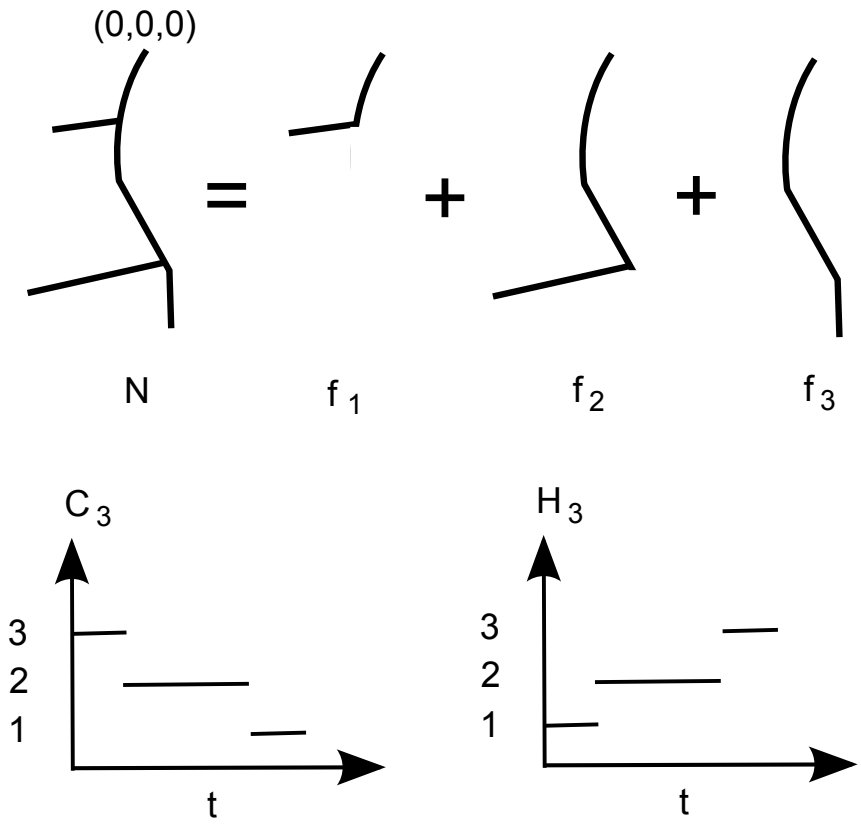


Figure 4.4: Neuron decomposition into paths for Path2Path and construction of the C and H functions for one of the paths.

energy P_{f_i,g_j} that transforms the triplet (f_i, C_{f_i}, H_{f_i}) into the triplet (g_j, C_{g_j}, H_{g_j}) is defined as:

$$P_{f_i,g_j} = \int_0^1 \frac{\mid C_{f_i}(t) - C_{g_j}(t) \mid\mid f_i(t) - g_j(t) \mid}{\lambda + \sqrt{H_{f_i}(t) H_{g_j}(t)}} dt \tag{4.4}$$

where λ is a parameter of the algorithm. Finally, given two neuronal trees $N = \{f_1, ..., f_n\}$ and $M = \{g_1, ..., g_m\}$ (with $m \geq n$), the distance between them is defines as:

$$D_{Path2Path}(N, M) = \min_{\sigma} \frac{1}{n} \sum_i P_{f_i,\sigma(f_i))} \tag{4.5}$$

where σ is a matching operator between both neurons that assigns each path of N to a path in M. It should be noted that the trees are aligned before the calculation of the distance by finding the optimum rotation (in the least squared sense) between them around the origin.

Path2Path combines topological and geometrical information but has a few

shortcomings. For example, the proximal part of the trees has a much larger influence on the distance than the distal part. Finally, the $L1$ norm is not easy to interpret, as opposed to metrics based on shape spaces, which embed stretching and bending forces necessary to transform one curve into another.

4.3.3 DIADEM Metric

The DIADEM metric was proposed as a way of quantifying the quality of the segmentation results submitted to the DIADEM Challenge. This metric was specifically designed for comparing different reconstruction of the same neuron by different methods or of the same neuron over time.

Let us consider two tracings of the same axon, one referred to as the gold standard and the other as the test tracing. Each bifurcation and end point is referred to as node. The scoring process begins by taking each node in the gold standard tracing (starting at the root) and matching it against all the nodes within a given distance threshold in the test tracing.

Each test match to the target gold standard node is then checked for proper connectivity based on the path from the target node to an ancestor (a node on the direct path to the root) that matches an ancestor of the target node. The E coefficient is used to determine whether the two are good matches or not:

$$E = \frac{L_G - L_T}{L_G} \tag{4.6}$$

where L_G, L_T are the lengths of the gold standard and test paths.

Terminal nodes (end points) without a match are marked as misses. If a node match is not found for a target bifurcation, a procedure is used to look for potential continuation matches after all gold standard nodes have already been checked for direct matches.

Once all the nodes in the gold standard have been processed and all the excess nodes have been determined, the total score (distance) between the two tracings is calculated.

The algorithm provides several parameters to modify the behavior of the metric depending on the needs of the user. For example, nodes can be weighted uniformly or by their degree in the tree (second and third level nodes are less important that main nodes).

4.3.4 Feragen et al.

In [Feragen 2013, Feragen 2010a, Feragen 2010b] the authors constructed a shape space for rooted tree like shapes and studied different metrics on this space. Under this framework, geodesics are defined between trees and used to compute distances. The resulting distances takes into account both topological and geometrical information of the tree.

Let us consider a tree-like shape in $\mathbb{R}^2$ or $\mathbb{R}^3$. The authors start by considering the simpler $2D$ case where trees present a natural edge ordering and then extend their method to the more general $3D$ setting. $2D$ trees are referred to as ordered trees.

In this paper, trees are represented as a pair $(\mathbb{F}, x)$, where $\mathbb{F} = (V, E, r)$ is a rooted planar binary tree that encodes the shape topology and $x \in \prod_{e \in E} A$ are the attributes that are used to describe the edge shape. A represents the attribute space defined as $(\mathbb{R}^d)^n$, where d is the dimension of the curve representing the edge and n the number of points per edge (see Figure 4.5). In order to simplify the comparison of trees with different topologies, all tree like shapes are represented by the same maximal binary tree. This maximal tree is a binary tree large enough to represent the topologies of all the trees in a database. Non existing (collapsed) edges are represented by a sequence of origin points in x. Thus, given a database of trees and the corresponding maximal ordered binary tree $\mathbb{F}$, any attributed tree T can be represented by a point $x = (x_e)_{e \in E}$ in $X = \prod_{e \in E} (\mathbb{R}^d)^n$, where x_e describes the shape of edge x. X is referred to as the pre-shape space since some trees can be represented by multiple points in X (we will encounter a similar problem in Chapter 6).

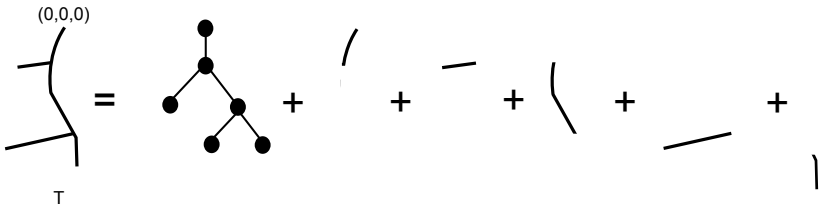


Figure 4.5: Tree decomposition.

This presents certain problems, which are solved by creating a quotient space $\bar{X} = (X, \sim) = \{\bar{x} | x \in X\}$ of equivalent classes $\bar{x}$. This new space is referred to as the shape space of ordered trees.

Given a metric d on the pre-shape space X, the authors define a metric on $\bar{X}$

as:

$$\bar{d}(\bar{x},\bar{y}) = \inf\left\{\sum_{i=1}^{k} d(x_i,y_i) | x_1 \in \bar{x}, y_i \sim x_{i+1}, y_k \in \bar{y}\right\} \tag{4.7}$$

Two metrics on X are considered:

$$\begin{aligned} \|x-y\|_1 &= \sum_{e\in E} \|x_e - y_e\| \\ \|x-y\|_2 &= \sqrt{\sum_{e\in E} \|x_e - y_e\|^2} \end{aligned} \tag{4.8}$$

Both generate metrics $\bar{d_{1,2}}$ on $\bar{X}$. This implies that given two trees, there is always a geodesic path between them.

To extend this result to the $3D$ case, a new quotient space is defined. Unlike in $2D$, trees in $\mathbb{R}^3$ have no standard edge order. Thus, one needs to consider all the possible orderings of the same tree. This presents some practical computational problems. The new quotient space $\bar{\bar{X}} = \bar{X}/G$ is defined, where G is the group of all possible reorderings of the trees. Like for $\bar{X}$, each of the two considered metrics on X induce metric $\bar{\bar{d}}_{1,2}$ on $\bar{\bar{X}}$. It can be shown that $\bar{d}_1, \bar{\bar{d}}_1$ corresponds to the TED metric (see 4.3.1) for ordered and unordered trees respectively. On the other hand, metrics $\bar{d}_2, \bar{\bar{d}}_2$ are referred to as QED metric on ordered or unordered trees.

Both metrics present certain advantages and disadvantages. In particular, the use of QED simplifies the computation of mean shapes in the shape space. However, it is computationally more expensive than TED.

The authors have applied their method to the study of airway trees from pulmonary CT scans and leaf morphology. This method presents good results and a very strong mathematical foundation but for the moment its application is restricted to small poorly branched trees due to its computational complexity.

4.3.5 Feature Based Classification

There are countless measurable morphological features that can be used to classify populations of neurons. However, considering too many features is counterproductive since it will increase the computational cost of the classification. In addition, redundant or non significant features will decrease the influence of more relevant ones. Therefore, choosing a group of key features is not a trivial matter.

Most authors consider that both the length and number of branches of an axon are important features since they are closely related to the spatial extent of the neuron and thus to the number of potential partners they can have. Moreover, the length of the paths in an axon is also functionally relevant since the longer the path,

the greater the signal attenuation throughout the axonal tree [Brown 2008].

The overall tree shape is also an important feature to consider when classifying axons. For example, the morphological signatures of pyramidal and moto neurons are quite different. This is due to the fact that they have different functions in the nervous system.

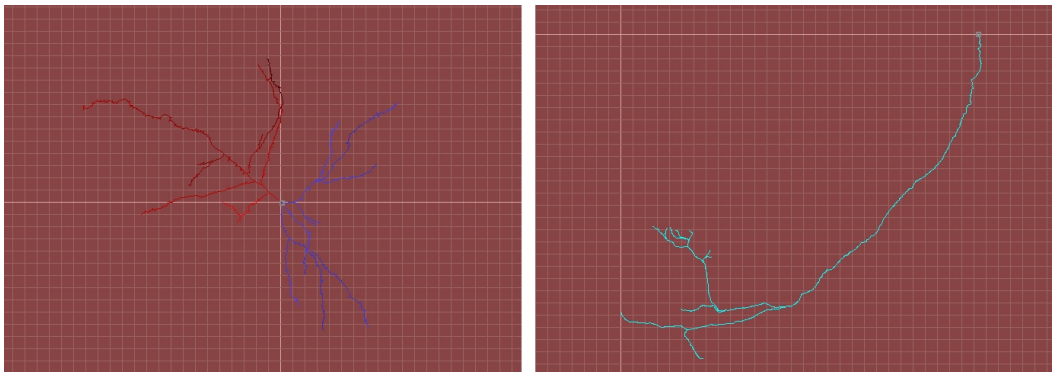


Figure 4.6: Pyramidal cell from the Neocortex of Human (left) and Motoneuron from the Peripheral nervous system of Mouse (right) (taken from neuromorpho.org, Allman and Lu-Lichtman collections).

In [Brown 2008] the authors consider twenty features, such as soma surface, number and length of branches and bifurcation angle. These same features were also chosen in [Rong 2011] to implement a neuronal classification and nomenclature method. Overall, these features are able to reasonably characterize the main morphological characteristics of neurons.

To improve the performance of the classification, authors such as [Rong 2011] propose to use PCA to obtain the principal components, thus finding which of the features are more relevant.

4.3.6 Conclusions

In this section we have presented different alternatives to the neuronal morphology comparison problem.

Some authors have developed distance functions that capture key morphological properties. These distances can take into account only topological information, or also consider the geometry of the curves between the nodes that make up the trees. An alternative approach consists in directly analyzing a set of key features.

Although the reviewed methods offer interesting approaches and good results, we have decided to propose a new approach specifically adapted to our images. This method will be presented in 6, and its performance compared against TED,

Path2Path and the Feature based approach. It will be shown that our method outperforms the other considered methods. Moreover, we propose a method for the computation of the mean shape of a population of trees, which can be used to summarize the morphological characteristics of the considered population.

4.4 Modeling

Mathematical modeling aims to make a given process easier to understand, define and simulate. It is based on identifying key aspects of the process, which are then simplified and formalized in a mathematical way. The use of models can help explain the system underlying processes and make predictions about its functioning. Models usually depend on several parameters that define their behavior. The estimation of these parameters is an important (and usually complicated) aspect of modeling.

Many methods have been proposed in the field of neuronal development modeling and simulation. These methods differ with respect to the processes being modeled, the methodology and type of data used for the parameter estimation and the dimension considered (2D, 3D or 4D). In this section we present three of such models.

4.4.1 Segev et al.

In [Segev 2000] the authors introduce a 2D off-lattice model of self wiring neural networks inspired from previous work on the study of patterning of bacterial colonies. Neurons are represented as being composed of the soma (represented by a fixed point on the lattice) and of growing growth cones that form the neurites. Somas and neurites communicate with each other by means of attractive or repulsive signals that determine the networks morphology. The signal concentrations are determined by reaction diffusion equations that depend on several parameters.

In the absence of chemical signals, the movement of the growth cones is modeled with a non-uniformed random walk. Given an initial growth direction d_0, the following direction is chosen from a non uniform probability distribution function P_0:

$$d_i = d_{i-1} + P_0(\theta) \tag{4.9}$$

where θ is the angle between vectors d_i, d_{i-1}. The highest probability is assigned to continue moving forward in the same direction, and the second most probable is to move backwards. In total, the authors consider 12 possible directions, which are measured relative to the last direction of movement.

In the presence of chemical signals, the distribution function P_0 is modified.

The probability changes linearly with the concentration gradient, and depends on the derivative of the concentration and on several other parameters such as the cones sensitivity to the two signals. The sensitivity of the cones to the chemical signals changes as a function of the neurite length. When the neurite first sprouts from the soma, it is mainly affected by a repulsive signal emitted from its mother soma. Once the neurites reaches a specific distance (a parameter of the model), its sensitivity to this signal greatly decreases, and its sensitivity to the attractive signals emitted from the other somas increases. As a result, the neurites grow toward a neighboring cell, thus establishing neural networks.

To quantitatively compare the model behavior to experimental data, the authors compare the histogram of the number of connections vs. lengths between connected neurons.

This model presents an interesting approach. In particular, it models the chemical signals present in the environment. However, the model parameters are not estimated from the data, and it is restricted to the 2D case. This reduces the utility of the model.

4.4.2 Kobayashi et al.

[Kobayashi 2010] presents a 3D stochastic model of neuronal growth cones (tips of the growing axon) that takes into account the concentration of guidance molecules to determine their behavior. The model considers that each cone has k filopodia (see Chapter 2) separated by an angle θ. Each folipodium is represented by a unit vector of a given direction. These filopodium act like sensors that can detect the presence of guidance molecules in the extra cellular medium. This molecules are represented by concentration fields that are imputed by the user before the growth process starts. The guidance molecules are assumed to be released from a given point and to diffuse into the environment. The concentration follows a normal distribution (with parameters imputed by the user).

The algorithm specifying the behavior of each cone is divided into two parts. First, an initial growth direction is determined. This is achieved by considering a random walk that follows a normal distribution. Then, for each time interval t_i, the position of the next point is determined.

Each of the k filopodium detects the concentration of the field at the current point and creates a vector pointing in the direction of the sensors. The initial growth vector is then modified by adding the concentration vectors at the cone. Since this process can sometimes result in an unrealistic large turning of the direction, a regularization procedure is implemented.

The model does not consider bifurcations. The behavior of each growth cone is determined independently, and all cones are initially placed in the same starting point (usually the origin in the x-y plane).

Although the model does not present many parameters, none of them are estimated. Instead, the parameters are manually tuned to simulate different scenarios and the results are compared to 2D plus time images of growing axons. This is an important shortcoming of the method.

4.4.3 Koene et al.

In [Koene 2009] Koene et al. propose a very complete stochastic model for the generation of 4D (3D plus time) neuronal networks that includes both axons and dendrites. For each neurite growth cone, the model considers the processes of elongation, turning and branching. The authors estimate some of the parameters from real 4D images or growing neurons, and others are set manually based on properties of real neurons.

The model considers no discretization in space. However, the authors consider a time discretization. The growth process starts with a growth cone at the root. Then, for each time interval Δ_t, this cone can elongate, change direction and bifurcate. Each process is determined by stochastic models, and they are independent from each other.

The model proposes two ways of assigning the elongation rate (distance travelled by a growth cone in a time interval Δ_t). By default, the model assigns elongation rates by randomly sampling a Gaussian distribution. On the other hand, authors also give the option to calculate the elongation rate by taking into account the number of growth cones (branches) currently present in the neural network. This is done to model the competition for resources between the cones. Therefore, the velocity of the growth cone is calculated with the following equation:

$$v(t) = v_0 n(t)^{-F} \tag{4.10}$$

where F is a parameter that controls the strength of the competition between the cones.

For each time t_i, a growth cone can either continue to elongate in the same direction or turn. The probability to change direction in time t_i is given by the turn probability $P_d(t_i)$, which depends on a parameter of the model (turns per micrometer). For each instant t_i the turn probability is calculated and compared with a random number taken from a uniform distribution. If the random number is smaller than P_d, there is a change in direction. The new direction of growth

is calculated by doing the weighted average of the previous directions and finally adding a random deviation to the resulting vector:

$$d = \sum_k W_k u_k + r \tag{4.11}$$

where W_k are the weights (more weight is assigned to closer segments), u_k are the previous directions and r is a random rotation drawn from a uniform distribution.

The bifurcation of a growth cone is controlled by a time dependent branching probability that depends on three terms: a baseline branching rate (approximated by an exponential function), a term that makes the probability depend on the number of existing branches and a final term making the probability depend on the position of the cone (with respect to the root). Each term depends on various parameters that need to be estimated or imputed by the user.

This model is very complex, and the results provided by the authors show that it is able to produce realistic morphologies. However, the model disregards the chemical attraction field present in the cellular medium. Moreover, given that the model is designed to work on 4D, one needs to use bi-photon time sequences of developing neurons to estimate the involved parameters. The automatic segmentation of this type of images (which is needed for the parameters estimation) is a very complex subject.

4.4.4 Conclusions

Segmented neurons can serve as an input into neuronal development models. These models can provide an insight into the underlying processes and natural parameters involved in the neuronal growth. The different models proposed in the literature vary in complexity and scale.

Two of the reviewed models consider the chemical attraction field that affects the growth cone behavior and that we consider is an important process to model. However, none of the reviewed methods estimate this external attraction field form the data.

In Chapter 7 we will propose a new model for the simulation of axonal biogenesis that considers an external attraction field, and all the involved parameters (including those defining the attraction field) are estimated from 3D confocal microscopy images.

4.5 Conclusions

The functioning of the nervous system is closely related to the morphological properties of the neurons that constitute neuronal circuits. Key aspects of neuronal morphology alter the way neurons connect and interact with each other. To analyze these morphological characteristics, researchers usually depend on neuronal images (such as the ones described in Chapter 3) that need to be processed.

The typical analysis starts with the segmentation of the images in order to obtain a compact representation of the morphology. This is followed by a morphological comparison step. Reconstructions can further be used as a basis on which to build bio-models capable of providing morphological realism.

For the different addressed topics, we have summarized some selected works in the state of the art that represent the main approaches presented in the literature. We have determined that there exist a wide variety of methods, each one adapted to a different specific problem. However, none were well suited for our problem. Therefore, in the following chapters we will propose new methods for each of the three considered steps: segmentation, morphology comparison and axonal biogenesis modeling. These methods will be compared against other state of the art algorithms presented in this chapter.

Chapter 5

Axon Extraction

Contents

In this chapter we address the problem of axonal tree segmentation from confocal microscopy images. A multi-step segmentation method is proposed, which combines algorithms for filament enhancement, binarization, skeletonization and gap filling in a pipeline capable of extracting axons from images containing a single tree. The performance of the proposed method was evaluated on real images. Results support the potential use of this technique in helping biologists perform automatic extraction of axons from fluorescent confocal microscopy images.

5.1 Introduction

As we have seen before, in order to perform morphological analyses, biologist rely on different imaging techniques capable of capturing cellular and subcellular structures. One of the most important image modalities is confocal microscopy, an imaging technique capable of generating 3D image stacks of thick specimens at very high resolutions.

Biologists have labeled single neurons within intact adult *Drosophila* fly brains and used a confocal microscope to acquired 3D images of their axonal trees. Both normal neurons and neurons in which certain genes related to neurological diseases in humans were inactivated have been imaged (see Figure 5.1).

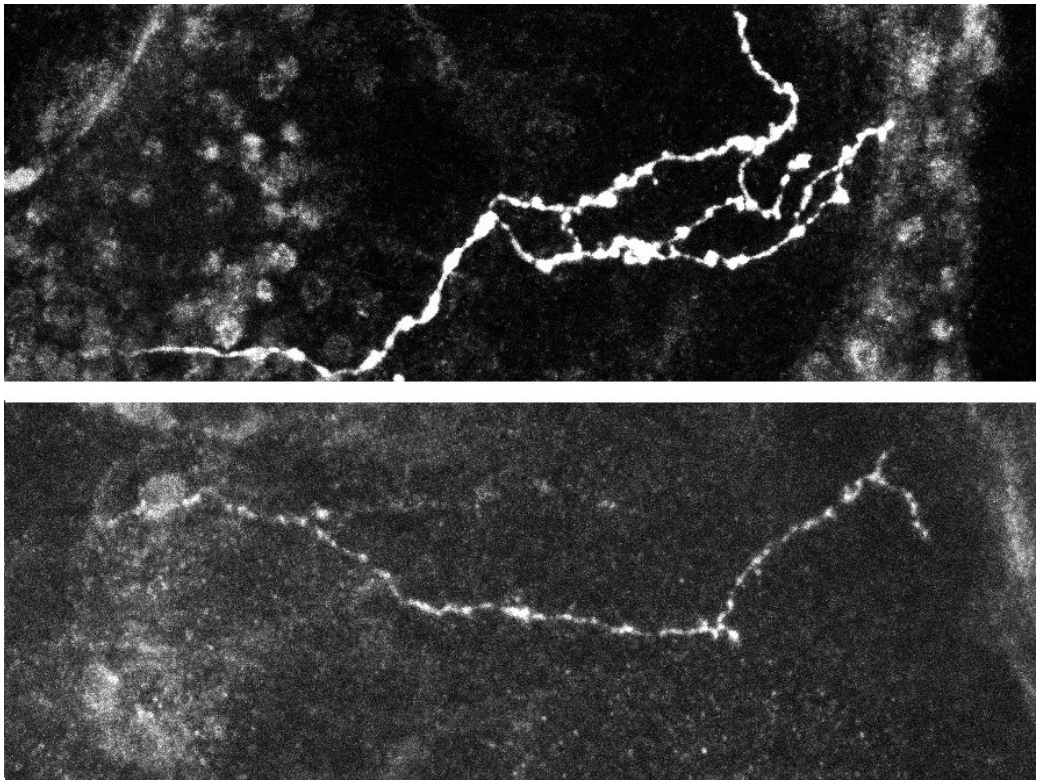


Figure 5.1: Normal (top) and mutant (bottom) axons (maximum intensity projections).

In order to perform the morphological analysis of the axons, it is first necessary to segment the images. Due to the high volume of image data (on average, each image has a size of 1024×1024 pixels by 16 layers) and the tortuous nature of the axons, manual processing is very time consuming. It is thus necessary to develop techniques for the automatic extraction and analysis of these neuronal structures.

As we have seen in Chapter 4, there are currently many methods for axon extraction in the literature. Some, such as [Zhang 2008, Wang 2007] concentrate on, given a 3D image stack containing many axons (sometimes overlapping each other), segmenting each individual axon. In most cases these images contain many non-bifurcating axons, and their objective is to segment each of them. Since the axons overlap on occasions, the authors have to resort to strategies that lead to overcomplicated and computationally heavy algorithms. Moreover, many of these methods require the user to manually select the center of each axon in the first slice of the stack.

A different family of methods (i.e, [Peng 2010]) requires the user to mark the root and end nodes. Then, the algorithm searches for the optimum path connecting the markers. This is not suited for automatically analyzing large databases of images,

which is necessary if one intends to statistically compare different populations of axons.

Other methods such as [Basu 2010] are limited to 2D images. This restricts their usefulness since a 2D analysis omits crucial information on the axon's morphometry and topology. In particular, the lengths and bifurcation points in the 2D projection might be different from the original 3D image. In addition, several computer applications (such as Neuromantic [1] and FeatureJ [2]) have been developed for the processing of this particular type of images (or similar ones). These applications, which include different extraction algorithms, provide manual or semi-automatic modes of operation along with friendly graphical user interfaces (GUI). These, however, require time consuming manual (or semi-automatic) tracing of the axon, obtain unacceptable results in our images and/or are restricted to the 2D case.

In this chapter we propose a solution that combines algorithms for filament enhancement, binarization, skeletonization and gap filling in a pipeline capable of extracting axons from confocal microscopy images containing a single labelled neuron (we do not consider the more challenging case of multiple labelled neurons that intertwine and synapse). Furthermore, this method is designed to work on 3D image stacks.

5.2 Axon extraction algorithm

Our method applies several consecutive steps for extracting the axons: Filament Enhancement, Binarization, Skeletonization and Gap Filling (see Figure 5.2).

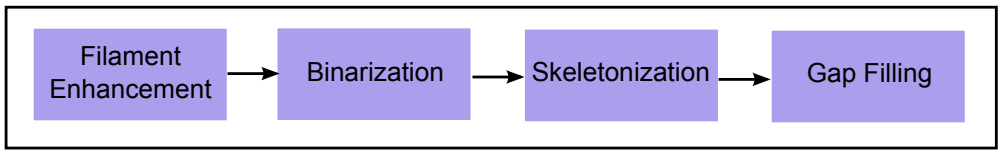


Figure 5.2: Block diagram of the segmentation process.

5.2.1 Filament Enhancement

The first step of the pipeline is filament enhancement. This is done with a multiscale filtering of the images based on Gabor functions. Due to the low resolution of our images in the axis perpendicular to the sections, we chose to perform this step of the process in 2D work on each slice independently. However, the analysis that follows can easily be extended to the 3D case.

[1] http://www.reading.ac.uk/neuromantic/
[2] http://www.imagescience.org/meijering/software/featurej/

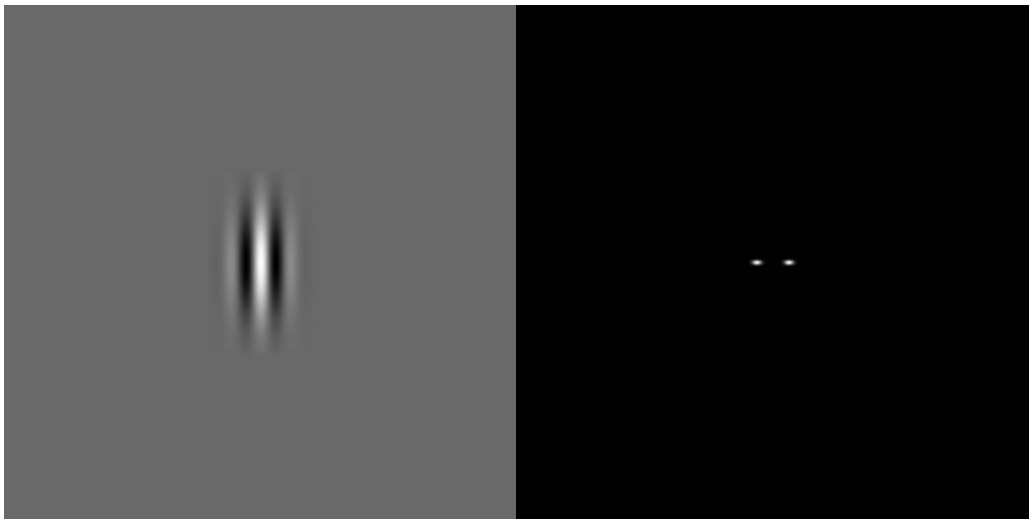


Figure 5.3: 2D Gabor function for a given set of parameters, space (left) and frecuency (right) domains.

Classical Gabor functions are defined as follows (see Figure 5.3):

$$G(x,y) = \cos\left(\frac{x_1\pi}{\sqrt{2}\lambda\sigma_g}\right)\exp\left(-\frac{x_1^2+\gamma^2y_1^2}{2\sigma_g^2}\right) \tag{5.1}$$

where $x_1 = x\cos(\theta) + y\sin(\theta)$ and $y_1 = x\sin(\theta) - y\cos(\theta)$. The parameters $\theta, \lambda, \sigma_g$ and γ represent the orientation, wavelength, scale and aspect ratio respectively. These functions have been widely used to characterize textures before [Grigorescu 2002], and by adapting the scale parameter σ_g can also provide an efficient way of characterizing local structures. If we consider the image $I(x,y)$, then the filtered image H is defined as $H = \|G * I\|$. A high value of H at a point (x,y) for a given θ is evidence of local structure (axon) for that direction and position.

Based on the previous G filter, we propose some modifications to improve the results. First, to avoid false positives on big edges in the image, we consider two Gabor filters each associated to a different side of the filament (left and right) (see Figure 5.4):

$$G^+(x,y) = \begin{cases} \cos\left(\frac{x_1\pi}{\sqrt{2}\lambda\sigma_g}\right)\exp\left(-\frac{x_1^2+\gamma^2y_1^2}{2\sigma_g^2}\right) & \text{if } x_1 \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

$$G^-(x,y) = \begin{cases} \cos\left(\frac{x_1\pi}{\sqrt{2}\lambda\sigma_g}\right)\exp\left(-\frac{x_1^2+\gamma^2y_1^2}{2\sigma_g^2}\right) & \text{if } x_1 < 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

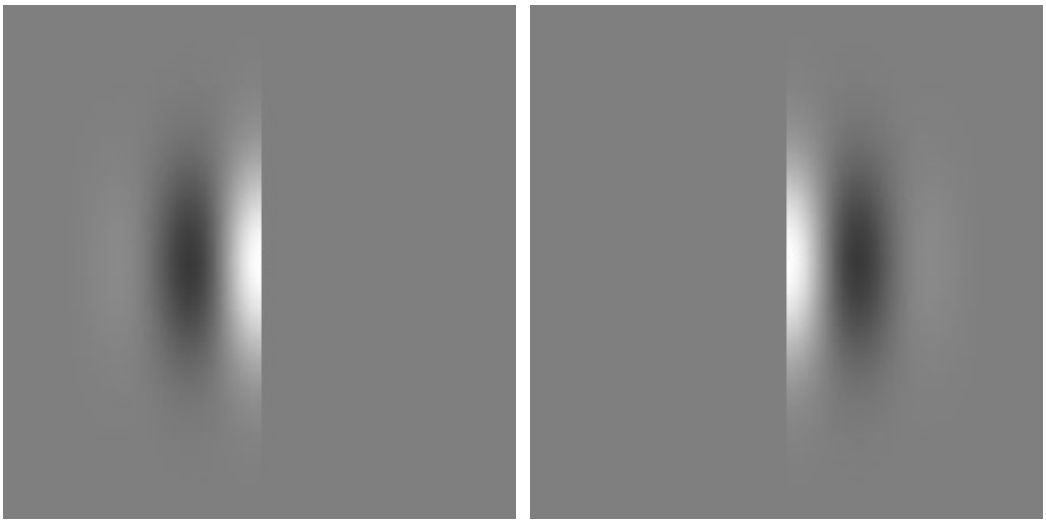


Figure 5.4: G^- (left) and G^+ (right) filters.

Finally, the filament enhancement filter is given by:

$$F = \min(\|\Psi^- * I\|, \|\Psi^+ * I\|) \tag{5.4}$$

where:

$$\Psi^-(x,y) = \begin{cases} \eta_-^- G^-(x,y) & \text{if } G^-(x,y) < 0 \\ \eta_-^+ G^-(x,y) & \text{if } G^-(x,y) \geq 0 \end{cases} \tag{5.5}$$

$$\Psi^+(x,y) = \begin{cases} \eta_+^- G^+(x,y) & \text{if } G^+(x,y) < 0 \\ \eta_+^+ G^+(x,y) & \text{if } G^+(x,y) \geq 0 \end{cases} \tag{5.6}$$

The constants $\eta_-^-, \eta_-^+, \eta_+^-$ and η_+^+ are obtained by the normalization of the two filters to eliminate their dependency to the absolute intensity of the image:

$$\begin{cases} \int_{\mathbb{R}^2} \Psi^{-,+}(x,y)\mathrm{d}x\mathrm{d}y = 0 \\ \int_{\mathbb{R}^2} \max(0, \Psi^{-,+}(x,y))\mathrm{d}x\mathrm{d}y = 1 \end{cases} \tag{5.7}$$

The remaining parameters involved in the filters are set by imposing certain conditions. First, we impose that the filter value is equal to zero at the edge of the filament, which means that we have:

$$\frac{\pi n}{2\sqrt{2}\sigma_g \lambda} = \frac{\pi}{2} \tag{5.8}$$

where n is the width (in pixels) of the filaments we want to detect. Then, given that the function $\exp(-x^2)$ is decreasing and starts to be close to zero for $x = 2$, we impose:

$$\frac{\pi x}{\sqrt{2}\sigma_g \lambda} = \frac{3\pi}{2} \tag{5.9}$$

Finally, we impose a ratio 3 between the length and the width of the filter, which gives:

$$\frac{2\gamma}{\sqrt{2}\sigma_g} = 2 \tag{5.10}$$

Thus, we obtain:

$$\begin{cases} \lambda = \frac{4}{3} \\ \sigma_g = \frac{3n}{4\sqrt{2}} \\ \gamma = \frac{1}{3} \end{cases} \tag{5.11}$$

To perform a multiscale analysis of the images, the parameter n is varied between the expected widths of the axons. Given the size of the filaments in our images, we have determined that a good set of values is to vary n from 4 to 10. We also run the filter for different θ to test for filaments in all directions. In practice, we do this by discretizing the $[0, \pi)$ interval into 16 equal intervals.

We have compared our approach with two other filament enhancement filters found in the literature [Frangi 1998, Wilkinson 2001], which are reviewed briefly in the following section.

5.2.1.1 Frangi et al.

The method is based on the fact that the main direction of a filament can be found by using the eigenvectors of the second order derivative matrix (Hessian matrix) computed from the values of the image at the point.

A first approximation of the derivative of an image can be computed by simply subtracting adjacent voxels. This estimator, however, presents a large sensitivity to noise. A more robust approach consists in convolving the image with a Gaussian kernel, thus including more points in the calculation of the derivative. Due to properties of the convolution operation, it can be shown that:

$$\begin{cases} I^G(x,y,z,\sigma) = G(x,y,z,\sigma) * I(x,y,z) \\ I^G_{ij}(x,y,z,\sigma) = \frac{\partial^2 G(x,y,z,\sigma)}{\partial i \partial j} * I(x,y,z) \qquad i,j = xx, xy, xz, yy, yz, zz \end{cases} \tag{5.12}$$

where $G(x, y, z, \sigma)$ is the Gaussian function. The second order derivatives can

later be used to construct the Hessian matrix:

$$H = \begin{bmatrix} I^G_{xx} & I^G_{xy} & I^G_{xz} \\ I^G_{yx} & I^G_{yy} & I^G_{yz} \\ I^G_{zx} & I^G_{zy} & I^G_{zz} \end{bmatrix} \tag{5.13}$$

Since the matrix is symmetric, the eigenvectors are orthogonal and the eigenvalues are real. Moreover, it can be shown that a spherical neighborhood centered at a given point of the image is mapped by H into an ellipsoid whose axes are along the direction of the eigenvectors (of H) and the corresponding axis semi lengths are the magnitudes of the respective eigenvalues. Thus, this ellipsoid describes the second order structure of the image at a local level. The three eigenvalues will be refereed to as λ_i, with $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$.

Given that the intensity variation along the direction of a filament is minimum, the eigenvector corresponding to the smallest absolute eigenvalue points in its direction. The remaining two eigenvectors form a base for the orthogonal plane. Therefore, a point belonging to a filament will present a small λ_1 and λ_2, λ_3 of large magnitude and equal sign (the sign indicating whether the filament is bright or dark). Consequently, one is interested in voxels whose eigenvalues fulfill the following conditions:

$$\begin{cases} |\lambda_1| \approx 0 \\ |\lambda_1| \ll |\lambda_2| \\ \lambda_2 \approx \lambda_3 \\ \lambda_2, \lambda_3 < 0 \quad \text{in the presence of light filaments} \end{cases} \tag{5.14}$$

Then, the authors define three features that depend on the eigenvalues of the ellipsoids. These features will later be combined into a filament function which measures the likeliness of a voxel to belong to a filament.

5.2.1.2 Wilkinson et al.

This article proposes morphological filters composed of connected-sets for the extraction of filamentous structures that is shape preserving. Two different strategies are proposed and compared: multi-scale filtering and single-step shape filtering using connected set thinning. In particular, the second method obtains comparable results but is more computationally efficient.

Single-step shape filtering can be achieved by thresholding an image using each possible grey level. Then, for each resulting binary image, all the connected com-

ponents that do not correspond to the searched shape are removed. Identifying a shape criterion capable of distinguishing filament-like objects from all others is the key problem under this methodology. Finally, all the resulting binary images are added again to obtain the final filtered result,

The authors propose a shape criterion based on the volume V and the moment of inertia I of a connected component C. It is known that the moment of inertia of a volume is minimal for a sphere and increases for elongated objects. Given a connected component C, its moment can be calculated using the following formula:

$$I(C) = \frac{V(C)}{4} + \sum_{x \in C} (x - \hat{x})2 \tag{5.15}$$

where $V(C)$ is the volume of C and $\hat{x}$ its center.

For a given component C, $I(C)$ scales with the size to the fifth power, while the the volume scales with the third power. Therefore the ratio $S = \frac{I}{V^{\frac{5}{3}}}$ is a purely shape dependent scaling invariant quantity. S presents its minimum for a sphere (0.23) and increases rapidly with elongation.

The authors propose to use S as the shape criterion in the shape filtering scheme, and eliminate every component C with $S > t$. In this equation, t is a parameter of the algorithm which should be tuned for different applications.

5.2.1.3 Filament Enhancement Filter Comparison

Figure 5.5 shows the results obtained for the different filament enhancement filters on an image. Although it is difficult to evaluate quantitatively, it can be appreciated that the proposed technique based on Gabor filters obtains better results for this type of images.

5.2.2 Binarization and Skeletonization

The next step in the extraction process is the binarization of the filtered images. The threshold is automatically computed by decreasing Otsu's threshold [Otsu 1979] by a certain percentage (found empirically for our images). Otsu thresholding is based on finding the threshold that minimizes the weighted within-class variance. This is equivalent to maximizing the between-class variance. The method makes several assumptions, the most important being that the image presents a bimodal histogram (see Figure 5.6).

Afterwards, the binary image is skeletonized. The skeleton of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton is usually

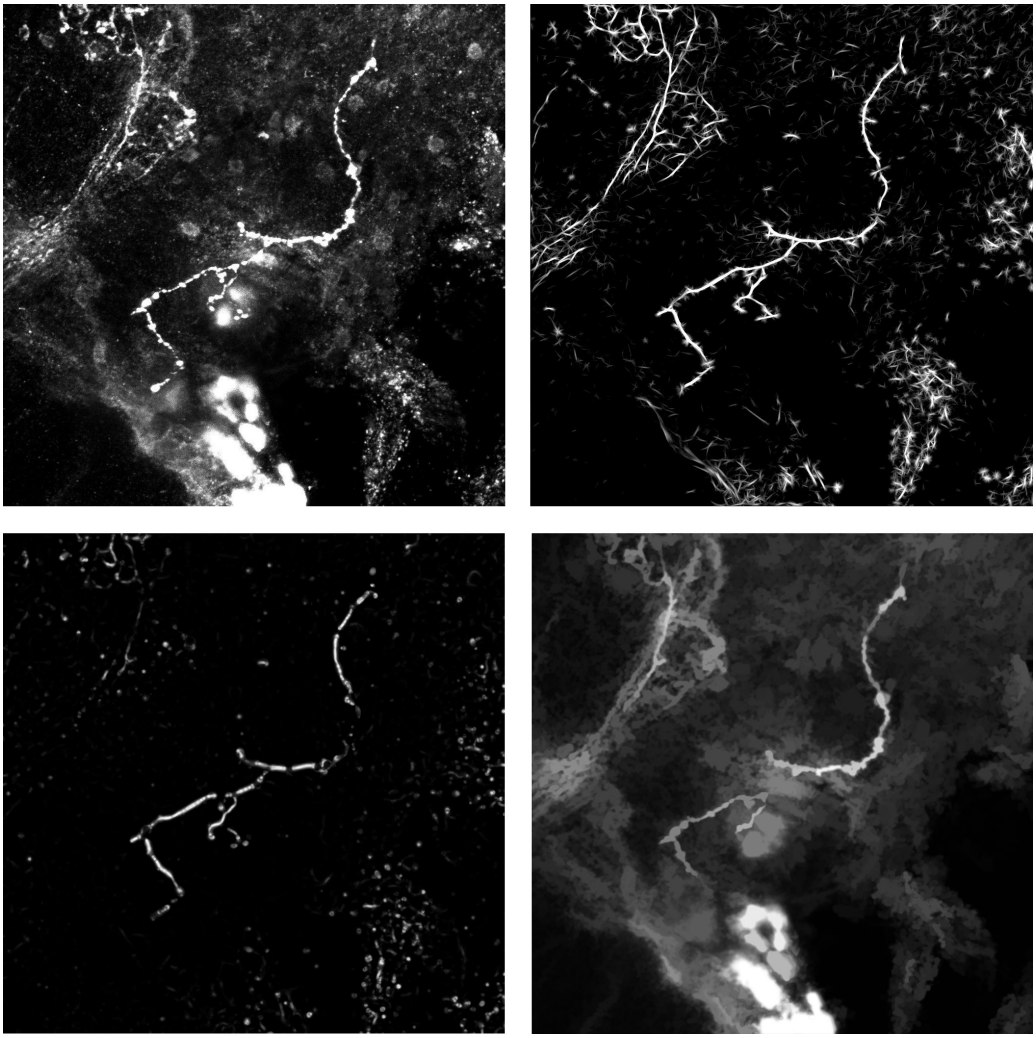


Figure 5.5: Comparison between the Gabor filament enhancer and the methods proposed in [Frangi 1998, Wilkinson 2001]: Top: original image (left) and Gabor filtering (right). Bottom: Frangi et al. (left) and Wilkinson et al. (right) (maximum intensity projections).

used as a way of summarizing geometrical and topological properties of a shape (see Figure 5.7). It looses the width information that is not relevant for our application.

We have used the implementation proposed in [Uitert 2007]. This skeletonization algorithm combines the level set method, an euclidean distance field calculation and fast matching propagation to compute a skeleton that preserve the topological and geometrical properties of the original object. The method is able to automatically discard spurious branches that are smaller than an user defined threshold and remove any potential loops. Moreover, the algorithm presents the advantage of having no user defined parameters (except the minimum branch length).

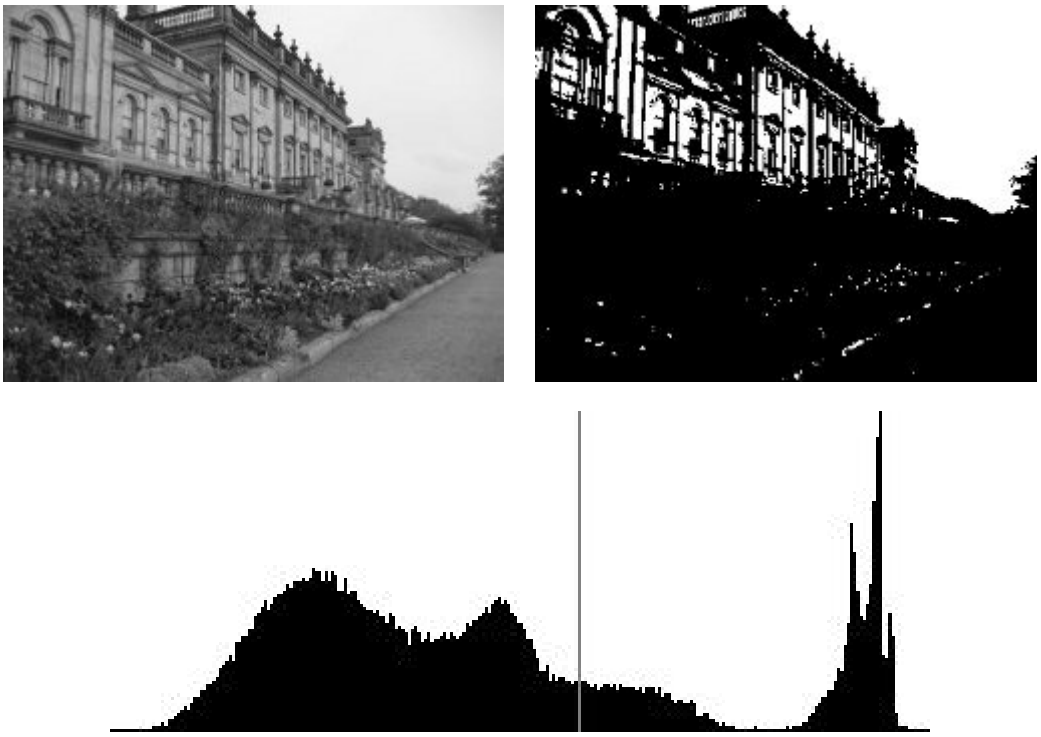


Figure 5.6: Example of Otsu thresholding: Original image (top left), binary image (top right) and histogram (bottom) (taken from http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html)

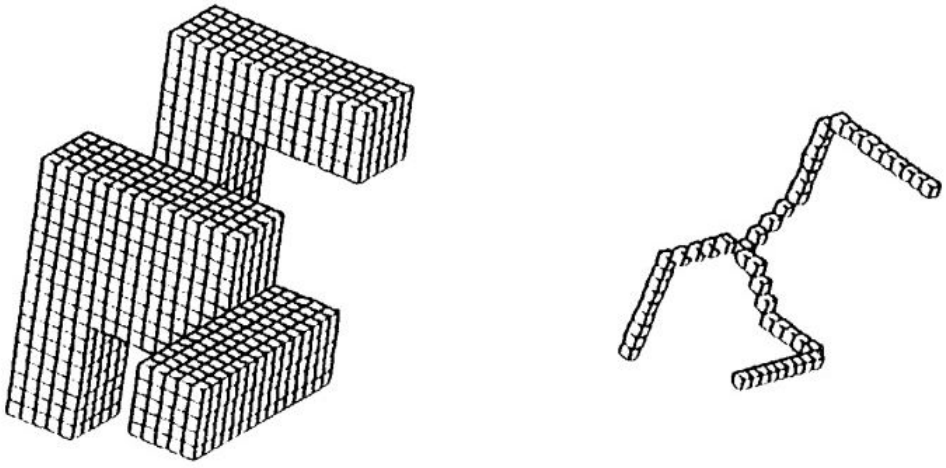


Figure 5.7: Example of the skeletonization of a 3D object (taken and modified from [Lee 1994]).

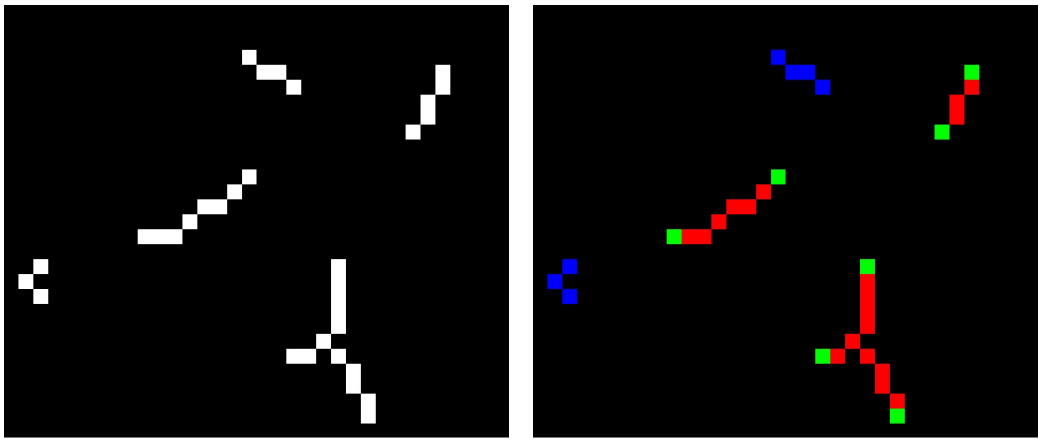


Figure 5.8: Original image (left) and its segmentation into tokens (right): islands (blue), segments (red) and segment ends (green).

5.2.3 Gap Filling

It is very common that, once the skeleton has been extracted, it presents discontinuities (gaps) due to the binarization in a noisy context. Several algorithms exist in the literature to solve this problem. In particular, we chose the method proposed in [Risser 2008]. In this work, the authors propose a tensor voting approach for filling gaps in 3D micro-vascular networks. The algorithm is divided into four steps. The first step (referred to as segmentation) involves separating the skeleton segments (referred to as a tokens) into 3 categories (see Figure 5.8):

- Island token: segments which are not connected to other segments or only connected from one side, and have a length of less that a given value (referred to as β).
- Segment token: segments which are either connected from both sides to other segments or have one or two free ends and length greater than β.
- Segment-end token : the last pixel of the free segment end for segments of length greater than β.

The second step is the creation of a tensor field T. It involves assigning to each point of the image a tensor (a 3x3 symmetric non-negative matrix). Each tensor depends on the contribution of all the tokens in the image:

$$T(i,j,k) = \sum_{n=1}^{N} TE_n(i,j,k) + \sum_{m=1}^{M} TI_m(i,j,k) + \sum_{q=1}^{Q} TS_q(i,j,k) \tag{5.16}$$

where TE_n is the tensor field created by the n^{th} segment-end token, TI_m the one created by the m^{th} island token and TS_q the one created by the q^{th} segment token. The calculation of these contributions depends on the type and properties of each token (see Figure 5.9.

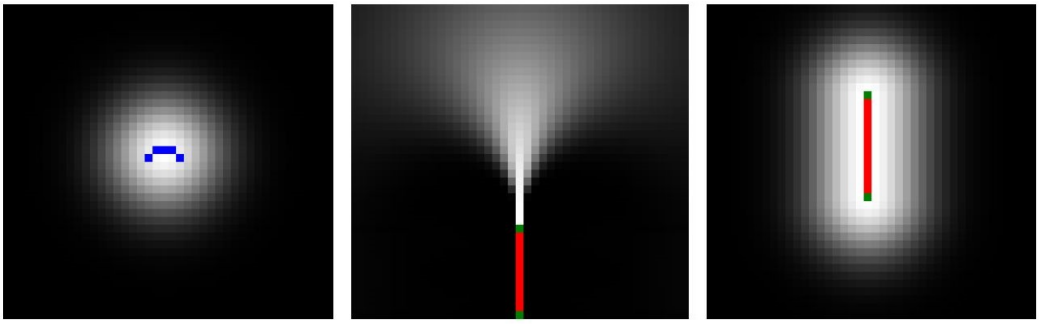


Figure 5.9: Tensor fields for the different types of tokens: island (left), segment end (middle) and segment (right).

Let us start by considering the island tokens. Since they do not have any reliable orientation information, we construct the tensor field around it to be equally strong in any direction, and to depend only on the distance from the token. Therefore, given an island token centered at point C, the tensor generated at point P of the image is given by:

$$TI = \overrightarrow{cp} \otimes \overrightarrow{cp}, \qquad \overrightarrow{cp} = \frac{\overrightarrow{CP}}{\|\overrightarrow{CP}\|} \exp\left(-\frac{\|\overrightarrow{CP}\|^2}{\sigma_t^2}\right) \tag{5.17}$$

where σ_t is a parameter of the algorithm related to the variation of intensity of the field with the distance.

Let O be a segment-end token (the end point of segment S), P a point in the image and $\overrightarrow{V}$ the tangent to segment S in O. We then define vector $\overrightarrow{W} = 2\overrightarrow{OP}(\overrightarrow{OP}.\overrightarrow{V}) - \overrightarrow{V}$. This vector is the oriented tangent at P to the circle C which contains O and P and which is tangent to V at O. Then, the tensor TE is defined as:

$$TE = \overrightarrow{w} \otimes \overrightarrow{w}, \qquad \overrightarrow{w} = \frac{\overrightarrow{W}}{\|\overrightarrow{W}\|} \exp\left(-\frac{r^2 + c\varphi^2}{\sigma_t^2}\right) \tag{5.18}$$

where r is the length of the arc $\widehat{OP}$ of circle C, φ its curvature and c a parameter of the algorithm related to the ratio between the distance and the curvature (see Figure 5.10).

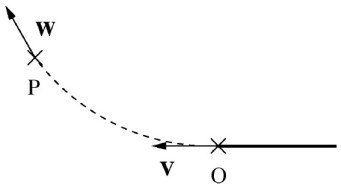


Figure 5.10: Arc of circle C used for the construction of the tensor field expressing a segment-end token represented with dotted lines between points O and P (taken from [Risser 2008]).

Finally, the tensor field TS associated to a segment token S is defined as follows:

$$TS = \overrightarrow{np} \otimes \overrightarrow{np}, \qquad \overrightarrow{np} = \frac{\overrightarrow{NP}}{\|\overrightarrow{NP}\|} \exp\left(-\frac{(\|\overrightarrow{NP}\|)^2}{\sigma_t^2}\right) \tag{5.19}$$

where P is a point of the image and N is the closest point of segment S to the point P.

The different parameters are set according to the author's recommendations in [Risser 2008]. Both σ_t and c were fixed for all our experiments and are thus not user dependent. The chosen values were:

$$\begin{cases} \sigma_t = 10 \\ c = 0.12 \end{cases} \tag{5.20}$$

The next step consists in generating a scalar vector field called the saliency map $S(i, j, k)$ (see Figure 5.11). For each point in the image, the value of the field is calculated in the following way:

$$S(i, j, k) = \lambda_1(i, j, k) - \lambda_2(i, j, k) \tag{5.21}$$

where $\lambda_{1,2}(i, j, k)$ are the two biggest eigenvalues (in absolute value) of the tensor field T in the point (i, j, k).

The final step is the reconnection step. The algorithm reconnects the paths along the points with maximum saliency. Starting from each segment end token the algorithms walks the paths of minimum saliency, thus creating paths that connect the different tokens in the original image. A path is stopped if it intersects a token or if a minimum saliency (a parameter of the algorithm) is reached. In this latter case, the path is discarded and the algorithm moves on to another segment end token.

In order to improve the speed of the algorithm, a small modification was introduced. The tensor map is not computed for all voxels in the image. Instead, cones of user defined height and radius are generated on each end point. Next, the tensors are computed in these regions (the cones) only if other points belonging to the skeleton fall inside.

At the end of the extraction process, a connected component labeling is performed and only the largest one is kept. This component corresponds to the axon.

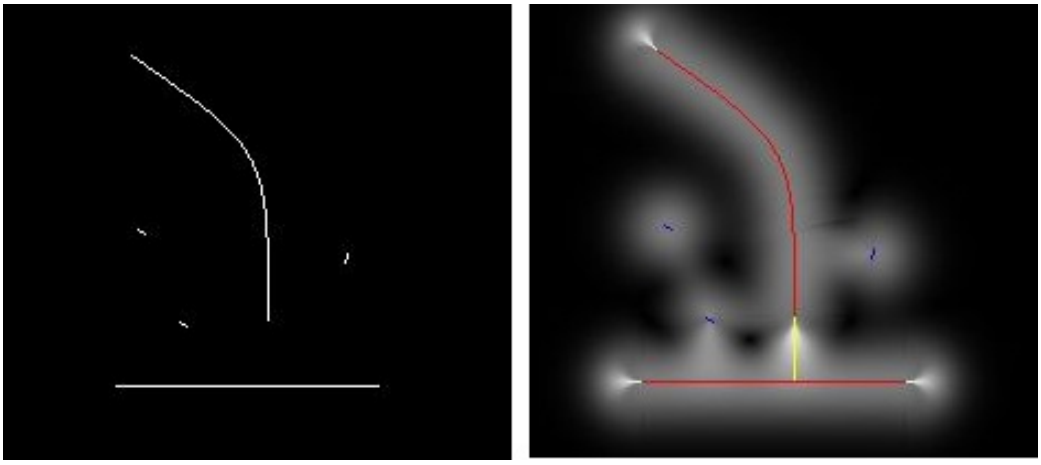


Figure 5.11: Original image (left) and corresponding saliency map (right). The gap to be filled is marked in yellow.

5.2.4 Post Processing

On occasions, the skeleton obtained after the gap filling can contain loops. This can occur when branches are joined with one another or with the main axon. We propose to perform a final post-processing step consisting on re running the skeletonization method described in Section 5.2.2. This will allow us to eliminate any loops and remove any remaining spurious branches. The minimum branch length was set to ten pixels. This value was calculated taking into account the minimum branch length of all the images in the database.

5.3 Validation

The proposed method (PM) was validated on a subset of the IBV image database introduced in Chapter 3. A group of 12 3D images, each containing one axonal tree, different noise levels and tree complexities were considered. Both normal and mutant axons were considered. The images were manually segmented

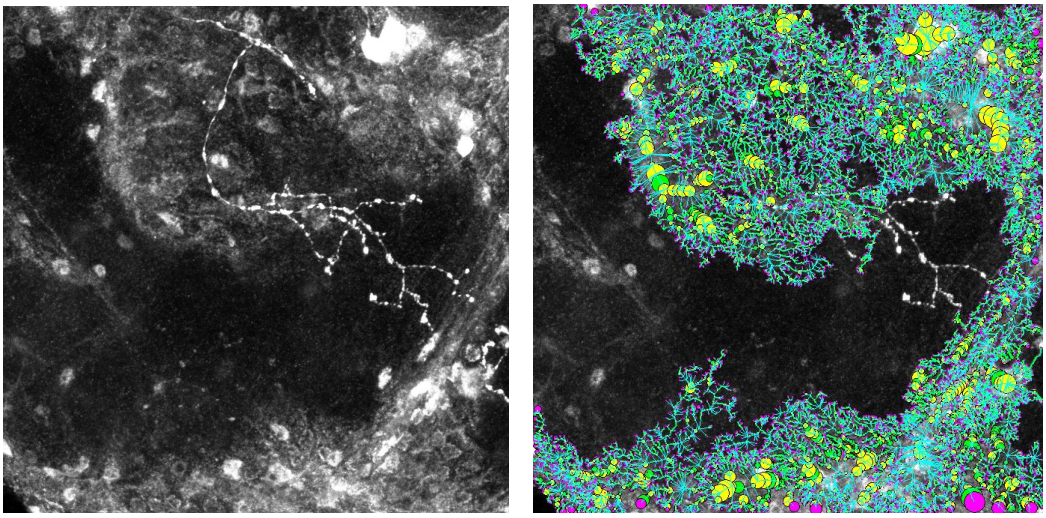


Figure 5.12: Results with NeuronStudio for an image of the database. The algorithm is unable to differentiate between the axonal tree and the background structures.

by an experienced biologist and the results compared with the ones obtained by our algorithm. In addition, we have segmented the images with V3D [Peng 2010] and NeuronStudio [Wearne 2005]. V3D is available freely in the Vaa3D software (http://www.vaa3d.org/), while NeuronStudio is can be downloaded from the authors website (http://research.mssm.edu/cnic/tools-ns.html). Since the results obtained by NeuronStudio directly on our images were not good enough (see Figure 5.12), we have decided to apply this method on the filtered images instead. This improves NeuronStudio results significantly.

The parameters of each of the three methods were tuned to maximize their performance. In some cases, this involved changing the parameters from image to image.

To quantitatively measure the quality of the results, we have used the following criteria:

- Directional Hamming Distance (Hamming) [Pont-Tuset 2013]: Given the images I and $I^{'}$, the distance is defined as:

$$D_H(I, I^{'}) = n - \sum_{R^{'} \in I^{'}} \max_{R \in I} |R^{'} \cap R| \tag{5.22}$$

  where $R, R^{'}$ are regions in $I, I^{'}$ respectively and n is the number of pixels in the images. If should be noted that this distance is not symmetric.

- Patt's Figure of Merit (FOM) [Chabrier 2004]: given the contours $C, C^{'}$ made

up of all the lines in images $I, I^{'}$ respectively, the criterion is defined as:

$$FOM(I, I^{'}) = \frac{1}{\max(|C|, |C'|} \sum_{i \in C'} \frac{1}{1 + d^2(i)} \tag{5.23}$$

where $|.|$ represents the cardinal function and $d(i)$ is the distance between pixel i of $C^{'}$ and the nearest pixel of C.

- Hausdorff Distance (Hausdorff) [Chabrier 2004]: given the pixel sets $P, P^{'}$ belonging to images $I, I^{'}$ respectively, the distance is defines as:

$$H(P, P^{'}) = \max(h(P, P^{'}), h(P^{'}, P)) \tag{5.24}$$

where $h(P, P^{'}) = \max_{t_i \in P} \min_{s_j \in P'} \|t_i - s_j\|$ and t, s are the pixels in each set. The value of $H(P, P^{'})$ can be interpreted as the maximum distance between the pixels in P and $P^{'}$.

- Root Mean Square Error (RMSE): given the pixel sets $P, P^{'}$ belonging to images $I, I^{'}$ respectively, the criterion is defines as:

$$RMSE(I, I^{'}) = \frac{\sum_{i \in P} d^2(i)}{|P|} \tag{5.25}$$

where $|.|$ represents the cardinal function and $d(i)$ is the distance between pixel i of P and the nearest pixel of $P^{'}$.

Note that for the Hausdorff distance and RMSE criteria, the lower the value the better the segmentation is. The opposite is true for the other two criteria.

5.3.1 Results

Figures 5.13 and 5.14 show the results obtained on each step of the pipeline for images of the data set. The proposed method is able to extract the axons even on images with significant noise. Figure 5.15 shows in more detail how the gap filling step is able to connect the gaps present in a skeleton. This step is very important since the skeletons usually present discontinuities. However, it is also a delicate step. Sometimes the algorithm can connect filaments that should not be connected. This can not only alter the original morphology of the tree, but more importantly, introduce unwanted loops. This loops should be corrected later on in a post-processing stage.

Figure 5.16 compares our results with the manual segmentations for two different cases. The figure shows how the axon extracted by our method is very similar to

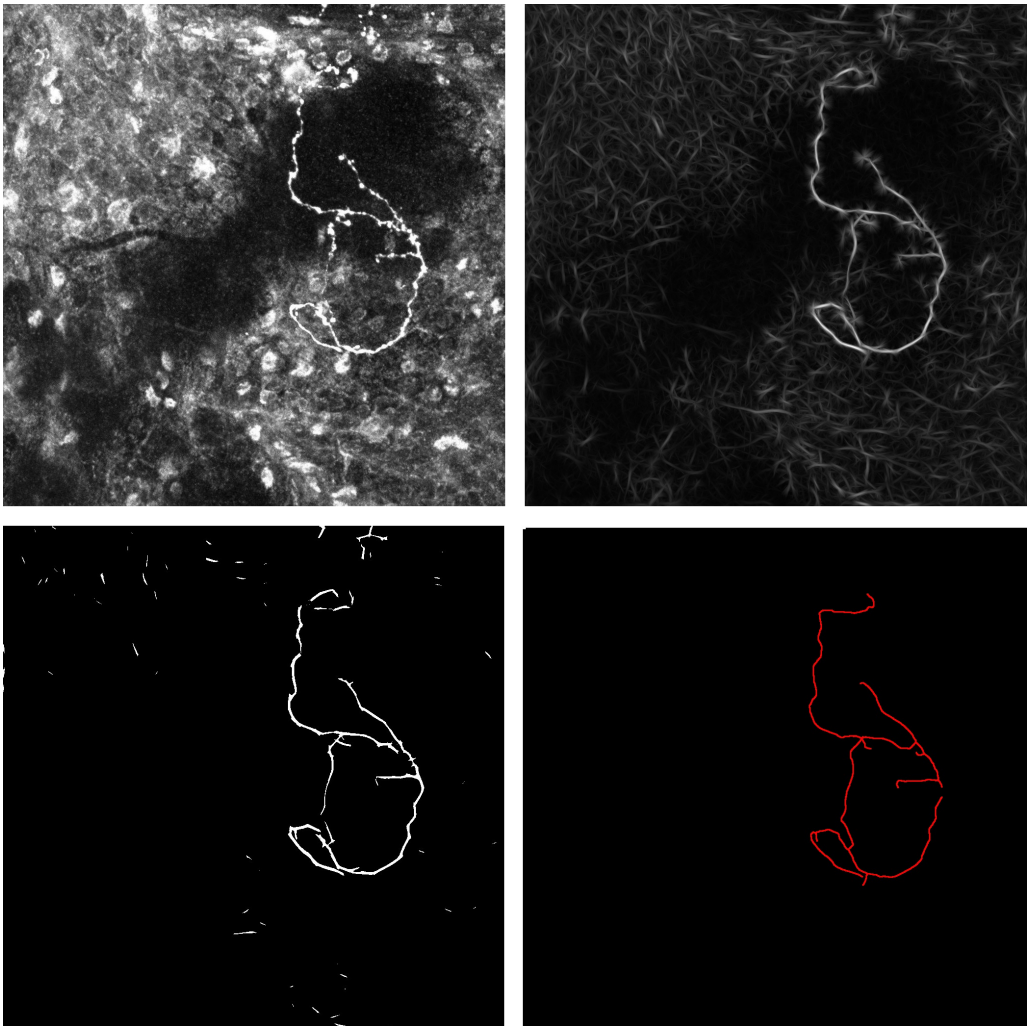


Figure 5.13: Results obtained on each step of the algorithm for one image stack. Top: original image (left) and filament enhancement (right). Bottom: binarization (left) and final result (right) (maximum intensity projections).

the one segmented by the biologist. However, on occasions they have short branches that are not present in the manual segmentations. This can be solved by increasing the length threshold in the skeleton pruning step.

The quantitative evaluation results are presented in Figures (5.17, 5.18, 5.19, 5.20) and Table 5.1. For the Hamming, FOM and RMSE criteria, our method outperforms V3D and NeuronStudio in all the images. The difference in performance between our method and the the other two is considerable for FOM and RMSE, but minor for Hamming. On the other hand, the Hausdorff criterion shows a more balanced performance between the methods. In some cases (such as in Figure 5.21) our method outperforms V3D. For this image we can see how V3D joined the end

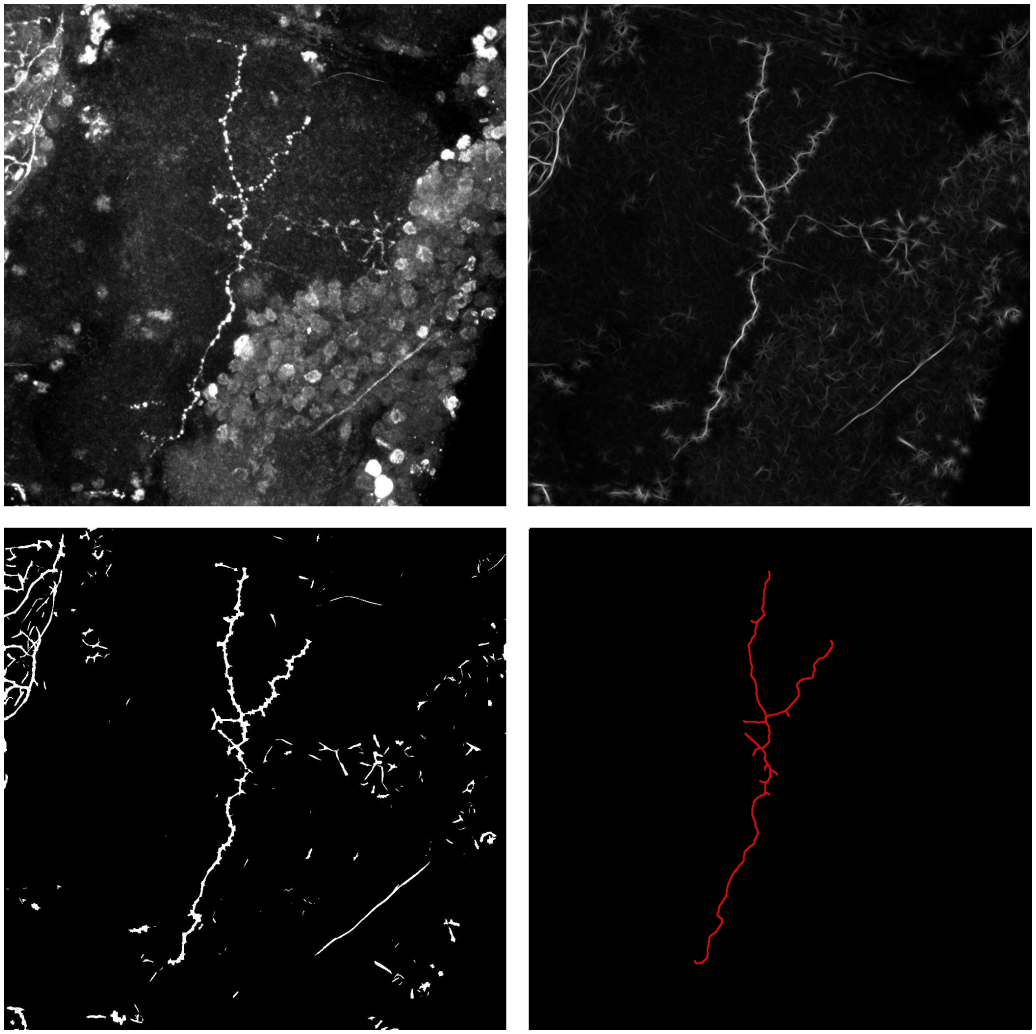


Figure 5.14: Results obtained on each step of the algorithm for one image stack. Top: original image (left) and filament enhancement (right). Bottom: binarization (left) and final result (right) (maximum intensity projections).

points through an incorrect path, thus generating false branches that are not present in the ground truth. This can be due to the fact that this image presents a considerable noise level. In particular, the image presents very bright structures in the top section of the image that the algorithm mistakes for a branch. On the contrary, in less noisy images where the axonal tree is clearly visible (such as in Figure 5.22), V3D obtains a better result. In this case, our results presents some small spurious branches while V3D segmentation is much cleaner and closer to the ground truth. This is to be expected since V3D is using the information input by the user (the end points) when segmenting the image. Finally, NeuronStudio is outperformed by the two other methods in all images except one. This concurs with the results

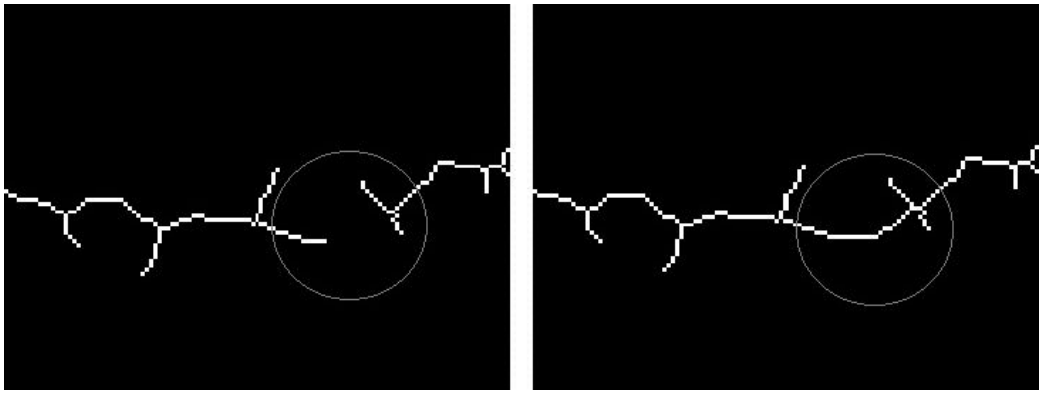


Figure 5.15: Result before (left) and after (right) the gap filling step (maximum intensity projections). The gray circle marks the gap to be filled.

Table 5.1: Quantitative evaluation: comparison of the proposed method (PM), V3D and NeuronStudio with the manual segmentation using the criteria described in 5.3 (mean and standard deviation)

Criterion (Method)	Mean	Standard Deviation
Hamming (PM)	1.3368E7	3.5523E6
Hamming (V3D)	1.3369E7	3.5524E6
Hamming (NeuronStudio)	1.3369E7	3.5524E6
FOM (PM)	0.34	0.08
FOM (V3D)	0.09	0.03
FOM (NeuronStudio)	0.01	0.04
Hausdorff (PM)	154.18	79.19
Hausdorff (V3D)	145.97	107.29
Hausdorff (NeuronStudio)	182.71	91.35
RMSE (PM)	2.83	0.54
RMSE (V3D)	4.73	1.20
RMSE (NeuronStudio)	9.10	1.11

presented in [Peng 2010], where the authors show that V3D obtains a better overall performance than NeuronStudio. In particular, Figure 5.23 presents a comparison between the three methods and the ground truth for one of the images.

5.4 Conclusions

An algorithm for the automatic extraction of axons from fluorescent confocal microscopy images was proposed. This method relies on several steps to perform the extraction: filament enhancement, binarization, skeletonization and gap filling. Due to the voxel anisotropy of our images, the filament enhancement step is performed on

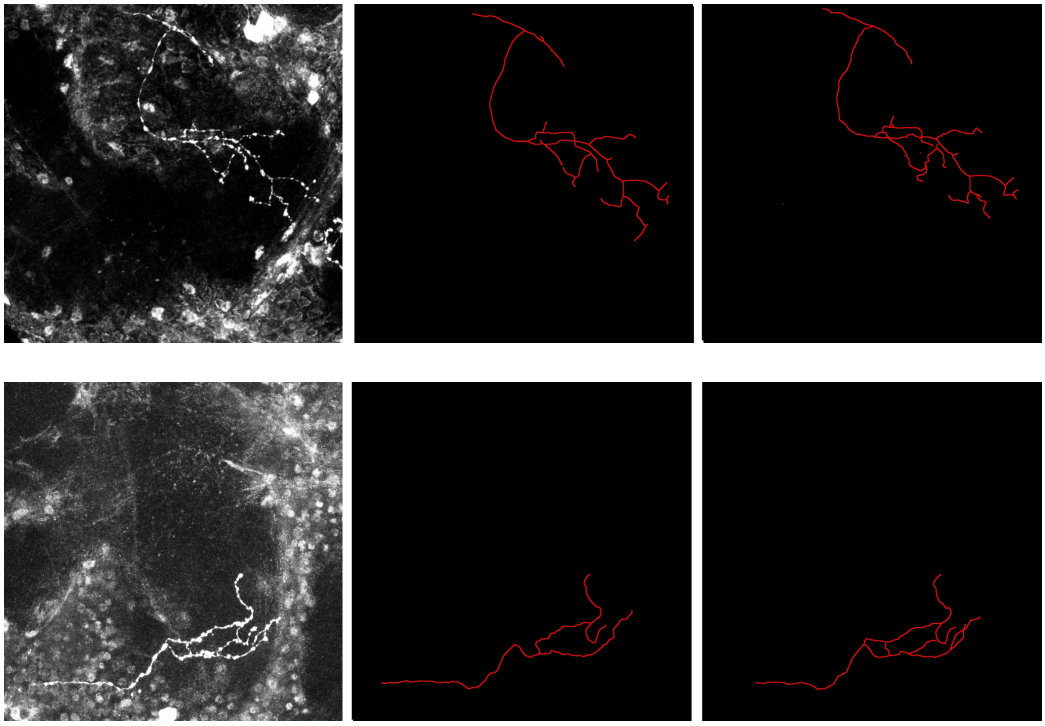


Figure 5.16: Comparison between original image (left), our result (middle) and ground truth (right) for two images (maximum intensity projections).

each 2D slice separately while the extraction (skeleton and gap filling) is performed in 3D.

The method performance was tested on 12 3D images of the IBV database and the results quantitatively compared with the ground truth provided by an experienced biologist and with the results obtain by two other segmentation methods (V3D and NeuronStudio). Results show that our method outperforms the other two in most cases, and that the proposed technique has the potential of helping biologists to extract axonal trees from confocal microscope images. This is the first step necessary if one intends to perform a morphological analysis of axons.

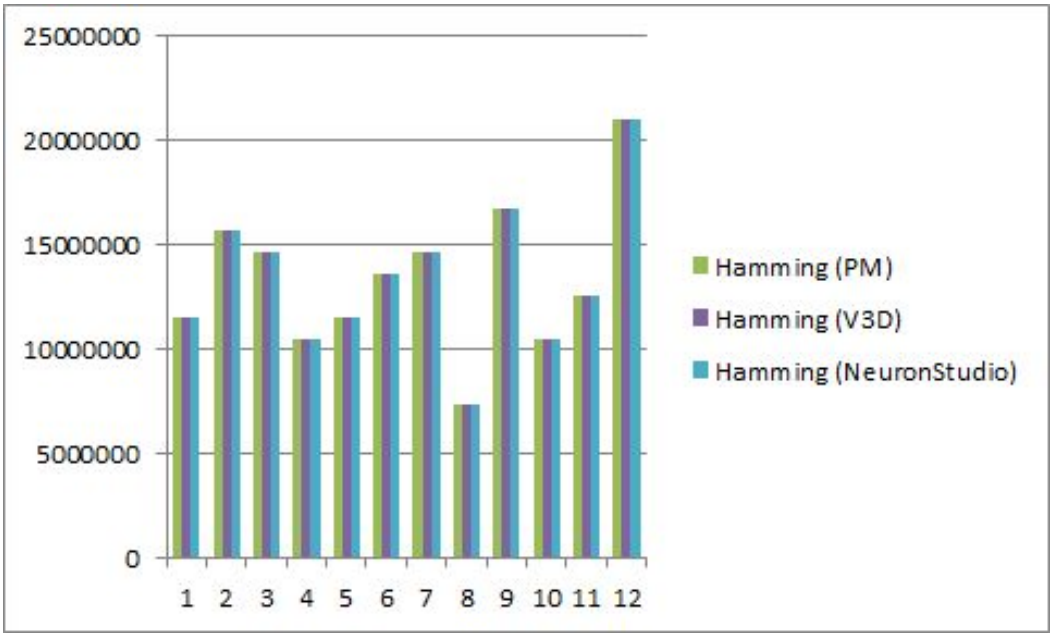


Figure 5.17: Quantitative evaluation: comparison between the proposed method (PM), V3D and NeuronStudio with the manual segmentation using the Hamming criterion (evaluation on each of the 12 considered images)

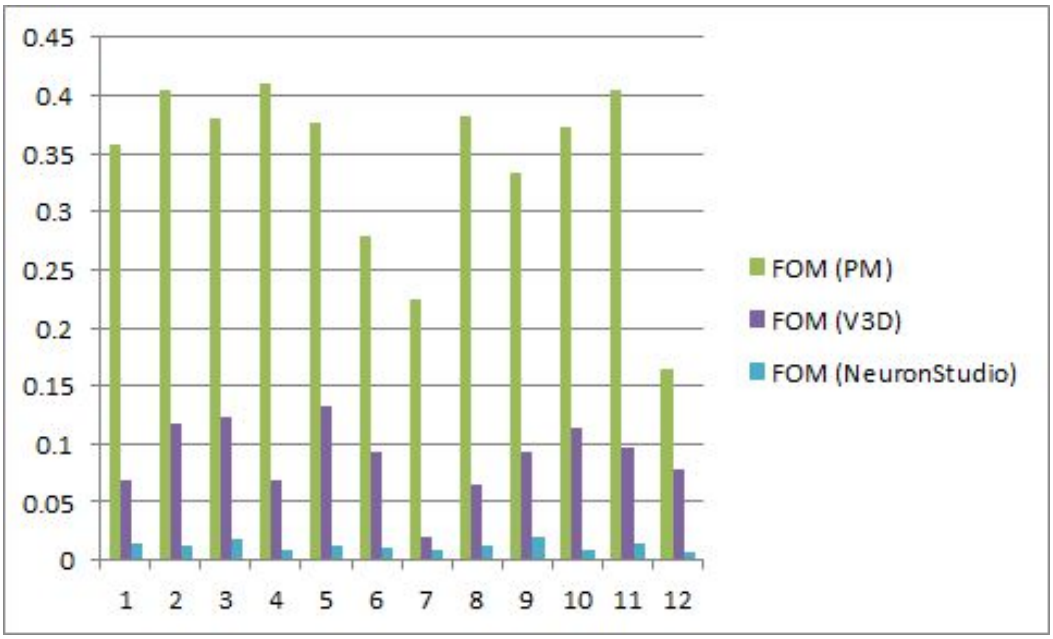


Figure 5.18: Quantitative evaluation: comparison between the proposed method (PM), V3D and NeuronStudio with the manual segmentation using the FOM criterion (evaluation on each of the 12 considered images)

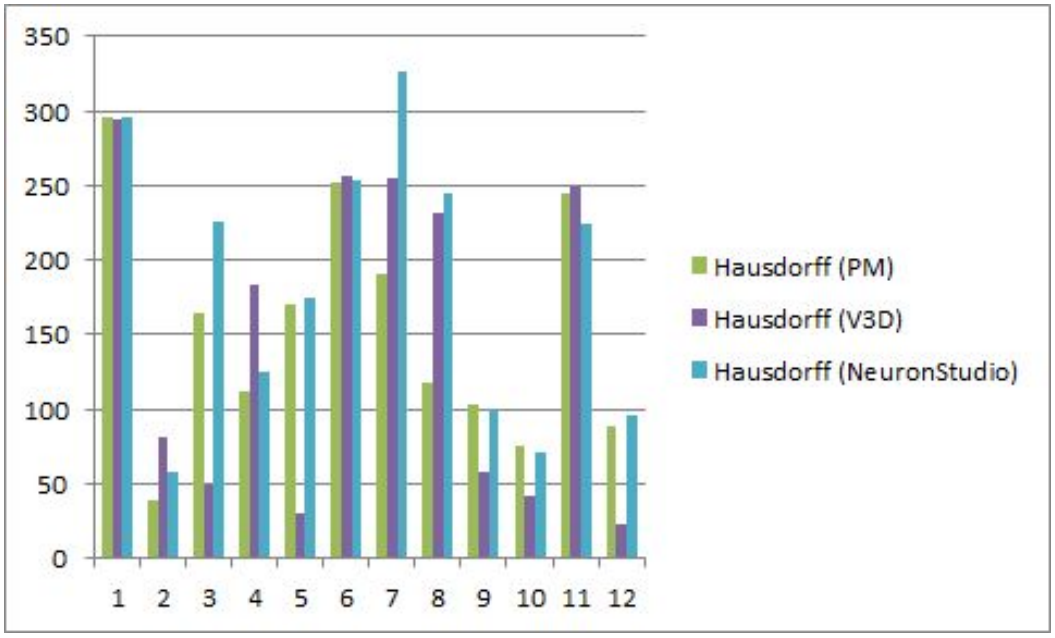


Figure 5.19: Quantitative evaluation: comparison between the proposed method (PM), V3D and NeuronStudio with the manual segmentation using the Hausdorff criterion (evaluation on each of the 12 considered images)

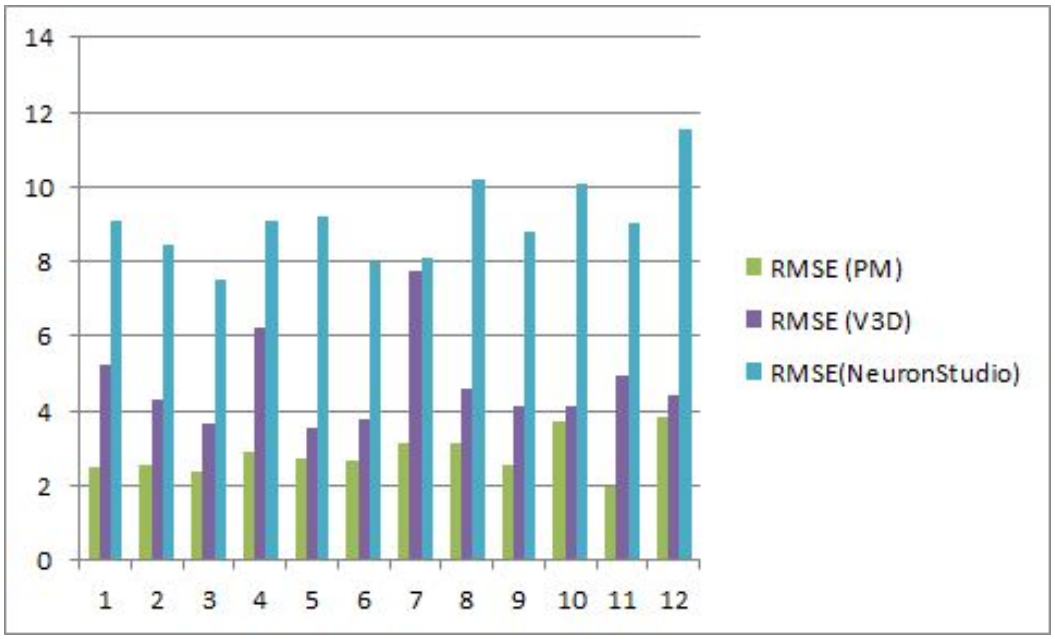


Figure 5.20: Quantitative evaluation: comparison between the proposed method (PM), V3D and NeuronStudio with the manual segmentation using the RMSE criterion (evaluation on each of the 12 considered images)

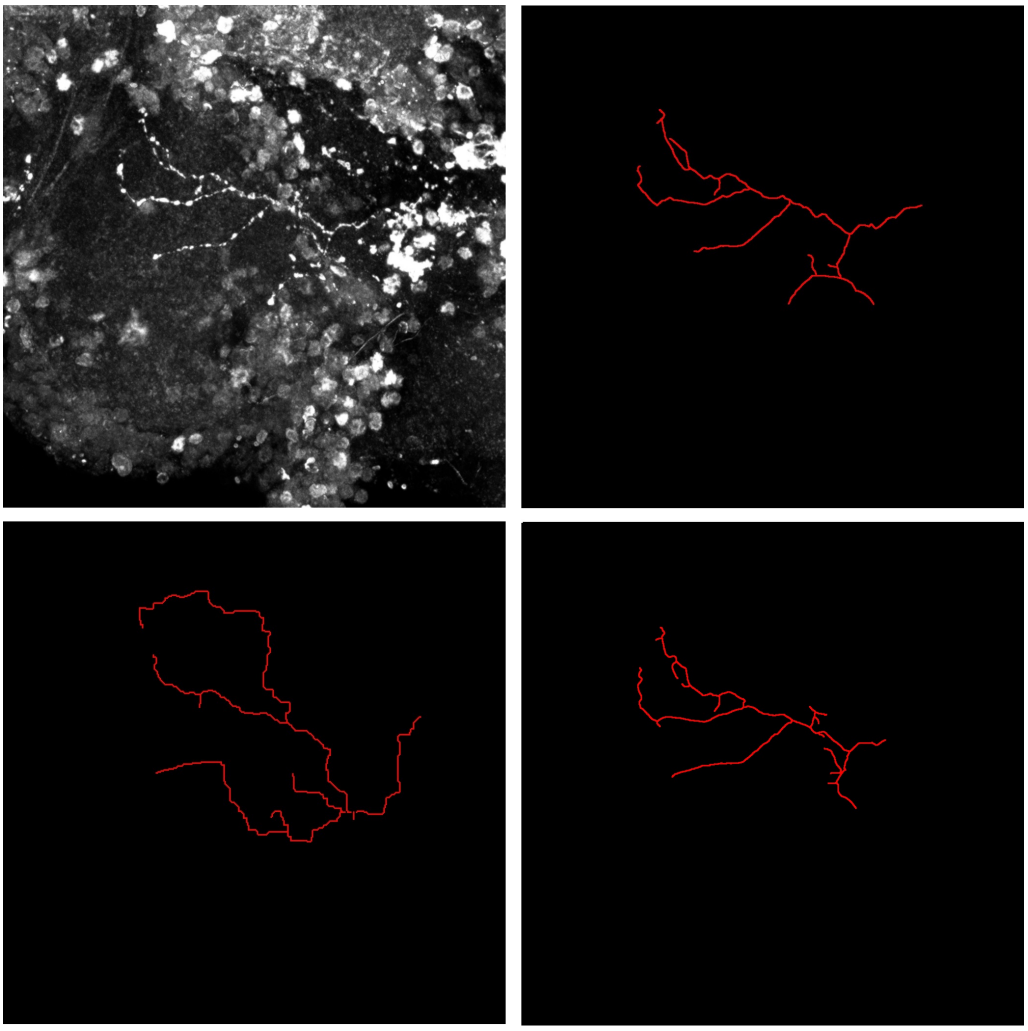


Figure 5.21: Comparison between original image (top left), the ground truth (top right), V3D (bottom left) and our result (bottom right) (maximum intensity projections).

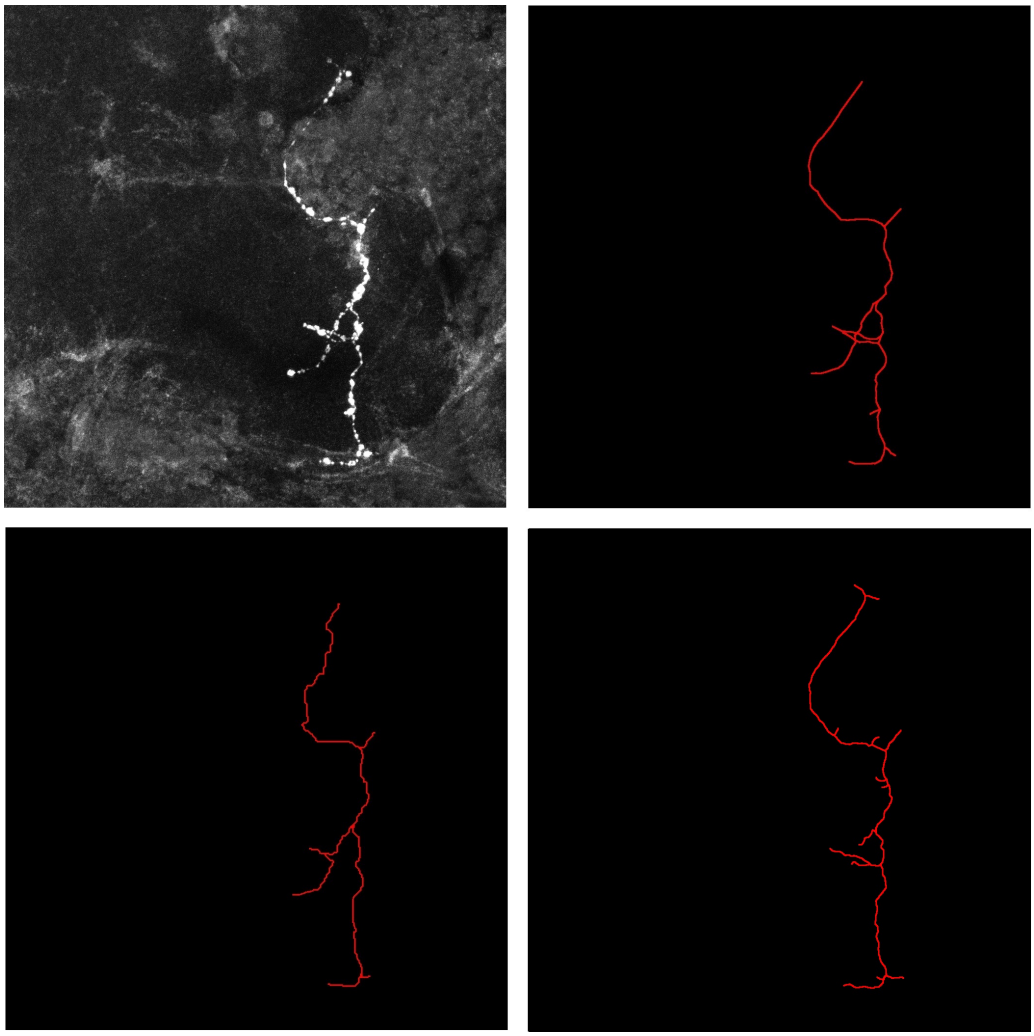


Figure 5.22: Comparison between original image (top left), the ground truth (top right), V3D (bottom left) and our result (bottom right) (maximum intensity projections).

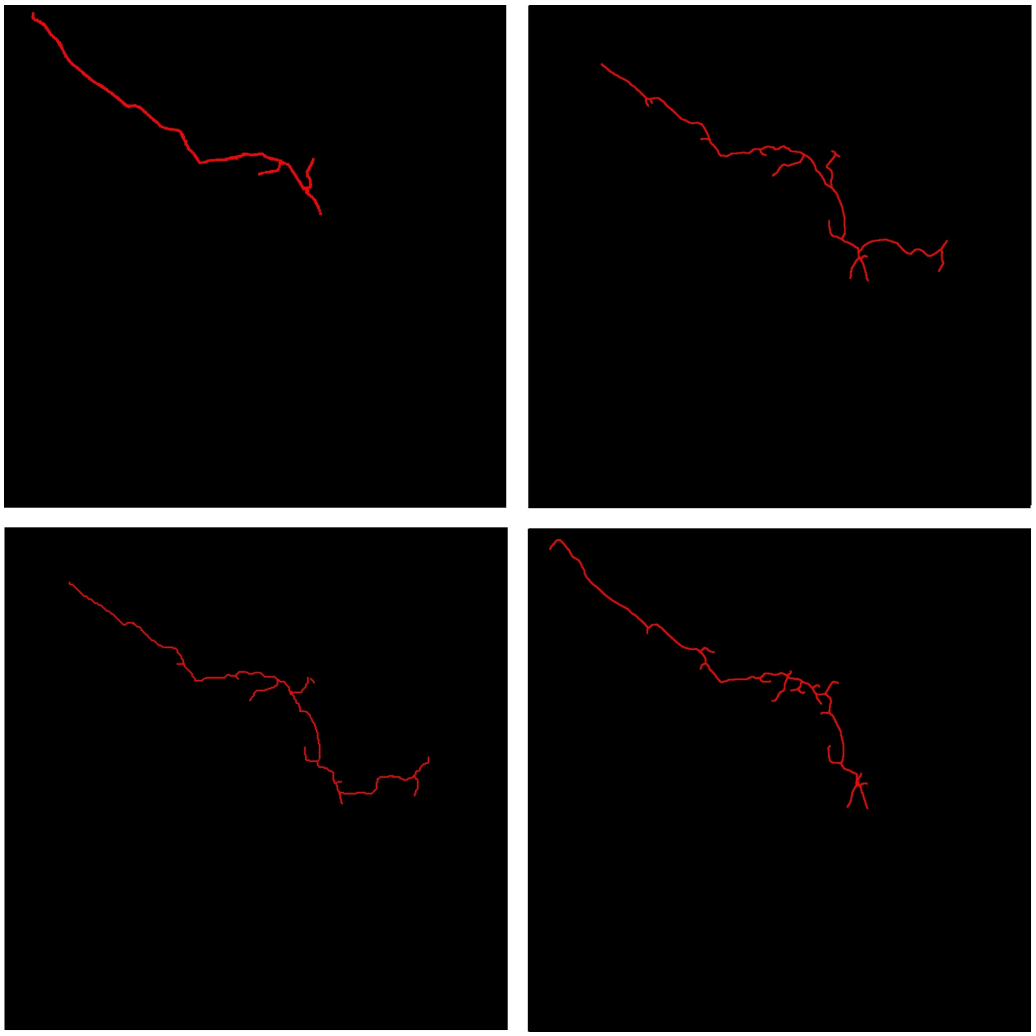


Figure 5.23: Comparison between NeuronStudio (top left), the ground truth (top right), V3D (bottom left) and our result (bottom right) (maximum intensity projections).

Chapter 6

An Elastic Shape Analysis Method for the Comparison and Classification of Trees

Contents

In this chapter we present a tree comparison method based on the Elastic Shape Analysis Framework that takes into account both topological and geometrical information. Moreover, we present a method capable of computing the mean shape of a population of trees. Although the method could potentially be applied to different types of trees, we concentrate on the comparison of axonal trees. The performance was tested on the databases presented in Chapter 3. In total, over 200 reconstructions were considered. We have compared the classification performance of our method against three other algorithms: TED, Path2PAth and a feature based approach. Results showed that the proposed method better distinguishes between the populations. In addition, we present the mean shape of each population. These shapes present a more complete picture of the morphological characteristics of each population, compared to the average value of some predefined features.

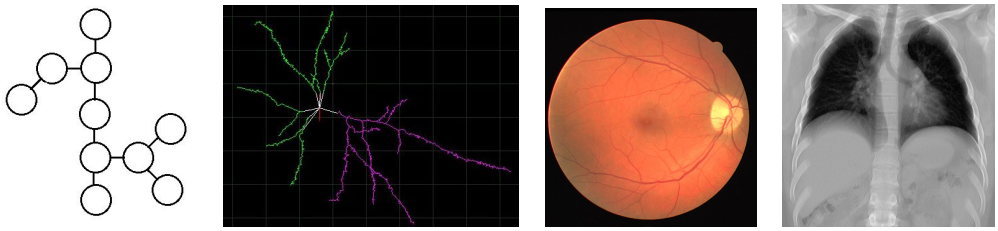


Figure 6.1: From left to right: Diagram of a rooted tree, neuron, blood vessels and lung airways. Taken from http://neuromorpho.org/, http://www.isi.uu.nl/Research/Databases/DRIVE/ and http://image.diku.dk/exact/index.php

6.1 Introduction

Trees are undirected graphs that have no cycles. Given two nodes, there is only one path between them. In particular, if one of the nodes is considered to be the root, it is called a rooted tree. This type of graph has been thoroughly studied since many naturally occurring structures can be described by it.

In the field of biomedical imaging, some important structures that can be modeled as trees are neurons, blood vessels and lung airways (see Figure 6.1).

As we have discussed before, certain morphological characteristics of these structures are tightly linked with their ability to function properly. In particular, the morphology of neuronal structures is related to their ability to interconnect and to process information. Therefore, it is important to develop frameworks for the analysis of their shapes and to quantify differences in their structures. By shapes, we not only mean the topological information defining the tree but also the geometry of the edges.

At present, there are a number of methods for the comparison of tree structures. One of the best known metrics is TED [Tai 1979, Selkow 1977, Zhang 1996]. This method is a pure topological metric based on the tree-edit distance between unordered labeled tree-graphs. TED has already been successfully applied to neuronal morphology analysis [Heumann 2009] as well as the analysis of other tree like structures such as RNA secondary structures [Zhang 1989].

A different approach was proposed in [Feragen 2010a, Feragen 2010b]. The authors constructed a shape space for rooted tree-like shapes and studied different metrics on this space. The reported distance takes into account both topological and geometrical information of the tree. They applied their method to the study of airways from pulmonary CT scans. This method presents good results and a strong mathematical foundation. However, its application is restricted to small poorly

branched trees due to its computational complexity.

Other methods have been designed for more specific applications. One example is the DIADEM metric [Gillette 2011], which was developed to serve as the evaluation method for the DIADEM Challenge [1]. It was specifically designed for comparing different tracings of the same neuron, and is therefore not adapted to the comparison of trees in a more general setting.

Finally, Basu *et al.* proposed Path2Path [Basu 2011], a metric that takes into account both topological and geometrical information of trees. The method is based on finding a deformation energy between two trees, which depends on the L_1 norm between the curves that make up the trees..

In this chapter, we present a new method for comparing tree-like shapes. The method takes into account both topological and geometrical information, and is specifically designed for analyzing rooted trees that consist of a main curve, branches and sub branches along it. It is based on the Elastic Shape Analysis Framework [Srivastava 2011], which was originally developed for comparing shapes of curves in the Euclidean space. We extend this work by defining a new distance between trees. Moreover, we propose an iterative algorithm for calculating the mean shape of a populations of trees.

Even though the proposed method could be used for the comparison of different types of trees, we concentrate on the comparison and classification of axonal trees. We have validated our approach on two different databases. The first set consists on 131 reconstruction of four different populations of neurons taken from the NeuroMorpho.org open database (see Chapter 3). The second set is the IBV database presented in Chapter 3, which is composed of 61 (20 normal, 24 type one mutant and 17 type two mutant) real 3D confocal microscopy images manually segmented by an experienced biologist.

We calculate the mean shape of each population and evaluate the classification accuracy of our method. We validate our method against three other algorithms: TED, Path2Path (P2P) and a classification scheme that calculates features from the trees, followed by applying PCA and K-Means.

6.2 The Elastic Shape Analysis Framework

Let us consider a set of open curves in $\mathbb{R}^3$ (for example, the main curve and each branch of a tree). Under this framework, each curve $\beta : [0, 1] \rightarrow \mathbb{R}^3$ can be repa-

[1] http://diademchallenge.org/

rameterized using its square-root velocity function (SRVF):

$$q(t) = \frac{\dot{\beta(t)}}{\sqrt{\|\dot{\beta(t)}\|}} \tag{6.1}$$

where $\|x\|$ is the standard Euclidean norm in $\mathbb{R}^3$ and $\dot{\beta(t)}$ is the derivative of $\beta(t)$.

Curves that have the same shape but differ by rotation/reparameterization result in different q functions. We solve this non-unicity problem by considering the quotient space with respect to the group of shape-preserving transformations (translation, scale, rotation and reparameterization. Since q depends on the time derivative of β, the analysis will be independent of the position of the curve (obtaining a translation invariant representation). Moreover, curves can be rescaled to length one in order to impose scale invariance. We then only need to remove the last two remaining shape preserving transformations: rotation and reparameterization.

Let $SO(3)$ be the group of 3×3 rotation matrices and Γ be the group of all reparameterizations (set of continuous increasing functions from $[0,1]$ to $[0,1]$ such that $\gamma(0) = 0, \gamma(1) = 1$). Given a curve β with SRVF q, a rotation $O \in SO(3)$ and a reparametrization $\gamma \in \Gamma$, the transformed curve can be expressed as:

$$q^{\star} = \sqrt{\dot{\gamma}} O(q \circ \gamma) \tag{6.2}$$

We can then define a curve in a unique way:

$$[q] = \{O(q \circ \gamma)\sqrt{\dot{\gamma}} | O \in SO(3), \gamma \in \Gamma\} \tag{6.3}$$

Each class $[q]$ defines a unique shape and the set of all these equivalence classes is called the shape space S.

By considering the L^2 norm on the tangent spaces of this shape space, the authors force a Riemannian structure on S. This corresponds to assuming that the curves and their derivatives are square integrable. Moreover, this allows the computation of geodesics between points in S, which can be interpreted as the optimal elastic deformation from one curve into the other. The distance between two points in S is defined as the length of the geodesic between them.

Given two curves β_1, β_2 represented by their corresponding SRVF q_1, q_2, the first step for computing the geodesic between the two is fixing q_1 and finding the optimal rotation and parametrization of q_2 such that:

$$(O^*, \gamma^*) = \underset{O \in SO(3), \gamma \in \Gamma}{\arg\min} \|q_1 - \sqrt{\dot{\gamma}} O(q_2 \circ \gamma)\|^2 \tag{6.4}$$

The optimization over rotation can be achieved using singular value decomposition, while the optimization over the reparametrization requires dynamic programming. The geodesic path between q_1 and q_2^* is then given by:

$$\alpha(\tau) = (1 - \tau) q_1 + \tau (O^*(q_2, \gamma^*)) \tag{6.5}$$

while the distance between the two curves is given by:

$$d_{shape}(\beta_1, \beta_2) = \min_{\gamma \in \Gamma, O \in SO(3)} \|q_1 - O(q2, \gamma)\|_2 \tag{6.6}$$

The classical Elastic Shape Analysis Framework was extended in [Liu 2008] to include additional information in the shape analysis. This is achieved by including an additional function $\beta_c : [0, 1] \to \mathbb{R}^k$ (where k is an arbitrary finite dimension) to the original representation of curve β. As an example, if the method were used for object matching, the curves could represent the shapes of the objects and the additional β_c functions represent the locations of certain landmarks on the objects. In [Liu 2010] it is used for protein structure alignment, where β represents the backbone geometry and β_c encodes the secondary structure of a protein.

The two components are combined to form a new curve:

$$\beta(t) = \begin{bmatrix} \beta_s \\ b.\beta_c \end{bmatrix} \in \mathbb{R}^{3+k} \tag{6.7}$$

where $b > 0$ is a control parameter that weights the influence of the additional function in the matching process. The remaining steps are done in the same way as before but with the new curve β (although special precaution not to rotate the β_c component are required when eliminating the rotation transformation).

6.3 Extension to Trees

In this section, we extend the Elastic Shape Framework to define a distance between trees.

Let us consider two trees T_1 and T_2, each consisting of a main curve and several branches (and possibly sub branches) represented by 3D open curves in $\mathbb{R}^3$. Each branch forms an angle $\phi \in [0, \pi]$ with respect to its mother branch (the main curve in the case of first level branches) and an angle $\theta \in [0, 2\pi)$ with respect to the first

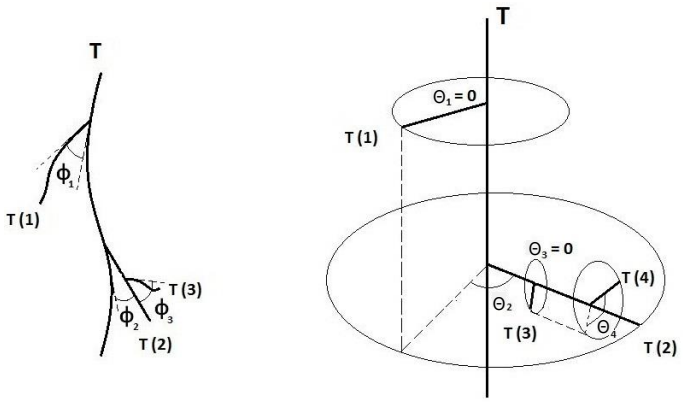


Figure 6.2: Tree diagrams showing how angles ϕ and θ are defined for a given branch. Angles ϕ_1 and ϕ_2 are defined with respect to the main branch, while ϕ_3 is defined with respect to $T(2)$ (its mother branch). A similar convention is taken for θ. In this example, the angle θ_2 associated to branch $T(2)$ is measured with respect to branch $T(1)$, while θ_4 is measured with respect to $T(3)$ (the first branch depending from its mother branch).

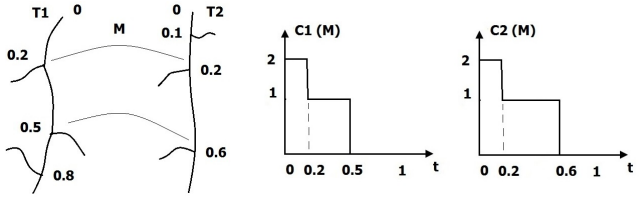


Figure 6.3: Tree diagrams (left) and their corresponding C functions for a given M (right).

branch of its mother branch (see Figure 6.2).

We start by defining the matching function M such that $M : (0, 1, 2, ...n) \times (0, 1, ..., m)$, where n and m are the number of branches in T_1 and T_2, respectively. This function matches the branches of the two trees as follows:

$$M(i,j) = \begin{cases} 1 & \text{if i matches j} \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

$M(i, 0) = 1$ and $M(0, j) = 1$ means i/j is assigned to nothing. In these cases we say we kill the branch. The distance between matched branches and between a branch and zero will be calculated differently, using Equations (6.12) and (6.14) (please refer to the example at the end of this section for additional clarification).

To improve the computational efficiency of the algorithm, we do not allow the matching of branches that are far away from each other. The maximum allowed separation between matched branches is a parameter of the algorithm, which was

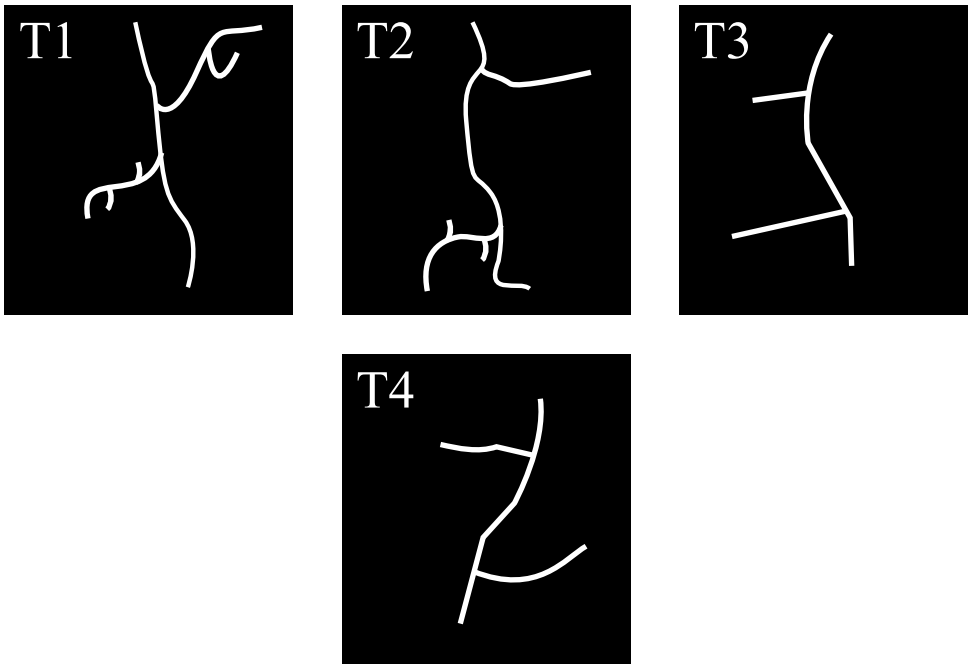


Figure 6.4: Four synthetic images (T_1 to T_4), each containing a simple tree with a main curve and some branches and sub branches.

set to forty percent of the length of the mother branch. Other constraints are also included, such as not allowing the crossed matching of branches (e.g, matching branches 1 and 2 of T_1 to branches 2 and 1 of T_2).

We then define a branch function C which indicates, for a given time t_c, how many branches remain after $\beta(t_c)$ (see Figure 6.3). We only take into account branches that have a match in the other tree. Therefore, the C function depends on M. C functions are defined for both trees and included as the fourth component of curves β_1, β_2. This allows us to take into account the difference of position of matched branches in the distance between trees.

Finally, we define recursively the distance between two trees T_1, T_2 as:

$$D(T_1, T_2) = \min_M [d((\beta_1(t), C1(t, M)), (\beta_2(t), C2(t, M))) + \\ + \sum_{(i,j)} \alpha_{i,j} M(i,j) D(T_1(i), T_2(j))] \tag{6.9}$$

where β_k is the main curve of tree k, C_k its branch function, M is the matching function, $\alpha_{i,j}$ a weight parameter and $D(T_1(i), T_2(j))$ is the distance between the matched branches (sub trees) of the two trees.

The first term of the distance embeds geometrical information on the main

branch by matching segments between branching points. Topological information is taken into account through the C function. The parameter b in Equation (6.7) weights the two characteristics. Sub branches are taken into account in a recursive manner in the second term. Note that the coefficients weight the different order of sub branches with respect to the main branch. Moreover, $\alpha_{i,j}$ controls how non matched branches ($M(i,0) = 1$ or $M(0,j) = 1$) are penalized. Typically, $\alpha_{i,j}$ equals a_1 if $i \neq 0$, $j \neq 0$ and it equals a_2 otherwise.

The distance between the two main curves is defined as:

$$\begin{aligned} d(((\beta_1(t), C1(t,M)), (\beta_2(t), C2(t,M))) = d_{length}(\beta_1(t), \beta_2(t)) + \\ + w_1 d_{shape}(\beta_1(t), C1(t,M)), (\beta_2(t), C2(t,M))) \end{aligned} \tag{6.10}$$

where d_{length} is defined as:

$$d_{length}(\beta_1(t), \beta_2(t)) = \frac{\|length(\beta_1(t)) - length(\beta_2(t))\|}{\|length(\beta_1(t)) + length(\beta_2(t))\|} \tag{6.11}$$

and d_{shape} is defined by Equation (6.6), using the C functions as the fourth components of the curves. The parameter w_1 controls the weight between the two terms.

In addition, the distance between two matched branches is defined as:

$$\begin{aligned} D(T_1(i), T_2(j)) = d_{length}(\beta_i(t), \beta_j(t)) + \\ w_1 d_{shape}(((\beta_i(t), Ci(t,M)), (\beta_j(t), Cj(t,M))) \\ + w_2 d_\phi(\phi_i, \phi_j) + w_3 d_\theta(\theta_i, \theta_j) \end{aligned} \tag{6.12}$$

where d_{length} and d_{shape} are defined in the same way as for the main curve, and the two remaining terms as:

$$\begin{aligned} d_\phi(\phi_1, \phi_2) = \|\phi_1 - \phi_2\|/\pi \\ d_\theta(\theta_1, \theta_2) = \|\theta_1 - \theta_2\|/2\pi \end{aligned} \tag{6.13}$$

Once again, the parameters w_i control the weight between the terms.

Finally, we define a cost for non matched branches as follows:

$$D(T_1(i), 0) = w_{kill} \tag{6.14}$$

where w_{kill} is a parameter of the algorithm.

The geodesic defined in Equation (6.5) is adapted to simple curves. To define the geodesic between trees, we consider that the transformations between matched

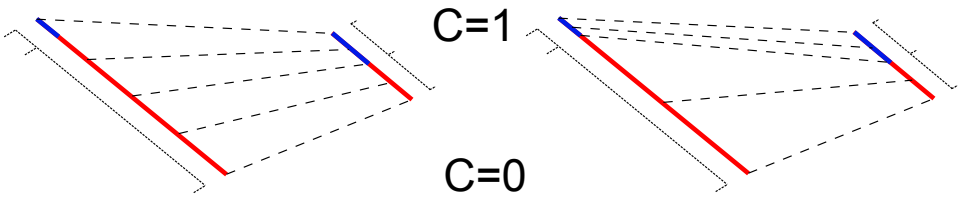


Figure 6.5: Registration between two curves including a simple step C function (blue for $C = 1$, red for $C = 0$), $b = 0$ (left) and $b = \infty$ (right). If b is big the registration is mostly determined by the C function.

branches and the creation/removal of branches occur uniformly during the same time interval $[0, 1]$.

To illustrate how the different parameters can influence the matching of the branches and the final results, let us consider a set of simple synthetic images (see Figure 6.4). Let us first consider trees T_1 and T_2. If the parameters are chosen in a given way, similar branches and sub branches should match each other, that is, $T_1(i)$ should match $T_2(i)$. In this case, the sub branch of branch $T_1(1)$ will have no match (it would be matched to 0, thus being "killed"). However, if we tune the parameters in a different way, this result could change significantly. For example, if the cost of killing a branch is relatively small compared to transforming one branch into another, we would be in a scenario where all the branches of T_1 are killed and all the branches of T_2 are being born (all branches matched to 0). On the other hand, if we consider trees T_3 and T_4, we can illustrate the effect of w_3 on the matching. For a set of parameters where w_3 is relatively small with respect to the other terms, one would expect $T3(2)$ to match $T4(2)$. However, if we increase the weight of d_θ in the total distance, the matching of these two branches becomes too expensive as it requires the rotation of branch $T3(2)$ by π. In this case, the method would match both branches to 0.

Finally, let us take T_1 and T_2 from Figure 6.4 to explain step by step how the distance between both trees would be computed. The algorithm works recursively, starting from level 0 until it reaches the last level. In this particular example, the last level is 2. Let us start at level 0, which is the level of the main axons. There are 6 possible combinations at this level (6 possible M) for the following level's branches, which are: [(1,1),(2,2)], [(1,2),(2,0)],[(1,0),(2,1)], [(1,0),(2,0)], [(1,0),(2,2)] and [(1,1),(2,0)]. Let us take as an example combination [(1,1),(2,2)]. This combination implies that the first level branches 1 and 2 of each tree will be paired. Given this M, we compute the C function of each main axon and use Equation (6.10) to find the distance between the main axons (the distance will depend on C and on the parameter b used to weight its influence) (see Figure 6.5). We then take the

first set of paired level one branches (branches 1). Each branch is actually a sub tree. In particular, branch 1 of T_1 has one sub branch and branch 1 from T_2 has no sub branches. We then repeat the process recursively for these sub trees. We can now proceed to the second set of paired level one branches (branch 2). Once again we repeat the process for these sub trees. Since each sub tree has two sub branches each, we will have 6 possible combinations at this level. For each combination we repeat the process as we have done for the previous sub trees. The whole process is repeated for each of the 6 possible combinations at level zero. Thus, 6 distances will be found. The total distance between both trees will be the minimum of the 6.

6.4 Solving the Optimal Branch Matching Problem

As explained before, to calculate the distance $D(T_1, T_2)$ between two trees it is necessary to find the optimum matching M between their branches (see Equation (6.9)). Although this can easily be achieved by computing the distance for all the possible matchings for simple images, it becomes computationally unfeasible for real images. Therefore, more efficient strategies need to be considered.

A first approach is to express our branch matching problem as an assignment problem. For an extensive review of the different variants of the assignment problem and the algorithms for solving them, the reader can refer to [Pentico 2007].

The classic assignment problem can be formalized as follows: given n agents, n tasks, and knowing the cost of assigning each agent to each task, solving the assignment problem consists of finding the one-to-one matching between agent and tasks that minimizes the total cost. The classic example is, having n workers and n tasks and knowing how much time it takes each worker to complete each task, assigning each worker to a task as to minimize the total time needed to complete all tasks. The assignment problem is usually express in matrix form (see Figure 6.6) and the matrix is referred to as Cost Matrix.

Many different algorithms for solving this problem can be found in the literature, the classical one being the Hungarian Algorithm. This iterative method was first proposed by H.W. Kuhn over 50 years ago [Kuhn 1955]. It is based on the property that, given a cost matrix, we can construct a new matrix with the same optimal solution by adding or subtracting a number from all the cells of a row or column of the original matrix. The algorithm requires the cost matrix to be squared, and it presents a $O(n^4)$ complexity.

Although this method can be used to solve the branch matching problem in algorithms such as [Basu 2011], it is not directly applicable to our matching problem since we include several constraints on the matching (no cross matching allowed,

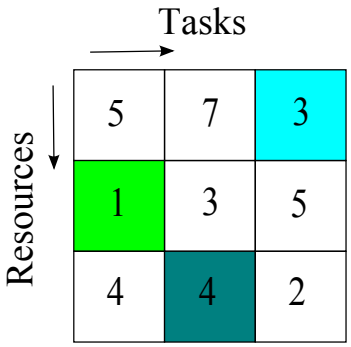


Figure 6.6: Example of an assignment problem expressed as a matrix. The optimal minimum cost solution is marked in colors.

some branches can have no match, etc).

To embed these constraints we have considered Murty's ranked assignment algorithm [Murty 1968]. Given a cost matrix, this method allows us to find the K best solutions ordered from best to worst. The best solution S_0 can be found using the Hungarian algorithm. The subsequent solutions S_i are found by solving a succession of assignment problems that are generated using S_0 and the cost matrix in a process called partitioning. We can therefore compute several ranked solutions until we find one that fulfills all the constraints on the matchings.

In particular, we have used the method proposed in [Pedersen 2008] since it presents a much better performance than the classical implementations. This work proposes a variant of the classic ranking algorithm that uses a new reoptimization technique for finding the optimal assignment for each partition. It presents a complexity of $O(Kn^3)$, where K is the number of solutions found. To use this algorithm, the first step is expressing the matching problem in matrix form. To simplify the explanation, we assume none of the trees have sub branches. However, the explanation can be extended to arbitrarily complex tree structures.

We start by assuming we have two trees T_1 and T_2 with m and n branches respectively. We construct a m by n matrix, where the component (i, j) contains the distance (cost) between branch i and j. If two branches are too far apart and we do not want to allow a matching that pairs them, we assign ∞ to that cell. Moreover, n extra auxiliary rows and m extra columns are added to the cost matrix to take into account the killing of branches. We assign the distance of killing each branch in these cells. To limit the number of equivalent combinations, we set to infinity all but one of the cells in the killing rows/columns. In addition, all cells in the section of the matrix corresponding to matches between the auxiliary rows and

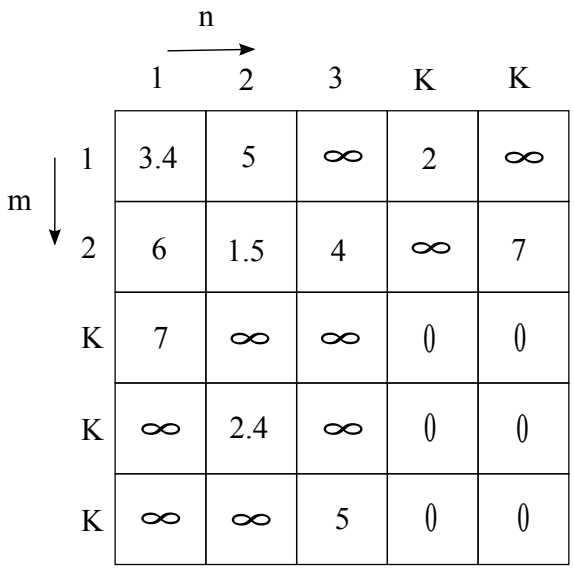


Figure 6.7: Example of an assignment problem expressed as a matrix. We assign infinity to the cell $(1, 3)$ to model the fact that both branches are far apart.

columns are set to 0. In this way we do not change the final total cost, and all rows and columns will have a match. Figure 6.7 presents an example of a cost matrix for calculating the optimum matching between two trees (for $m = 2$ and $n = 3$).

As mentioned before, the optimum solutions that do not fulfill the no-crossed-matching condition are discarded. Finally, we need to address one last problem. Since the distance between the two main curves takes into account the C functions of the two curves, it also depends on the matching of the branches (see equation (6.9)). It is not possible to include this dependency in the cost matrix. Therefore, the optimum match given by the optimization algorithm (after filtering for crossed matches) might not give the global minimum of the distance. This is specially true for a large value of b (see equation (6.7)).

To solve this problem, we compute the first valid optimum solution of the cost matrix and we find the total distance between the trees for this matching. This distance will be considered as the initial guess of the global minimum. We then compute the total distance for the following optimal solutions of the cost matrix. If the new total distance is smaller than our previous guess of the global minimum, we update it. We continue this process until we find a solution of the cost matrix which is larger that the current minimum total distance. Since the distance between the main curves can never be smaller than zero, we know that none of the remaining solutions of the cost matrix will be able to obtain a smaller total distance.

Although this method provides the global minimum, it becomes very time con-

suming when dealing with trees with many branches. We propose instead to compute an approximated solution using the Dynamic Time Warp framework (DTW) [Kruskal 1983].

This algorithm measures the difference between two signals by stretching and compressing sections of the sequences (non linear matching) in such a way as to minimize the distance between them. It it widely used in speech recognition [Sakoe 1978] and in other fields of the signal processing community. More formally, given two signals $I = [i_1, i_2, ..., i_n]$ and $J = [j_1, j_2, ..., j_m]$, a path is defined as $P = [p_1, p_2, ..p_K]$ (where $p_x = [i_q, j_r]$). Then, if we define a distance measure $d : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$, the DTW algorithm will find the optimum path P_{min}, such that the total time-normalized distance between the two signals $D(I, J)$ is minimized. This total distance is defined as:

$$D(I, J) = \frac{1}{n+m} \min_P \sum_{l=1}^{l=K} d(p_l).w_l \tag{6.15}$$

where $w_l = (i_l - i_{l-1}) + (j_l - j_{l-1})$ and $d(p_l)$ the distance between the corresponding matched i, j.

The first step of the algorithm is to construct a cost matrix. This matrix contains all the pairwise distances between the two signals (see Figure 6.9). Given the cost matrix, the algorithm finds the path that minimizes the total distance between the two signals. This path is generally referred to as the warping function.

Since the number of possible paths grows exponentially with the size of the cost matrix, the following restrictions are imposed on the warping function to make the computation more efficient:

- Boundary condition: The first and last points of both signals always match (the path starts at the top left and ends at the bottom right of the cost matrix).

- Monotonic condition: The path does not turn back on itself. This condition will be used later to assure that the optimum matching between branches will fulfill the no-crossed-matching condition.

- Continuity condition: The path does not jump in time. Both i and j can only increase by 1 on each step along the path. This condition will later be relaxed to adapt the DTW framework to our problem.

The next step is to construct an auxiliary $m \times n$ matrix g. This matrix will be used to find the optimum path. The matrix is initiated by setting $g(1, 1) = d(1, 1)$.

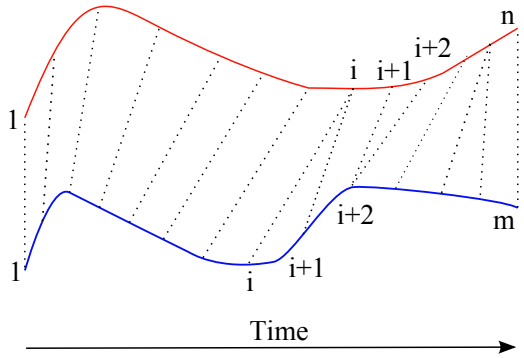


Figure 6.8: Example of the match between two signals using DTW. The matching is non-linear.

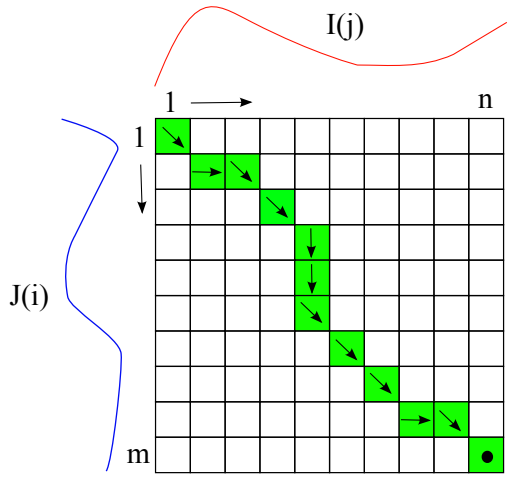


Figure 6.9: Example of a cost matrix and a possible path (green).

Then, the remaining values are calculated using the following rule:

$$g(i,j) = \min \begin{cases} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + d(i,j) \\ g(i-1,j) + d(i,j) \end{cases} \tag{6.16}$$

The Slope and Warping window conditions can be imposed at this point. Finally, the optimum path can be traced back, starting from $g(m,n)$, passing trough the points of minimum g and finishing at $g(1,1)$.

The DTW framework does not directly apply to our problem as some branches are not matched (killed). Therefore, the framework was adapted to solve the op-

timum branch-matching problem between two trees. Let us consider two trees T_1 and T_2 with m and n branches each. Once again, for the sake of simplicity, we assume they do not have sub branches and that $m \geq n$. We start by building the cost matrix. Instead of having a row per point of signal J and a column per point of signal I, like in the classical DTW framework, we first generate a $m \times n$ matrix. Each row and column will represent a branch of a tree. Since DTW forces paths to start at $(1, 1)$ and finish at (m, n), this would force us to always match the first and last branches together. To avoid this, we include two auxiliary rows and columns in the cost matrix (the first and last row/column).

To allow the algorithm to find combinations where branches are killed (assigned to zero), we alternate n auxiliary rows between the original m. A path passing through one of these cells implies that the corresponding column will be matched to zero (killed). Thus, we assign to cell (h, j) the cost of killing branch j. Moreover, cell (i, j) will contain the distance between branches i and j minus the cost of killing branch i, and the final cell $(m + n + 2, n + 2)$ will contain the total cost of killing all the m branches in T_1. In this way, we take into account the effect of killing a branch, while the algorithm is calculating the optimum path.

Finally, we need to relax the continuity constraint to avoid, for example, having multiple matches for one branch. In the classical DTW framework, paths are forced to be continuous in both i and j. When constructing the auxiliary $g(i, j)$ matrix, we take into account the values of $[g(i, j-1), g(i-1, j-1), g(i-1, j)]$. When working with branches, this would allow branches in T_2 to match multiple branches of T_1. Moreover, only neighboring branches could match (for example, branch 2 could not match branch 7). To correct this behavior we change the continuity constraint and the way of creating the g matrix as follows:

$$g(i, j) = \min_{1 \leq k < j} [g(i-1, k)] + d(i, j) \tag{6.17}$$

In this way, the path will be continuous in j (columns) but not in i (rows).

Lets us consider a concrete example. Let us assume we have two trees (T_a, T_b) with two branches each $(m = n = 2)$. The costs of matching the different branches are summarized in Table 6.1. In this simple example it is easy to see that the optimum solution is the one matching branches 1_a to 2_b and 2_a and 1_b to zero, with a total cost of 32.

Figure 6.10 shows the cost matrix associated to the matching problem along with the optimum path marked in green. The modified DTW algorithm is able to find this optimal solution.

Given that the distance between the main curves of the two trees also depends

Table 6.1: Costs of matching the different branches of T_a, T_b. The cost of assigning a branch to zero (killing) is equal for all branches.

Branch	Match	Cost
1_a	1_b	32
1_a	2_b	12
2_a	1_b	20
2_a	2_b	15
$1_a, 2_a, 1_b, 2_b$	0 (killing)	10

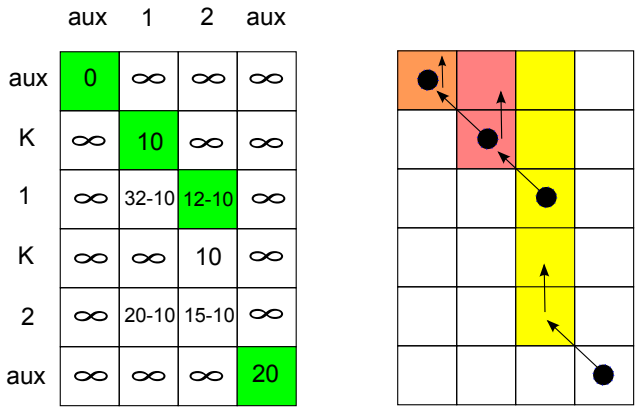


Figure 6.10: Cost matrix and optimum path (green) for the example considered in Table 6.1 (left). Panel right shows, in different colors, the search area for each column, starting from (m, n). The black circles show the location of the minimum value for each search area. The optimum path will pass through these points as shown in green in the panel left.

on the matching, the optimum solution of the cost matrix does not correspond to the global optimum solution. To improve the obtained approximation, we propose to add a term in the cost matrix that penalizes the matching of branches that are far form each other. This improves the obtained approximation. After performing tests, we determined that the difference on the average intra and inter class distance for the normal and the type 1 mutant populations is only 5%, with respect to the distances using the ranked assigned optimization method.

Using the DTW approach, the computation of the distance between two trees from the database (see Section 7.4) only takes on average a few seconds. This computational time is comparable with the one presented by the other reviewed methods.

6.5 Mean Shape

The definition and calculation of the mean tree is the first step necessary for the statistical analysis of a population. Although this is trivial when working in Euclidean spaces, it presents several difficulties in more general manifolds. Commonly, the mean m of a set of points x_i in a space S is defined as:

$$m = \arg\min_{x \in S} \sum_{i=1}^{n} d(x, x_i)^2 \tag{6.18}$$

where d is a metric in space S. Note that this definition requires a proper metric definition on the manifold.

In [Laborde 2011] the authors have found the minimizer for this equation to compute the mean of a collection of simple curves. The calculation is done using an iterated gradient approach. This methodology however is not directly applicable to our space of trees. Instead, we propose an iterative algorithm. At each iteration, the quantity defined in Equation (6.18) decreases, thus guarantying the convergence. It should be noted that we do not have unicity of the minimizer. The algorithm provides one possible solution.

Let us consider a set of trees T_i, each one with n_i number of branches. To simplify the problem, we only consider trees composed of the main axon and first level branches (no sub branches). However, the method is easily extendable to trees of arbitrary order. We start by assuming that the mean has m branches, where m is the minimum number of branches in all the trees of the set ($m = n_{min}$). We find the mean main axon using the methodology described in [Laborde 2011] on the set of all the main axons in the population. This algorithm is applicable to the main axons since they are open curves. The mean main axon will remain constant throughout the iterative process. We then initialize the iterative process with an initial estimation of the mean tree, which is composed of the main mean axon and m initial branches b_i uniformly distributed on the mean main axon. We find the optimum matches (optimum in the sense that it minimizes the distance between the two) between the initial mean tree and each tree in the set. Then, for each b_i of the initial mean tree, we use the matches found before to create an auxiliary subset B_i of branches that best match this b_i. Finally, we calculate the new mean tree by replacing each b_i by the mean curve of each B_i.

Once we have found the new mean branches, we then determine where to place them on the mean main axon. For each subset B_i, we find the mean position of the branches in it. This will give us the position of the mean branch i. We then use these mean positions to construct a mean branching function (see Figure 6.11).

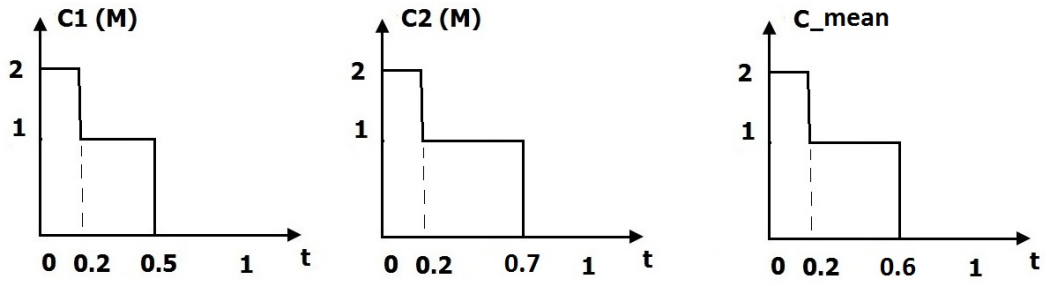


Figure 6.11: Given two branching functions C_i (left), we find the mean C function (right) which is used to position each mean branch.

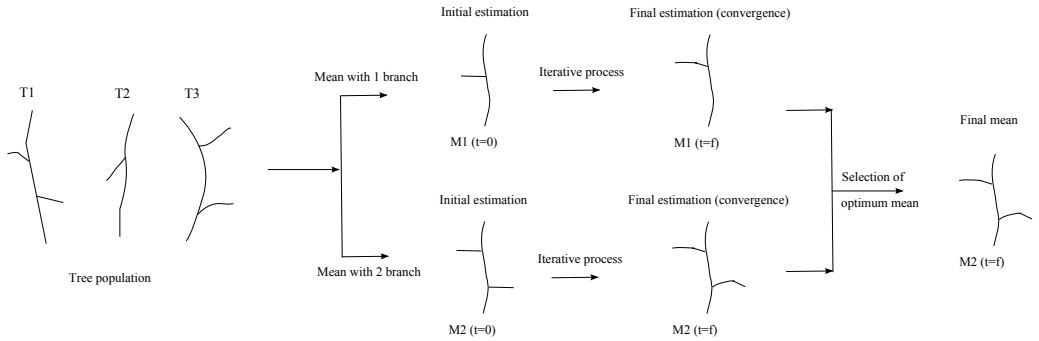


Figure 6.12: Diagram representing the iterative process set to calculate the mean of a population of three trees. Since the trees in the database have between one and two branches, we find two means (one with one branch and one with two). Finally, we chose the optimum mean as the one presenting the minimum average distance to the trees in the database.

All the processes are repeated for each iteration until the average distance between the mean tree and the population remains unchanged. The process needs to be iterative, since the optimum matching between the mean tree and a particular tree of the population can change when the shapes and positions of the mean branches change.

We repeat this process for $m = n_{min}$ to $m = n_{max}$. Thus, we find $n_{max} - n_{min} + 1$ mean trees. Finally, we define the total mean of the population as the one presenting the minimum average intra distance. The entire process is summarized for a basic example in Figure 6.12.

To illustrate how the different parameters can influence the resulting mean shape, let us consider the set of three simple synthetic images presented in Figure 6.13. All images have the same main curve and both branches of T_3 are equal (but different to the branch in T_2). The first branch of T_2 and T_3 are placed on the same point of the main curve.

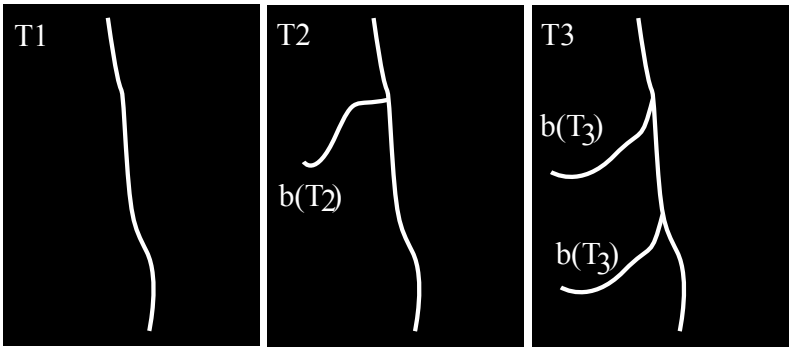


Figure 6.13: Three synthetic images, each containing one simple tree (T_1, T_2 and T_3).

The iterative algorithm will consider mean shapes with one, two and three branches, and then choose the one that presents the smallest distance between the mean and the population. Since all the images have equal main curves, the main curves of the three means will be equal to each other and to the main curve of the database. Let us start with the first mean shape (that will have zero branches). If we define $K(b)$ as the cost of killing branch b and $Transf(b_1, b_2)$ as the cost of transforming b_1 into b_2, the mean's distance to the images on the database is equal to

$$d_0 = 0 + K(b_{T_2}) + 2K(b_{T_3}) \tag{6.19}$$

The second mean shape will have one branch (referred to as b_m) and its distance to the database is equal to:

$$d_1 = K(b_m) + Transf(b_m, b_{T_2}) + Transf(b_m, b_{T_3}) + K(b_{T_3}) \tag{6.20}$$

Finally, the mean with two branches (referred to as b_m^1 and b_m^2) will have a distance defined as:

$$\begin{aligned} d_2 = K(b_m^1) + K(b_m^2) + Transf(b_m^1, b_{T_2}) + K(b_m^2) \\ + Transf(b_m^1, b_{T_3}) + Transf(b_m^2, b_{T_3}) \end{aligned} \tag{6.21}$$

To determine the final mean shape, we simply choose the one with the minimum distance to the database. However, it is clear that this will depend on the relative weight of killing with respect to transforming. In particular, if the weight of killing is very large, the mean shape will have one branch. On the other hand, the mean will have zero branches for a small killing weight.

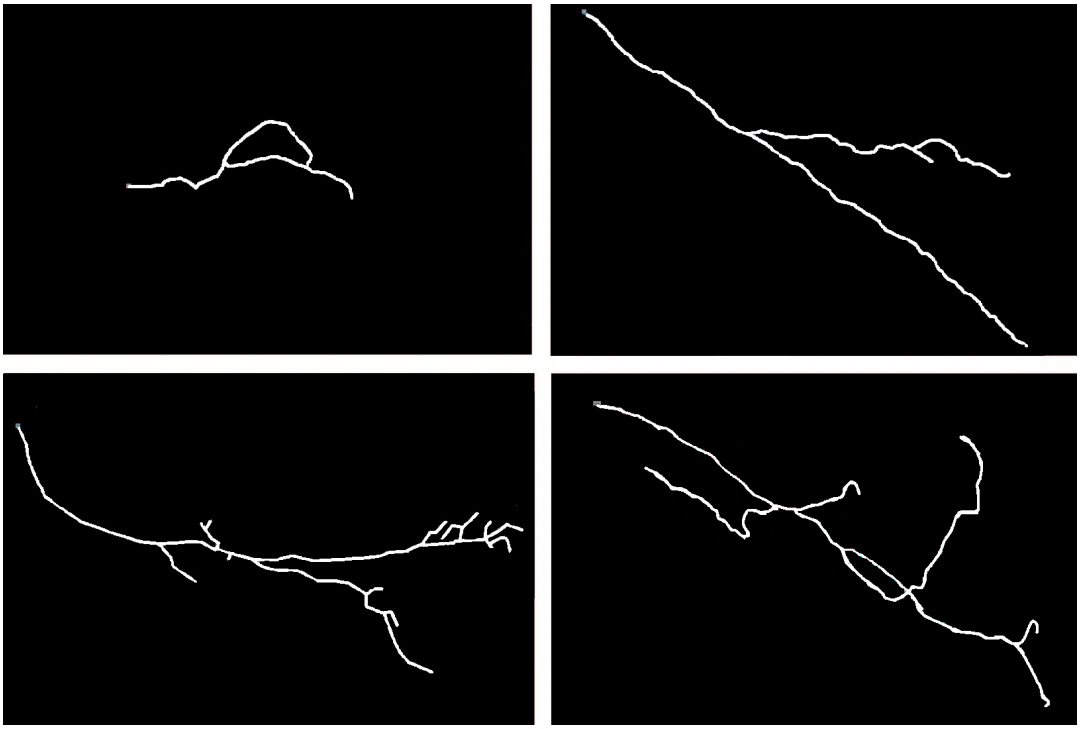


Figure 6.14: Examples of neurons from each of the four chosen populations of NeuroMorpho: Mouse granule cell (top left), Agouti type I (top right), *Drosophila* uniglomerular projection neuron (bottom left) and Mouse motoneuron (bottom right); (2D projections). The scale is different on each image to simplify the visualization.

6.6 Validation

For the validation, we have considered the two databases presented in Chapter 3.

The first set of reconstructions was taken from the NeuroMorpho.Org database [Ascoli 2007]. We have considered four different populations of neurons from the central and peripheral nervous system, and from different animals: granule cell (Mouse, Hippocampus) [Walter 2007], type I (Agouti, Neocortex) [Santiago 2010], uniglomerular projection neuron (*Drosophila*, Olfactory bulb) [Jefferis 2007] and motoneuron (Mouse, Peripheral nervous system) [Lu 2009b]. In total, we have considered 131 images (10 granule cell, 41 type I and 40 for each of the remaining two populations) (see Figure 6.14).

The second set consists on the IBV database. We have considered the full database composed of 61 images (20 normal, 24 type 1 mutant and 17 type 2 mutant). Although we could have obtained the reconstruction by segmenting the IBV images with the axon extraction method presented in Chapter 5, small segmentation errors could alter the results of the methods presented in this Chapter. Thus, we

Table 6.2: Chosen parameter values for the ESA method.

Parameter	Value
b	0.5
w_1	0.25
w_2	0.05
w_3	0.05
w_{kill} (level 1)	90
w_{kill} (level 2)	300

have decided to use the biologist's manual reconstructions instead.

We compare our results to these obtained by TED and Path2Path. In addition, we compare our method to an alternative classification scheme. We propose to start by calculating several features that are taken into account in our ESA distance. In total, eight features were used: length of the main axon, number and total length of branches and sub branches, distance (using Equation (6.6)) between the main axon and a straight line, the mean and standard deviation of the angles between the branches and their respective mother branches. We then apply PCA and only retain the principal components that account for 90% of the variance. Finally, the feature based classification is performed using the K-means algorithm.

The parameters were kept constant for both sets of experiments (see Sections 6.6.1 and 6.6.2), and were chosen manually (found by direct experimentation) to obtain the best overall accuracy in the classification. For the ESA method, the values are summarized in Table 6.2. For Path2Path, λ was set to 85. For TED, the costs of all the operations were set to 1.

6.6.1 Results: Neuromorpho Database

We have compared the performance of the three distances: RTED (an efficient solution for the tree edit distance problem) [Pawlik 2011], Path2Path and ESA with respect to a feature based classification scheme. We have embedded the three distances in the K-means algorithm to classify the populations (see Table 6.6). Since both TED and Path2Path are unable to compute the mean of a given population, we had to modify the classification algorithm. We start with three centroids randomly chosen among the total set of samples. We assign each sample to the cluster whose centroid is closest to it. Finally, we calculate the new centroid of each cluster to be the observation that has the minimum total intra distance (instead of calculating the cluster mean as in K-means). This algorithm allows us to quantify the relevance of the distances by blindly clustering the whole population into four relevant classes.

The sensitivity and overall accuracy were used as performance measures in the

Table 6.3: Sensitivity for the granule cell (S-G), type I (S-TI), uniglomerular projection neuron (S-U) and motoneuron (S-M) populations and overall accuracy (OA) for ESA, the feature-based classification scheme (FBC), Path2Path (P2P) and RTED.

Method	S-G	S-TI	S-U	S-M	OA
ESA	100.0%	75.6%	97.5%	80.0%	85.5%
FBC	0.0%	36.6%	92.5%	47.5%	54.2%
P2P	40.0%	58.5%	75.0%	30.0%	53.4%
RTED	0.0%	63.4%	87.5%	17.5%	51.9%

Table 6.4: Sensitivity for the granule cell (S-G), type I (S-TI), uniglomerular projection neuron (S-U) and motoneuron (S-M) populations and overall accuracy (OA) for ESA using the mean shape (ESA WM).

Method	S-G	S-TI	S-U	S-M	OA
ESA WM	100.0%	95.1%	100.0%	95.0%	96.9%

evaluation [Sokolova 2009]:

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{6.22}$$

$$\text{Overall accuracy} = \frac{\text{Total True Positive}}{\text{Total Number Images}} \tag{6.23}$$

The classification results are summarized in Table 6.3. Results show that the ESA distance outperforms the other methods considerably. Although the remaining methods obtain a similar overall accuracy, Path2Path obtains a more balanced sensitivity between the classes.

We have calculated the mean trees of each population (presented in Figure 6.15). The mean shapes of populations allow biologists to visualize the morphological differences between the classes. They give additional information (such as the shape) and provide a more convenient way of analyzing the populations than studying the average values of the different features that characterize axons.

To validate the mean computation and the relevance of the obtained shape for characterizing a population, we have used the mean shapes to classify the images. For each image in the database, we calculated its distance to each of the four means using the ESA distance. We then assign the sample to the class corresponding to the closest mean tree. The results are shown in Table 6.4. This classification method obtains the best result of all the tested methods, which suggest that the mean trees represent well the populations.

Moreover, we have implemented a 3-fold cross validation scheme to test the per-

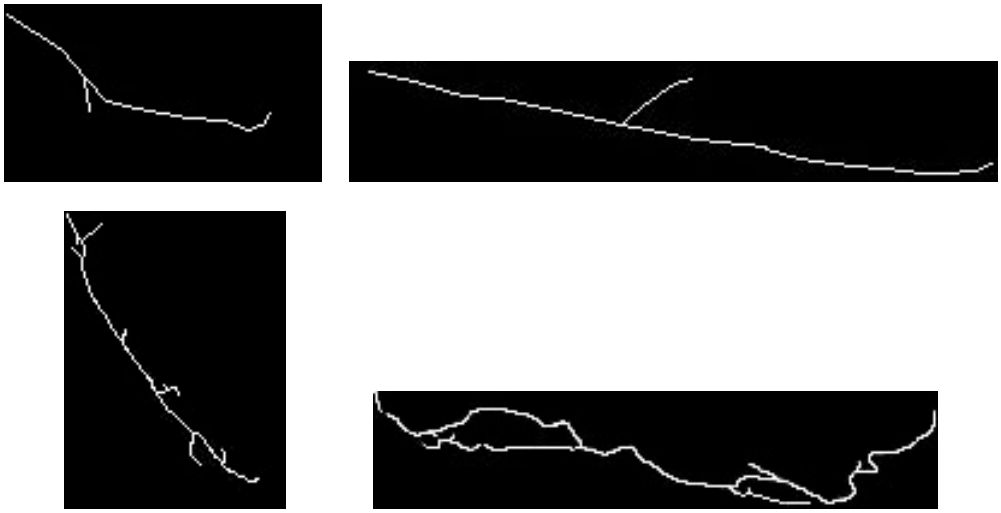


Figure 6.15: Mean shapes for the four populations taken from NeuroMorpho: granule cell (top left), type I (top right), uniglomerular projection neuron (bottom left) and motoneuron (bottom right); (2D projections). The scale is different on each image to simplify the visualization.

formance of the classification using the mean shapes and to establish the robustness of the mean computation. We have divided the dataset into three subsets of images by randomly assigning each image to one of the subsets. We then combine two subsets to form the training set, while the remaining subset is the test set. We consider the three possible assignments of subset/test-set. Moreover, we computed the mean shape of the populations on the training set. The four mean shapes were then used to classify the test sets by assigning each sample to the class corresponding to the closest mean tree, and the overall accuracies were calculated. The entire process was repeated 33 times. Therefore, we have calculated 99 overall accuracy values.

Given that we cannot compute the mean tree shape using Path2Path and TED, these methods cannot be trained. Therefore, we need to modify the previous cross validation scheme. For these two methods, the same procedure is followed, except that the computation of the mean shape of each population in the training set is replaced by the computation of the centroid of each population. The centroid is defined as the sample that presents the minimum intra distance to the other samples of the same population. The remaining steps are performed as explained before. In total, 99 overall accuracy values are computed for each method (ESA with mean, Path2Path and TED).

The mean, standard deviation, maximum and minimum values of the overall accuracy on the test sets for each method are presented in Table 6.5. Results show that the proposed methodology produced stable results, and a higher average overall

Table 6.5: Mean, standard deviation (Std), maximum (Max) and minimum (Min) of the overall accuracy on the test sets for each method (ESA with mean, P2P and TED).

Set	Mean	Std	Max	Min
ESA	95.3%	2.8%	100.0%	84.6%
P2P	39.8%	9.2%	61.0%	23.3%
TED	22.7%	8.6%	56.6%	11.1%

accuracy than the other tested methods.

Finally, to statistically compare the performance of ESA (with mean) and the other two methods, we performed a Kruskal-Wallis one-way non parametric analysis of the overall accuracy values obtained for each method. This test allows us to test whether the samples originate from the same distribution or not. We have obtained a $p = 5.3e - 34$. Thus, we can conclude that the accuracy difference between ESA and the other methods is statistically significant.

6.6.2 Results: IBV Database

The same methodology used in Section 6.6.1 was applied for the evaluation on the normal/mutant populations.

As shown in Table 6.6, Path2Path outperforms RTED. This was expected since Path2Path includes both topological and geometrical information. Notably, we obtain the best classification accuracy using the ESA distance. This could be explained by the fact that the ESA distance captures more information about the axonal trees than the other methods. Moreover, despite its simplicity, the feature based classification scheme outperforms both Path2Path and TED. It is however, outperformed by ESA, which obtains an overall accuracy of 73.8% despite the similarity between some individuals of different classes (see Figure 3.9).

In addition, the average distance (using ESA) between the mean normal tree and the normal population is 1.32, while the average intra population distance for the same population is 1.94. The same is true for the remaining two populations, where the average distance between the mean mutant type 1 tree is 0.37 and 4.11 for mutant type 2. Both distances are smaller than the average intra population distances (1.00 and 5.67 for mutant type 1 and type 2 respectively). Moreover, the classification results using the mean shapes are presented in Table 6.7. The accuracy of the classification improves significantly when using the mean shape.

We have calculated the distances between the mean normal tree and both mutant populations and determined that they are larger than the average distance between the normal population and its mean tree (respectively, 1.34 and 5.45 for mutant

Table 6.6: Sensitivity for the normal (S-Normal) and mutant (S-Mutant) populations and overall accuracy (OA) for ESA, the feature based classification scheme (FBC), Path2Path (P2P) and RTED.

Method	S-Normal	S-Mutant T_1	S-Mutant T_2	OA
ESA	85.0%	66.7%	70.6%	73.8%
FBC	90.0%	58.3%	47.1%	65.6%
P2P	70.0%	79.2%	23.5%	60.7%
RTED	47.4%	50.0%	47.1%	48.3%

Table 6.7: Sensitivity for the normal (S-Normal) and mutant (S-Mutant) populations and overall accuracy (OA) for ESA using the mean shape (ESA WM).

Method	S-Normal	S-Mutant T_1	S-Mutant T_2	OA
ESA WM	90.0%	83.3%	70.6%	80.3%

type 1 and type 2). This shows that both mutant populations are affected, but that the mutant type 1 population is only mildly affected while the mutant type 2 population is severely affected. This concurs with the biologist's assessment of the corresponding mutations.

Finally, the mean shapes of each of the three populations are presented in Figure 6.16. One can notice that the mean normal tree is bigger and presents more branches and sub branches than the mutant type 1. On the other hand, the mutant type 2 mean has a size similar to the mutant type 1 but presents more branches and sub branches. These results concur with our visual assessment of the database.

6.7 Conclusions

In this chapter we have presented a method for the comparison of tree-like structures that embeds both geometrical and topological properties. The method is an extension of the Elastic Shape Analysis Framework, which was originally developed for simple curves in the Euclidean space. The distance between two structures is defined as the length of the geodesic between them in a space of tree-like shapes. In addition, we propose a method for the computation of the mean shape of a population of trees.

We have used our method for the comparison of axon morphology. The performance was tested on two sets of images. The first set is composed of four different populations of neurons from different animals and sections of the brain from the NeuroMorpho.org open database. in total, 131 reconstructions were considered. For the second set of images, we considered all 61 (20 normal, 24 type one mutant and

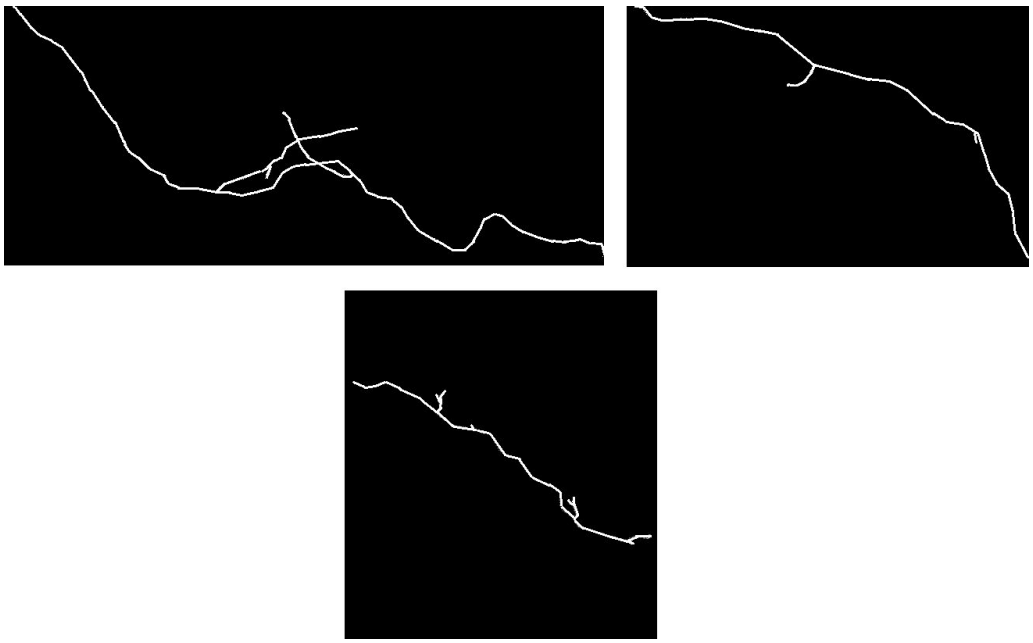


Figure 6.16: Mean normal (top left), mutant type 1 (top right) and mutant type 2 (bottom) axonal trees (2D projections). The crossing in the normal mean is due to the projection.

17 type two mutant) 3D confocal microscopy images of the IBV database, each containing one axonal tree manually segmented by an experienced biologist. Three different classes of neurons were imaged (all of the same population), each class presenting different morphological characteristics.

For the evaluation we have implemented a classical unsupervised classification scheme based on K-means. We have compared the results obtained by four methods and concluded that ours (ESA) obtains a better overall performance. Moreover, we have calculated the mean shape of each population and used them for the classification. The results obtained using the mean shapes are significantly better. The mean shapes appear to correctly characterize the populations.

Chapter 7

A Markov Chain Model for the Generation of Axonal Trees

Contents

In this chapter we propose a 2D discrete stochastic model for the simulation of axonal biogenesis. The model is defined by a third order Markov Chain. It takes into account two main processes: the growth process that models the elongation and shape of the neurites and the bifurcation process that models the generation of branches. The growth process depends, among other variables, on the external attraction field generated by a chemoattractant molecule secreted by the target area. We propose an estimation scheme of the involved parameters from a subset of the IBV image database. In total, 53 images (18 normal, 21 type 1 mutant and 14 type 2 mutant) were used. The model parameters allow us to describe pathological characteristics of the mutated populations.

7.1 Introduction

The growth of an axonal tree is determined by different biological processes. Many of the mechanisms involved in their morphogenesis are still not completely understood.

Therefore, a simulation framework for the generation of realistic axonal trees could provide an insight into these processes. Besides, if we are able to estimate the model parameters from axons obtained from images, it will provide a framework for characterizing populations.

Several models are currently proposed in the literature, which differ with respect to the biological processes being considered, the methodology and type of data used for the parameter estimation and the dimension considered (2D, 3D or 4D).

In [Segev 2000] the authors introduce a 2D model of self wiring neural networks inspired from previous work on the study of patterning of bacterial colonies. Neurons are represented as being composed of the soma, dendrites and axons. Neurites and somas communicate with each other by means of attractive or repulsive signals that determine the networks morphology.

On the other hand, [Kobayashi 2010] presents a 3D stochastic model of neuronal growth cones (tips of the growing axon) that takes into account the concentration of guidance molecules to determine their behavior. Although this model presents an interesting approach, the authors do not use the model to estimate the attraction field. Instead, it is an input necessary to simulate the behavior of the axons. Moreover, they do not consider bifurcations.

Finally, Koene et al. [Koene 2009] propose a very complete model for the generation of 4D (3D and time) neuronal networks that includes both axons and dendrites. For each neurite growth cone, the model considers the processes of elongation, turning and branching. However, no attraction field is considered. The authors estimate some of the parameters from real 4D data, and others are set manually based on properties of real neurons.

In this chapter we propose a 2D discrete stochastic model for the simulation of axonal biogenesis. The model is defined by a third order Markov Random Chain. It considers an external attraction field, and all the involved parameters (including those defining the attraction field) are estimated from real data. These estimated parameters are used to characterize populations from in vivo data.

7.2 Markov Chains

A discrete stochastic process is a set of random variables $\{X(t) \in S, t \in T\}$ with the same state space S, where index t represents time and $T = \mathbb{N}$. Thus, $X(t)$ can be interpreted as the state of the process at time t.

In addition, we introduce the Markov property (sometimes referred to as the

memorylessness property), which can by summarized by the following equation:

$$P(X_t = x_t | X_0 = x_0, ..., X_{t-1} = x_{t-1}) = P(X_t | X_{t-1} = x_{t-1}) \tag{7.1}$$

In short, the property states that the conditional probability distribution for the system at the next step depends only on the current state of the system and not on the previous states. If a discrete stochastic process is defined over a countable set S and fulfills the Markov property, the process is referred to as a Markov chain.

To fully determine a Markov chain M we need to specify the initial state X_0 (which is usually chosen according to some probability distribution) and the transition probabilities $P(i,j) = P_{i,j} = P(X_t = j | X_{t-1} = i)$ for all $i, j \in S$. This probabilities are usually estimated using different methods such as Maximum likelihood. An example of a Markov chain with its transition probabilities is presented in Figure 7.1.

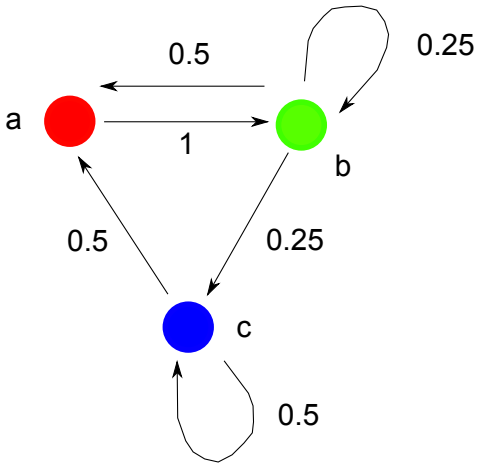


Figure 7.1: Example of a Markov chain. Note that the sum of the transition probabilities for each state is always equal to one.

If the state space S is finite, it can be convenient to express the transition probabilities in the form of a matrix P referred to as a stochastic matrix:

$$P = \begin{pmatrix} P_{0,0} & P_{0,1} & \cdots & P_{0,d} \\ P_{1,0} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ P_{d,0} & \cdots & \cdots & P_{d,d} \end{pmatrix} \tag{7.2}$$

The reader should notice that since the matrix summarizes the transition probabilities of the process, all the entries in P are non negative and the sum of each

row is equal to one.

Markov chains have been used extensively in a wide range of fields, such as Queuing theory, speech recognition and Protein modeling.

In some particular applications, it can be useful to build chains with limited memory. In these cases, a special type of chains are used. A Markov chain is said to be of order m (with $m \in \mathbb{N}$) if the process satisfies the following condition:

$$P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ..., X_0 = x_0) = \\ = P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ..., X_{t-m} = x_{t-m}) \quad (7.3)$$

This implies that, unlike the normal chains, the future state does depend on the past m states (the system has a memory). This limited dependence on the previous states can have a predictive value, which can be used to improve the results. In return, the number of parameters that need to be estimated grows exponentially with the order m. For example, for a process with n possible states and a memory length of m, the number of parameters to estimate is n^{m+1} [Saul 1998]. The probabilities involved in this type of chains can also be expressed in matricial form.

7.3 Model Description

We consider a discrete stochastic process to model the axonal tree formation that considers growth and bifurcation. The growth process models the elongation and shape of the neurites, and depends on both the internal rigidity of the neurite and on an external attraction field. On the other hand, the bifurcation process models the generation of branches. Both processes are considered to be independent from each other.

7.3.1 Growth Process

This process models the formation of the neurites on a discrete lattice $\mathbb{L} \subset \mathbb{Z}^2$.

The number of points N on a filament (i.e. its length) is determined using a Gaussian distribution function $\mathcal{N}(\mu, \sigma^2)$, where μ, σ^2 are parameters to be estimated. This choice was made after fitting different distributions to the lengths observed in our database (see Figure 7.2).

The shape of the path is determined point by point using a third order Markov Chain that depends on several parameters: E_a, E_b, E_c, E_d, E_e (referred to as elasticities) and Δ_R, Δ_T. The E_i parameters are related to the elastic properties of the neurites, while Δ_i represent an external attraction field generated by a chemo-

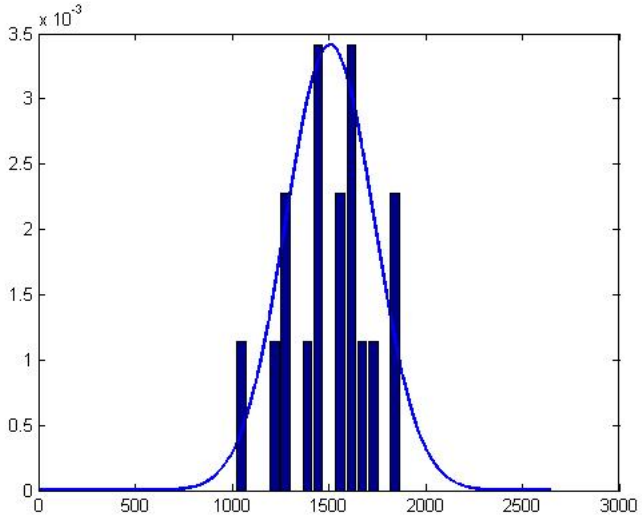


Figure 7.2: Histogram of the lengths of the main axons for the normal population, superimposed with the Gaussian that best fits the data.

attractant molecule secreted by the target area.

Given a point n_t on a path, the position of the next point on the path (n_{t+1}) depends on n_t, the two previous points n_{t-1}, n_{t-2} and on the external field in n_t ($\Delta(n_t)$). Two main cases are defined (plus all the possible rotations) depending on the configuration of n_t, n_{t-1}, n_{t-2}. For each case, the elasticity contribution is defined on Figure 7.3. The attraction field is given by the vector $\Delta(n_t) = (\Delta_R(n_t), \Delta_T(n_t))$. Δ_R is the attraction to the right and Δ_T to the top of the image (see Figure 3.6), which are assumed to be positive.

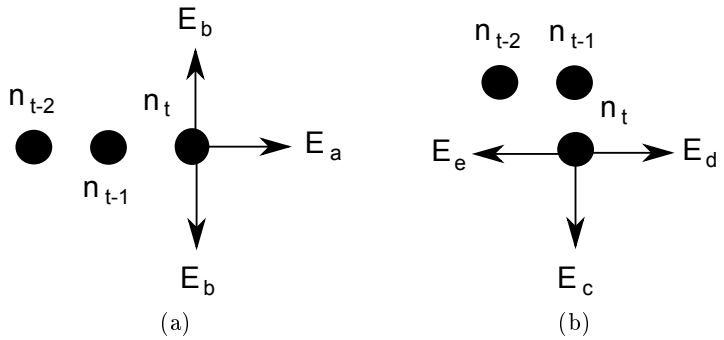


Figure 7.3: Diagram showing the two possible main configurations and the different elasticities considered in each case.

The reader should notice that the probability to go backwards is defined as zero for both configurations. This is due to the fact that the data we use to estimate the

probabilities is static and represents the mature axons. Therefore, even in the cases when the neurite does retract, we would not be able to see it and thus not be able to estimate these parameters.

To compute the conditional probabilities, we consider the sum of the elasticities and the external field for each configuration and normalize it. Let us consider the cases presented on Figure 7.3. We then define the following quantities:

$$\begin{cases} S_1 = E_a + 2E_b + \Delta_R(n_t) + \Delta_T(n_t) \\ S_2 = E_c + E_d + E_e + \Delta_R(n_t) + \Delta_T(n_t) \end{cases} \tag{7.4}$$

Then, the conditional probabilities for n_{t+1} are summarized on Figure 7.4.

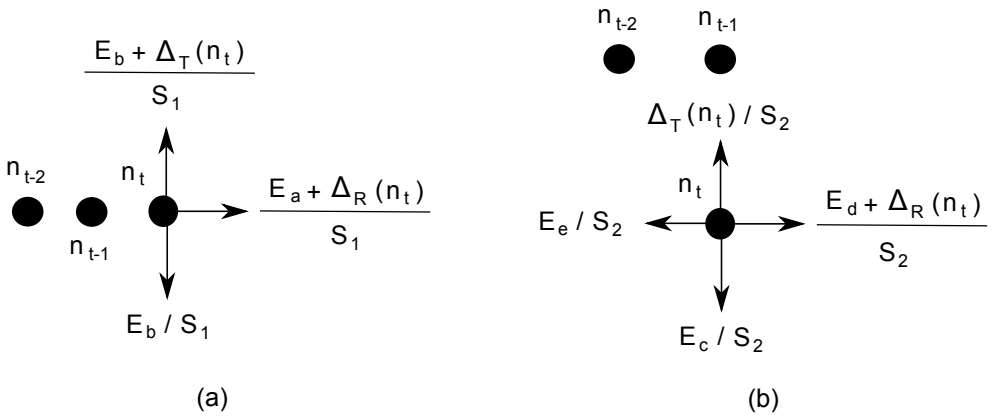


Figure 7.4: Conditional probabilities $P(n_{t+1}|n_t, n_{t-1}, n_{t-2})$ associated with the cases presented on Figure 7.3

The configurations obtained by the possible rotations of the configurations in Figure 7.3 are treated in a similar way.

7.3.2 Bifurcation Process

The bifurcation process is defined by the probability of branching P_b. For a given point n_t in a path, the probability of a branching event taking place in that point is determined by $P_b(n_t)$. We consider this probability as piecewise constant during time.

Branches grow independently of the main axon following the same model.

7.3.3 Parameter Estimation

The model parameters are given by $(P_b, \mu, \sigma^2, E_a, E_b, E_c, E_d, E_e)$ and the external field $(\Delta_x(u,v), \Delta_y(u,v))$, for $(u,v) \in L$.

Let us consider a set of axonal trees $\mathbb{A}$ (where $\mathbb{A}$ could be equal to 1), Denote by l_a, $a \in \mathbb{A}$ the length (i.e., number of pixels) of the main branch of axon a. We then have:

$$\hat{P}_b = \frac{\sum_{a \in A} \sharp_b(a)}{\sum_a l_a} \tag{7.5}$$

where $\sharp_b(a)$ is the number of branches in a.

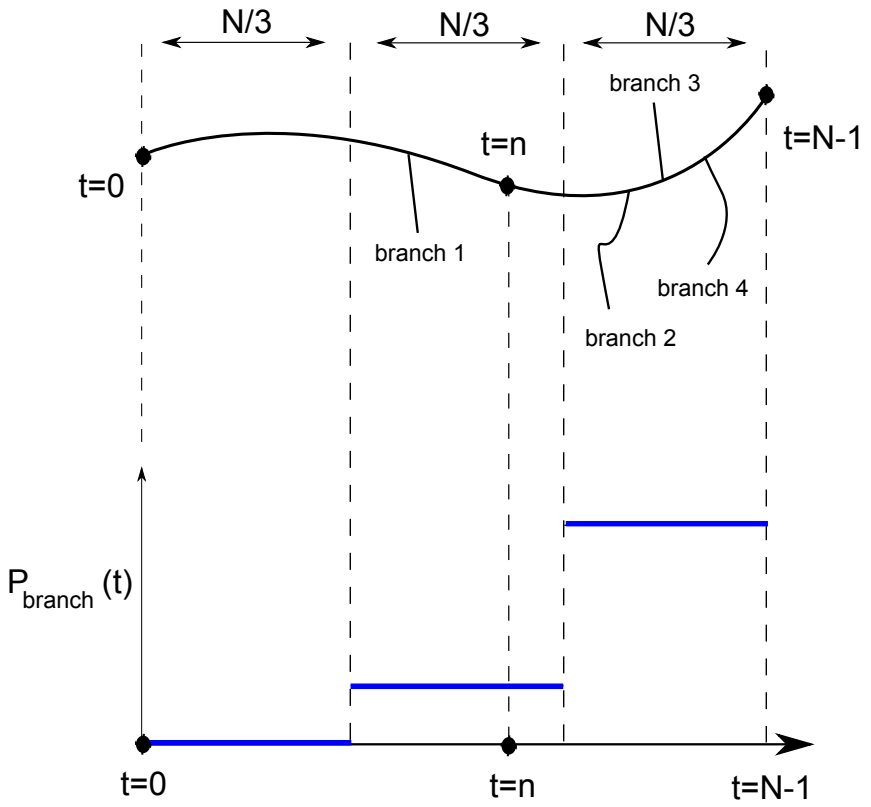


Figure 7.5: Diagram showing one axonal tree and the estimated P_b function for the main path.

Moreover, the mean and variance of the probability distribution that determines the length of a path is equal to μ and σ^2. Therefore, we have the following estimators:

$$\hat{\mu} = \frac{\sum_{a \in \mathbb{A}} l_a}{card(A)} \qquad \hat{\sigma^2} = \frac{\sum_{a \in \mathbb{A}} (l_a - \hat{\mu})^2}{card(A)} \tag{7.6}$$

For the remaining parameters, we assume that Δ is constant. As we will see in section 7.4, Δ can be approximated as being constant over subsets of L. Notice also that if we consider several axons, they all need to be registered in the same space.

There remains seven parameters. For each configuration of the past (i.e, n_{t-1}, n_{t-2})) we obtain three or four equations depending on which case we are on

(see Figure 7.4). This gives forty eight equation in total when we take into account all the possible rotations. If we consider the cases associated to configuration (a) in Figure 7.4, we obtain the following three equations:

$$\begin{cases} \frac{\sharp_1}{\sharp_{T_a}} = \frac{\hat{E_a}+\hat{\Delta_R}}{\hat{S_1}} \\ \frac{\sharp_2}{\sharp_{T_a}} = \frac{\hat{E_b}+\hat{\Delta_T}}{\hat{S_1}} \\ \frac{\sharp_3}{\sharp_{T_a}} = \frac{\hat{E_b}}{\hat{S_1}} \end{cases} \tag{7.7}$$

where $\sharp_{1,2,3}$ are the number of times that the configurations given on Figures 7.6 (a, b, c) are present along the axon, $\sharp_{T_a} = \sharp_1 + \sharp_2 + \sharp_3$ and $\hat{S_1} = \hat{E_a} + 2\hat{E_b} + \hat{\Delta_R} + \hat{\Delta_T}$.

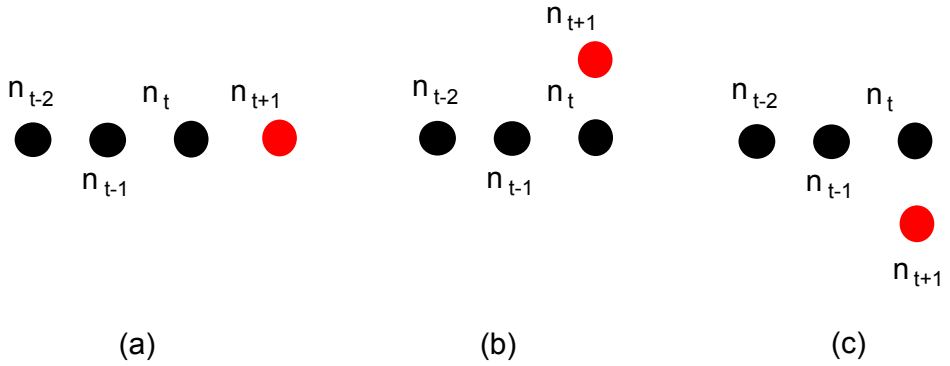


Figure 7.6: The 3 possible positions of n_{t+1} for the configuration present in Figure 7.4 (a).

Thus, we obtain the following linear equations:

$$\begin{cases} (\frac{\sharp_1}{\sharp_{T_a}} - 1).\hat{E_a} + 2.\frac{\sharp_1}{\sharp_{T_a}}.\hat{E_b} + (\frac{\sharp_1}{\sharp_{T_a}} - 1).\hat{\Delta_R} + \frac{\sharp_1}{\sharp_{T_a}}.\hat{\Delta_T} = 0 \\ \frac{\sharp_2}{\sharp_{T_a}}.\hat{E_a} + (2.\frac{\sharp_2}{\sharp_{T_a}} - 1).\hat{E_b} + \frac{\sharp_2}{\sharp_{T_a}}.\hat{\Delta_R} + (\frac{\sharp_2}{\sharp_{T_a}} - 1).\hat{\Delta_T} = 0 \\ \frac{\sharp_3}{\sharp_{T_a}}.\hat{E_a} + (2.\frac{\sharp_3}{\sharp_{T_a}} - 1).\hat{E_b} + \frac{\sharp_3}{\sharp_{T_a}}.\hat{\Delta_R} + \frac{\sharp_3}{\sharp_{T_a}}.\hat{\Delta_T} = 0 \end{cases} \tag{7.8}$$

The same reasoning is applied to configuration (b) in Figure 7.4:

$$\begin{cases} \frac{\sharp_4}{\sharp_{T_b}} = \frac{\hat{E_d}+\hat{\Delta_R}}{\hat{S_2}} \\ \frac{\sharp_5}{\sharp_{T_b}} = \frac{\hat{\Delta_T}}{\hat{S_2}} \\ \frac{\sharp_6}{\sharp_{T_b}} = \frac{\hat{E_e}}{\hat{S_2}} \\ \frac{\sharp_7}{\sharp_{T_b}} = \frac{\hat{E_c}}{\hat{S_2}} \end{cases} \tag{7.9}$$

where $\sharp_{1,2,3,4}$ are the number of times that the local configurations given on Figures 7.7 (a, b, c, d) are present on the axon, $\sharp_{T_b} = \sharp_4 + \sharp_5 + \sharp_6 + \sharp_7$ and $\hat{S_2} = \hat{E_c} + \hat{E_d} +$

$\hat{E_e} + \hat{\Delta_R} + \hat{\Delta_T}$. We then obtain the following four equations:

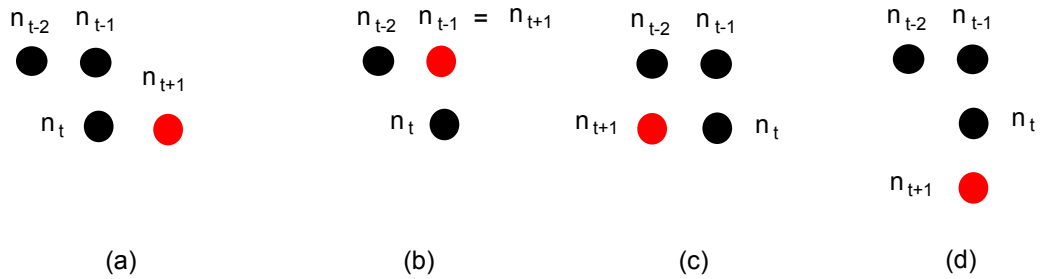


Figure 7.7: The 4 possible positions of n_{t+1} for the configuration present in Figure 7.4 (b).

$$\begin{cases} \frac{\sharp_4}{\sharp_{T_b}}\hat{E_c} + (\frac{\sharp_4}{\sharp_{T_b}} - 1)\hat{E_d} + \frac{\sharp_4}{\sharp_{T_b}}\hat{E_e} + (\frac{\sharp_4}{\sharp_{T_b}} - 1)\hat{\Delta_R} + \frac{\sharp_4}{\sharp_{T_b}}\hat{\Delta_T} = 0 \\ \frac{\sharp_5}{\sharp_{T_b}}\hat{E_c} + \frac{\sharp_5}{\sharp_{T_b}}\hat{E_d} + \frac{\sharp_5}{\sharp_{T_b}}.\hat{E_e} + \frac{\sharp_5}{\sharp_{T_b}}\hat{\Delta_R} + (\frac{\sharp_5}{\sharp_{T_b}} - 1)\hat{\Delta_T} = 0 \\ \frac{\sharp_6}{\sharp_{T_b}}\hat{E_c} + \frac{\sharp_6}{\sharp_{T_b}}\hat{E_d} + (\frac{\sharp_6}{\sharp_{T_b}} - 1)\hat{E_e} + \frac{\sharp_6}{\sharp_{T_b}}\hat{\Delta_R} + \frac{\sharp_6}{\sharp_{T_b}}\hat{\Delta_T} = 0 \\ (\frac{\sharp_7}{\sharp_{T_b}} - 1)\hat{E_c} + \frac{\sharp_7}{\sharp_{T_b}}\hat{E_d} + \frac{\sharp_7}{\sharp_{T_b}}\hat{E_e} + \frac{\sharp_7}{\sharp_{T_b}}\hat{\Delta_R} + \frac{\sharp_7}{\sharp_{T_b}}\hat{\Delta_T} = 0 \end{cases} \tag{7.10}$$

When we include all the possible rotations we obtain the total system of forty eight equations, which we solve using the Least Mean Squared (LMS) method.

In a real scenario, we would expect the values of the probabilities to change depending on the position of the considered point. This is particularly true for the attraction field, since the sensitivity to attractive gradients varies along the gradient and gets lower close to the target present at the upper edge of each image (see Section 7.4). Therefore, we calculate the values of the probabilities for each point of a path using a sliding window scheme. The optimum size of the estimation window was calculated by first using the model to simulate paths with known probabilities, and then running the estimation procedure using different sizes of windows and calculating the error between the two. We determined that a good compromise between error of the estimation and size of the window is 1200 points (see Figure 7.8).

7.3.4 Generation of an Axonal Tree

The growth process starts at an initial point (e.g. (0,0)). Since the Growth Process requires us to know the position of the 2 previous points, we assume them to be in a linear configuration (e.g. (-1,0) and (-2,0)). The first step is to calculate the number of points that the current path will by sampling the Gaussian law. Let us assume it will have N points: $[1, 2, ..., N]$.

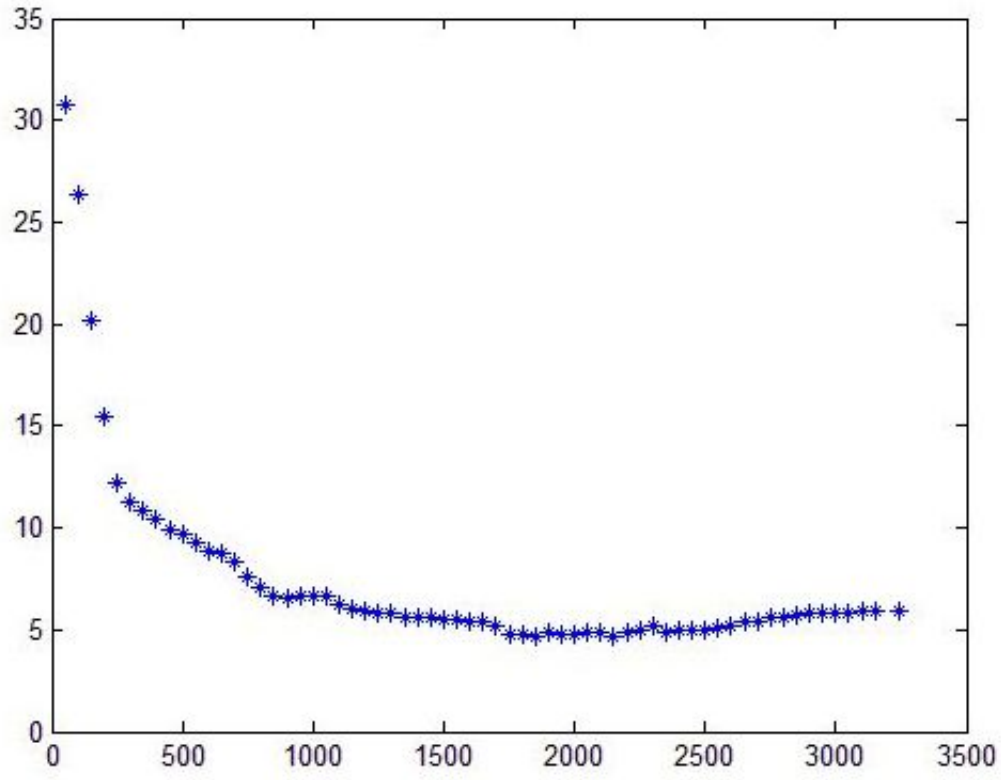


Figure 7.8: Plot of the average error between the estimated and real norm of the Δ field for different window sizes (expressed in percentage of the real norm).

The next step is to determine if this point will generate a new path (branch). For this, a random number X_1 generated from a Bernoulli distribution of parameter $P_b(1)$. If a secondary path occurs, it will be generated independently once we have finished generating the main path.

Finally, we need to determine the position of the next point using the probabilities involved in the growth process. Since the two previous points are aligned with the current one, we are currently in case (a) of Figure 7.4. A random number X_2 is generated using the corresponding probabilities and used to determine the direction of the jump.

The whole process is repeated for the N points. A possible path is shown in Figure 7.9, along with the parameters involved in each point. Once the main path is generated, we follow the same procedure to generate the secondary paths (branches) and sub branches. Each path is generated independently from the rest.

It should be noted that, even though it happens with low probability, loops can occur during the growth process. Given that in real axons loops are impossible to occur, this presents a problem that needs to be solved in the future.

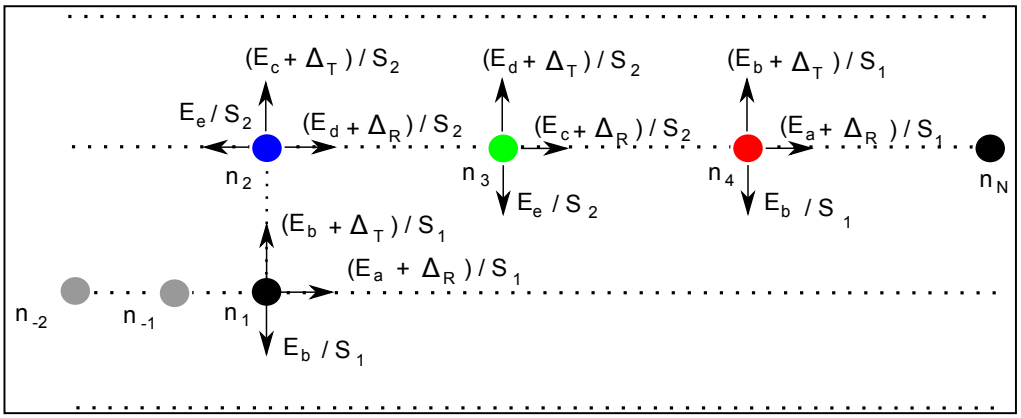


Figure 7.9: Diagram showing one possible path and the parameters involved in each point.

7.4 Validation

For the validation, we have used the IBV image database described in Chapter 3. Each image stack has two channels, and we have considered three populations of neurons (normal and two types of mutations).

In order to study the attraction field of the populations, all stacks were registered against the first image of the normal population. This was performed using the second channel of each images, and the transformations were then used to align the first channel. Images were registered by first aligning them with a rigid registration algorithm [Goshtasby 2012], followed by a non linear demons registration step [Thirion 1998].

Rigid-body transformations consist of only rotations and translations. Thus, this type of transformation preserves distances and angles between three points. In the 3D space, they have six degrees of freedom (three for the rotation and three for the translation).

On the other hand, demons registration is a particular algorithm of the non-rigid registration family. Under this approach, each image is viewed as a set of iso-intensity contours. The main idea is that a regular grid of forces (the deformation field) deforms an image by pushing the contours in the normal direction. This displacement is calculated from the following equation:

$$D. \bigtriangledown F = -(M - F) \tag{7.11}$$

where F is the fixed image, M the image to be registered and D the displacement between the two images. The previous equation is insufficient to specify D locally. Thus, the displacement field is calculated using an iterative algorithm.

Moreover, given that each pixel is able to move independently from the others, the reconstruction of the deformation field is a very ill-posed problem. To regularize the field, it is periodically convoluted with a Gaussian filter.

We have used the implementation provided by the Insight Segmentation and Registration Toolkit (ITK)(http://www.itk.org/) library to perform both types of registrations.

Due to the fact that the structure in channel 2 was on occasions not well captured in the images, we were forced to only consider a subset of the image database (see Figure 7.11). In total, 53 images (18 normal, 21 type 1 mutant and 14 type 2 mutant) were used.

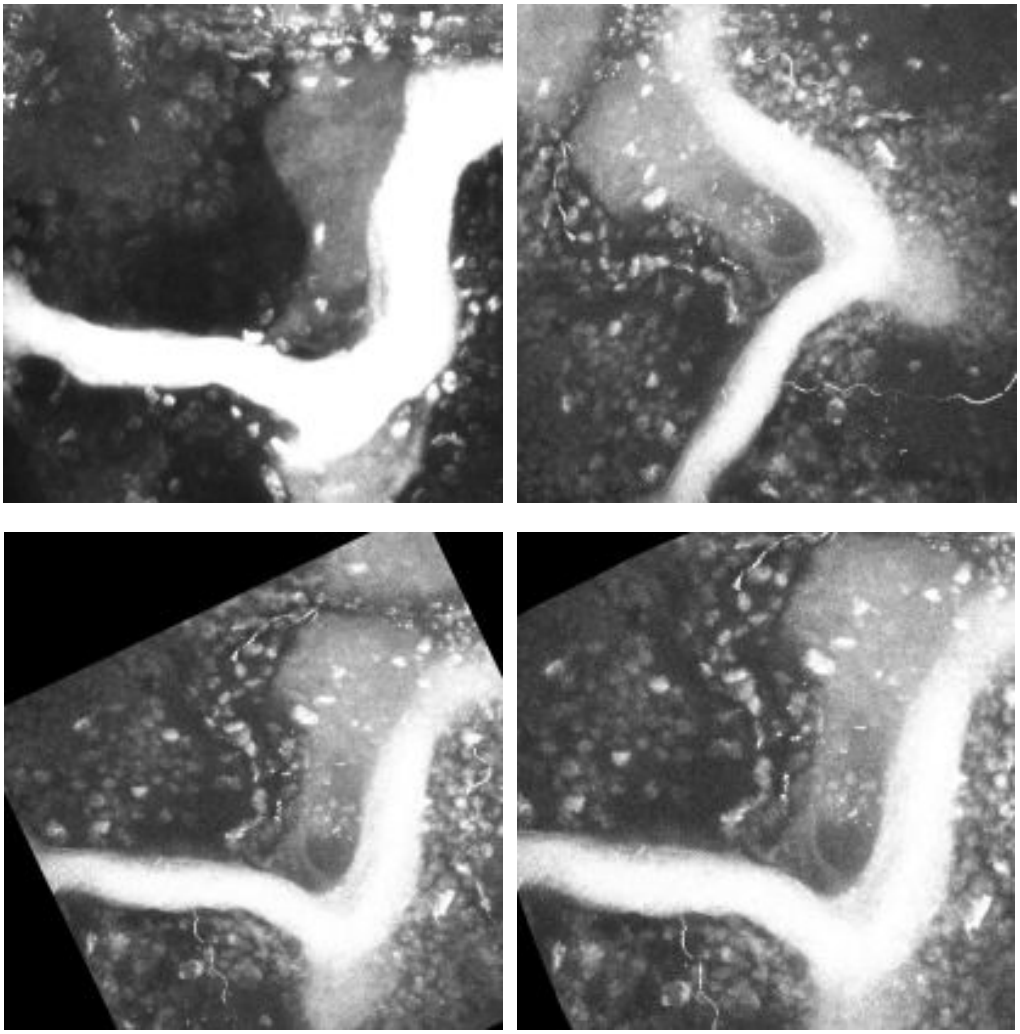


Figure 7.10: Registration result for a pair of images: reference image (top left), moving image (top right), result after rigid registration (bottom left) and after Demons registration (bottom right).

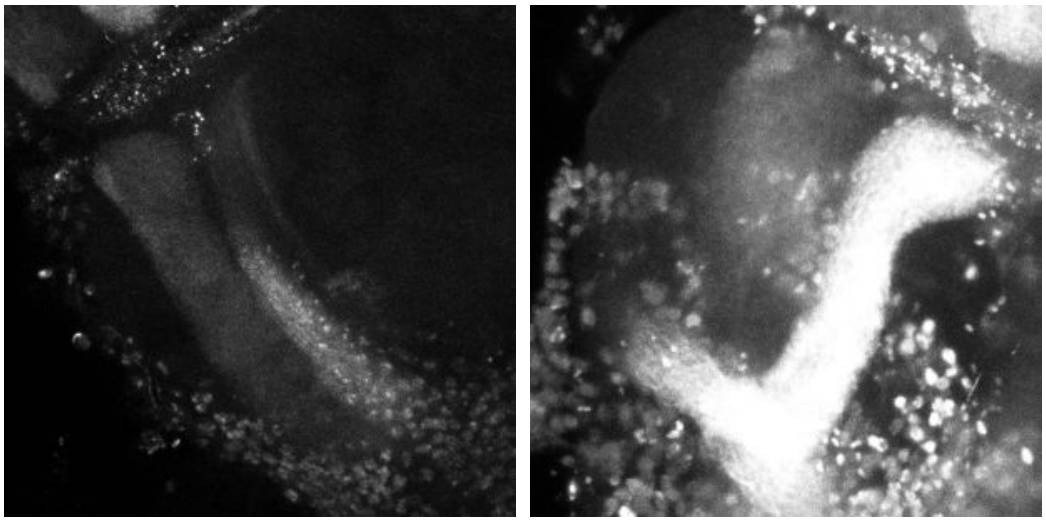


Figure 7.11: Example of an image were the structure present in channel two is not complete (left) and a valid image (right).

Given that all images were registered, we are able to use the estimated parameters to generate two fields for each population, a scalar field that represents the magnitude of E_a/E_b and a vector field that represents the attraction field Δ. Since we obtain some estimates on a sparse set of points in the x,y plane, we extrapolate the fields using a Gaussian Markov Random Field (see Section 7.4.1).

Due to the ratio between the resolution on the z axis and on the x,y plane, we have considered the maximum intensity projections of the images along the z axis.

7.4.1 Markov Random Fields

Let $X_i, i = 1, ..., n$ be a set of random variables taking values in some discrete or continuous state space S, and forming a random vector $X = (X_1, ..., X_n)$. If we assign each variable to a node of a graph G, we obtain a random field on graph G. Usually, the used graphs are regular 2D lattices on the plane. If the random field fulfills the following condition, it is said to be a Markov random field (MRF):

$$P(X(s) = x_s | X(r) = x_r, s \neq r) = P(X(s) = x_s | X(N(s)) = x_{N(s)}) \tag{7.12}$$

where $N(s)$ is the neighborhood of point s in the graph (in 2D, this could be for example, either a system of 4 or 8 neighbors) (see Figure 7.12). This equation implies that the field fulfills the markovian property [Winkler 2006]. In particular, it can be shown that Markov Chains are 1D MRF.

An important result to consider is the Hammersley-Clifford Theorem. It states

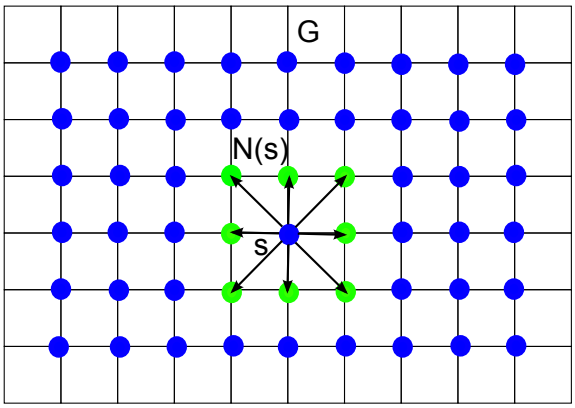


Figure 7.12: Neighboring system for a field on graph G

that X is a Markov random field with $P(X = x) > 0, \forall x$, if and only if $P(X = x)$ has the form of a Gibbs distribution:

$$P(X) = \frac{1}{Z} \exp^{\frac{-U(X)}{T}} \tag{7.13}$$

where Z is the normalizing constant for the density (known as the partition function), T is referred to as the temperature (a scale parameter for the distribution) and $U(X)$ is known as the energy function. This theorem provides a convenient way of writing the density function of a MRF. Different models are obtained depending on the formulation chosen for the energy U (for example, the Ising or the Potts models [Besag 1974, Geman 1984]).

The optimization of these types of models is quite complicated due to the existence of local optima. Thus, special algorithms need to be considered. The two most used methods to sample the model are Gibbs sampler and the Metropolis algorithm. These sampling methods are usually integrated with an annealing schedule, which consists on starting with a high temperature, and then decrease it slowly. This allows the method to find the global minimum [Winkler 2006].

MRF are widely used in image processing problems where we need to infer an original unobservable image X from an observed degraded image. Here, a MRF can be used as a prior model.

We have used these concepts to extrapolate the Δ and E_a/E_b fields. Let us consider X, Y such that:

$$\begin{cases} Y = [y_r, r \in D] \\ X = [x_r, r \in \mathbb{L}] \end{cases} \tag{7.14}$$

where D is the collection of the estimated field values on the images and $\mathbb{L}$ is the 2D lattice representing the images. Thus, Y contains the estimated data and X

is the extrapolated field we want to find. We initialize X^0 in the following way:

$$\begin{cases} x_r^0 = y_r, \forall r \ inD \\ x_r^0 = \alpha_r | \alpha_r \sim U[\min(Y), \max(Y)] \end{cases} \tag{7.15}$$

where U is a uniform distribution. We then start an iterative procedure. For each iteration i, we compute the values of X^i by sampling a Gaussian distribution:

$$\forall r \neq D, x_r^i \sim \mathcal{N}(\mu_r^i, \sigma_i^2) \tag{7.16}$$

where $\mu_r^i = \frac{1}{4} \sum_{r' \in N(r)} x_{r'}^{i-1}$ is the mean of pixel r four neighbors ($N(r)$) and $\sigma_i^2 = \frac{T_i}{2}$ is the variance (T_i being the temperature of the simulated annealing scheme that decreases at each iteration).

The iterative process continues until the maximum number of iterations has been reached.

The procedure is applied to the E_a/E_b scalar field and to the attraction field. In the case of the Δ field, the norm ρ and angle ϕ of the vector at each point are converted to x-y coordinates $(\rho\cos(\phi), \rho\sin(\phi))$ and the procedure is then applied to each component separately.

At the end of the regularization process we obtain a smooth field defined on all the points of the image (see Section 7.4.2).

7.4.2 Results

We have estimated the model parameters for each branch level of each image (main axon, branches and sub branches), and then averaged these values between the images to obtain sets of representative values for each population. We have used these values to generate several axonal trees for each population to visually compare the obtained simulations with the real images (see Figure 7.13). We determined that the real and synthetic images present a similar morphology.

Moreover, to test the parameter ability to characterize the populations, we have used them as features in a classification scheme. We then apply PCA and only retain the principal components that account for 90% of the variance. Finally, K-Means is applied to perform the unsupervised classification. We follow the same procedure with the following eight features: length of main axon, number and total length of branches and sub branches, distance (using the Elastic Shape Analysis Framework) between the main axon and a straight line and the mean and standard deviation of the angles between the branches and their respective mother branches. (quantities not taken into account by our model). We have compared both results, which are

Method	S-Normal	S-Mutant T_1	S-Mutant T_2	OA
F	76.5%	90.5%	21.4%	67.3%
MP	76.5%	57.1%	28.5%	55.8%

Table 7.1: Classification Results: Sensitivity for the normal (S-Normal) and mutant (S-Mutant T_1 and T_2) populations and overall accuracy (OA) using the model's parameters (MP) and the alternative selected features (F)

summarized in Table 7.1. The results show that the model parameters are able to characterize the neuron populations. However, this approach is outperformed by the other set of features.

In addition, we have averaged the second channel of all the registered images to obtain an approximation of the average shape of the overall neuronal structure in which axons are developing (see Figure 7.14). We have then plotted the extrapolated Δ field for each population on these three mean images.

By qualitatively analyzing the resulting images we have determined that there is no relevant difference on the field between the three populations (see Figures 7.15,7.16 and 7.17). We can appreciate that the field points towards the target area (which is located at the top right corner of the image) and that its norm is stronger at the starting point of the axons (bottom right section) and weaker near the target area.

The same procedure was used to analyze the difference on the scalar fields for each population (see Figure 7.18). In this case we can detect a significant difference between the populations. Given that E_a/E_b is related to the flexibility of the axons, we can determine that the normal axons have a more constant direction throughout its length, followed by mutant type 2 and mutant type 1 (see Figure 7.18).

7.5 Conclusions

In this chapter we have proposed a 2D discrete stochastic model for the simulation of axonal trees based on Markov Chains. The model considers several independent processes (elongation, shape and bifurcation) that are determined by several parameters, all of which can be directly estimated from real static images.

We have validated the model on a set of 53 images of the IBV database. Both normal and mutated neurons were considered (18 normal, 21 mutant type 1 and 14 type 2). For each class of neurons, we have estimated the parameters and generated synthetic axonal trees that resemble the real ones. Moreover, results show that the parameter values provide information about the axon growth properties of the populations.

This is a first approach using a simple model. In the future we intend to extend the model to 3D and to include the growth speed by considering 3D time sequences of developing axons.

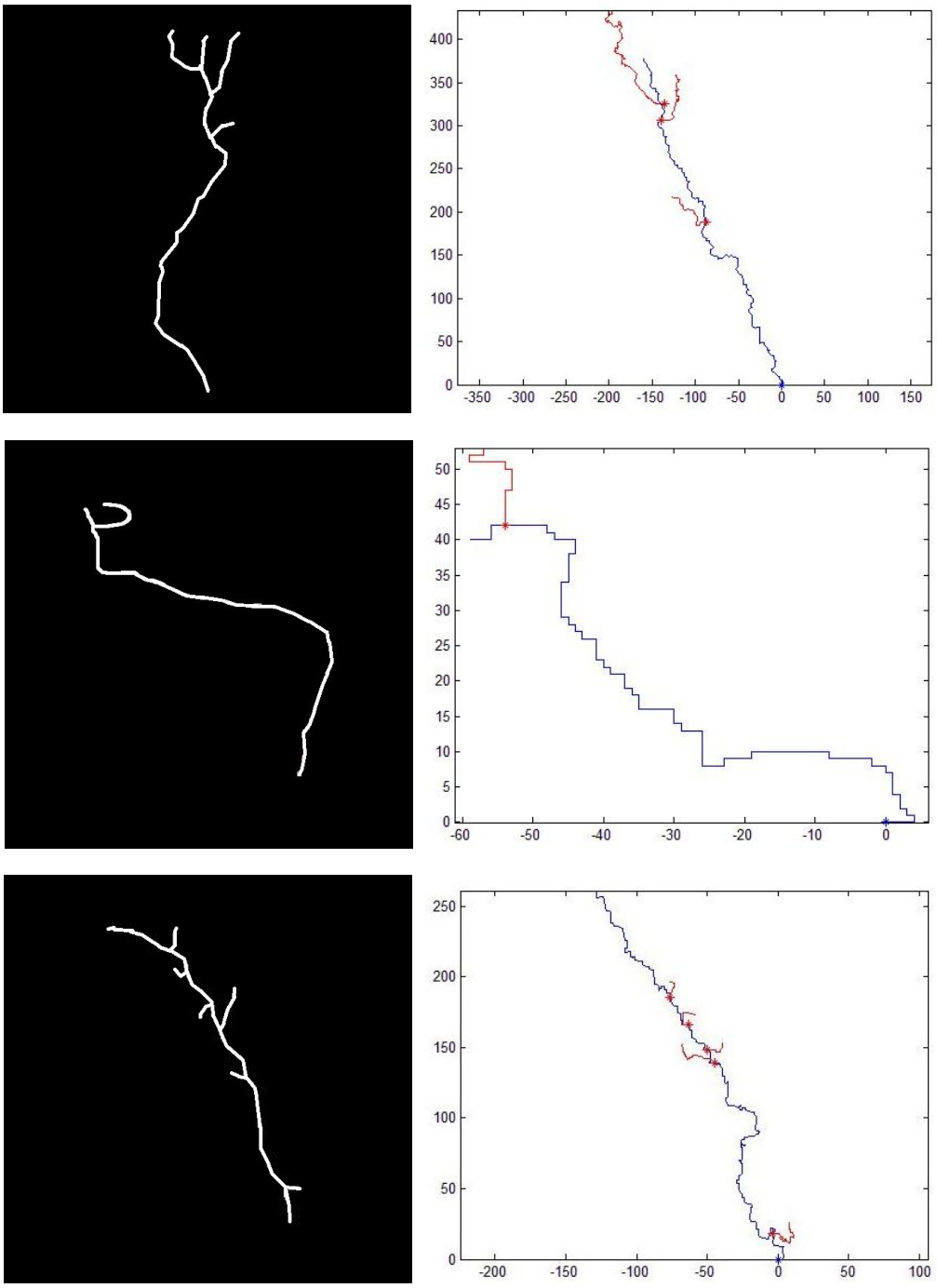


Figure 7.13: Real normal and mutant type 1 and 2 axonal trees (left top, middle and bottom respectively) and synthetic trees (right top, middle and bottom) generated using the parameters estimated from each respective image.

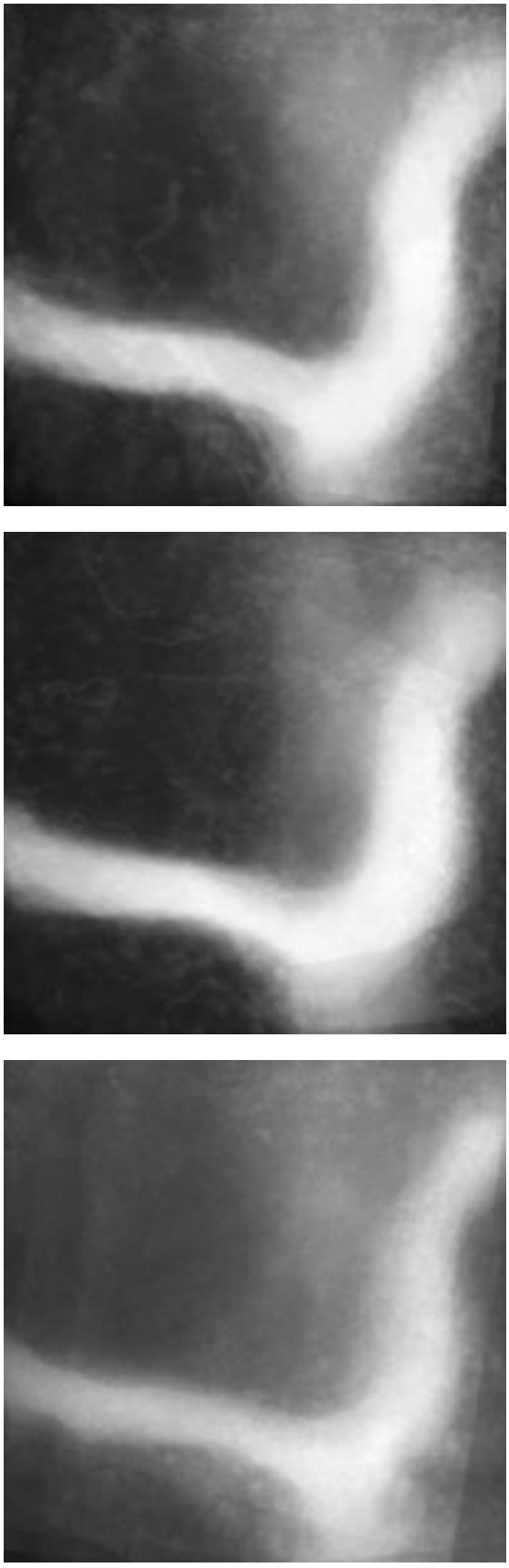


Figure 7.14: Mean second channel image (after registration) for each population: normal (top), mutant type 1 (middle) and mutant type 2 (bottom).

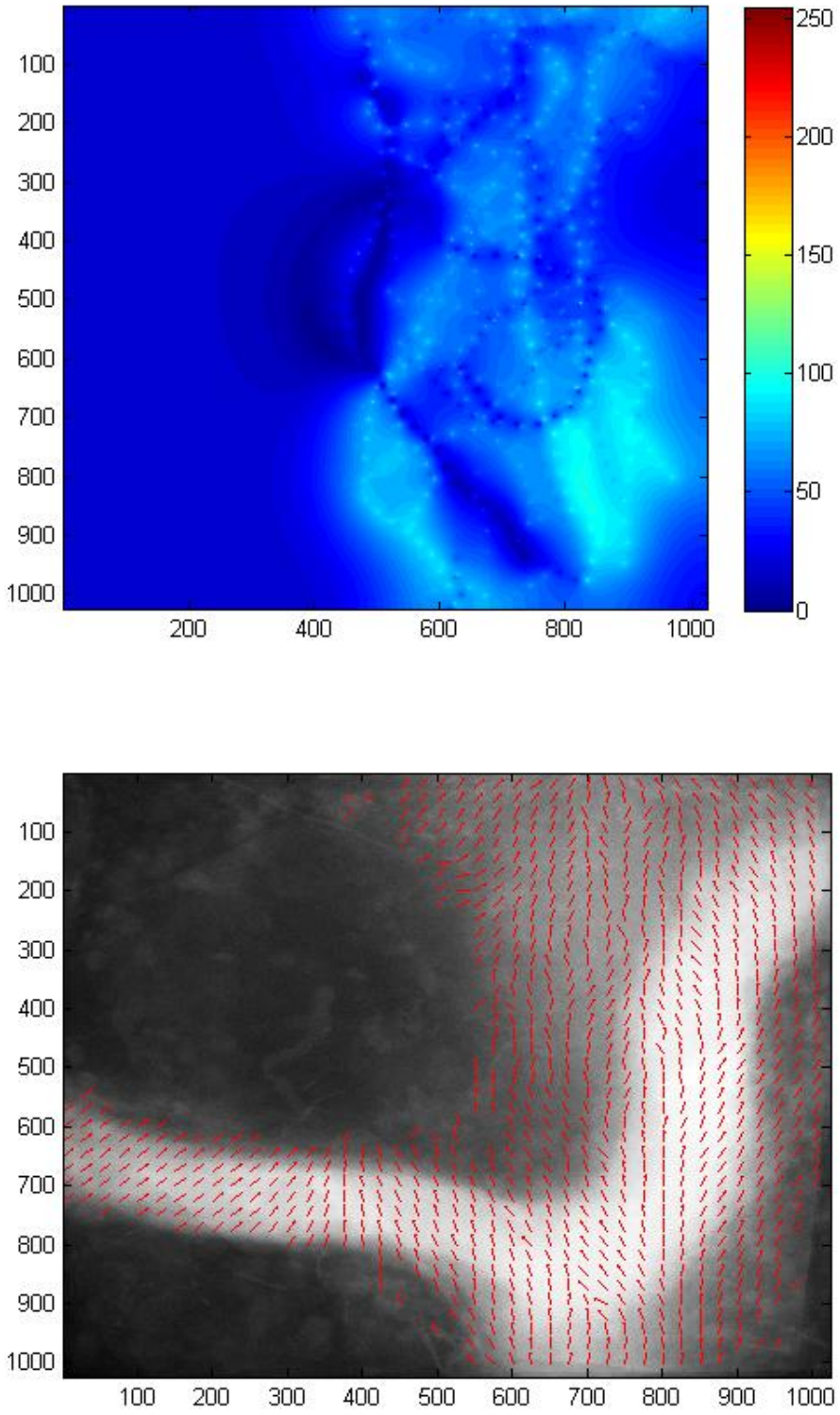


Figure 7.15: Norm (top) and direction (bottom) of the attraction field for the normal population.

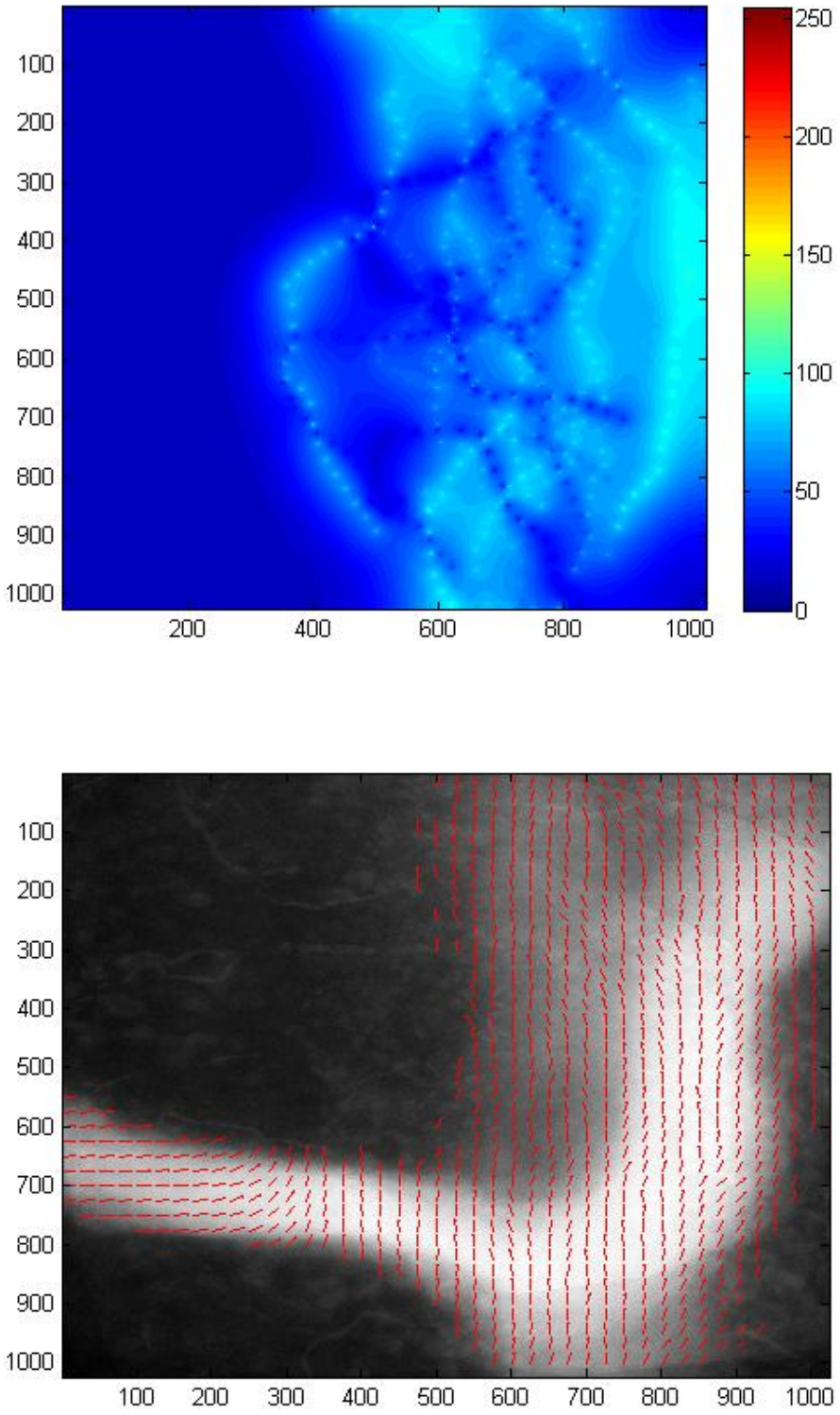


Figure 7.16: Norm (top) and direction (bottom) of the attraction field for the mutant type 1 population.

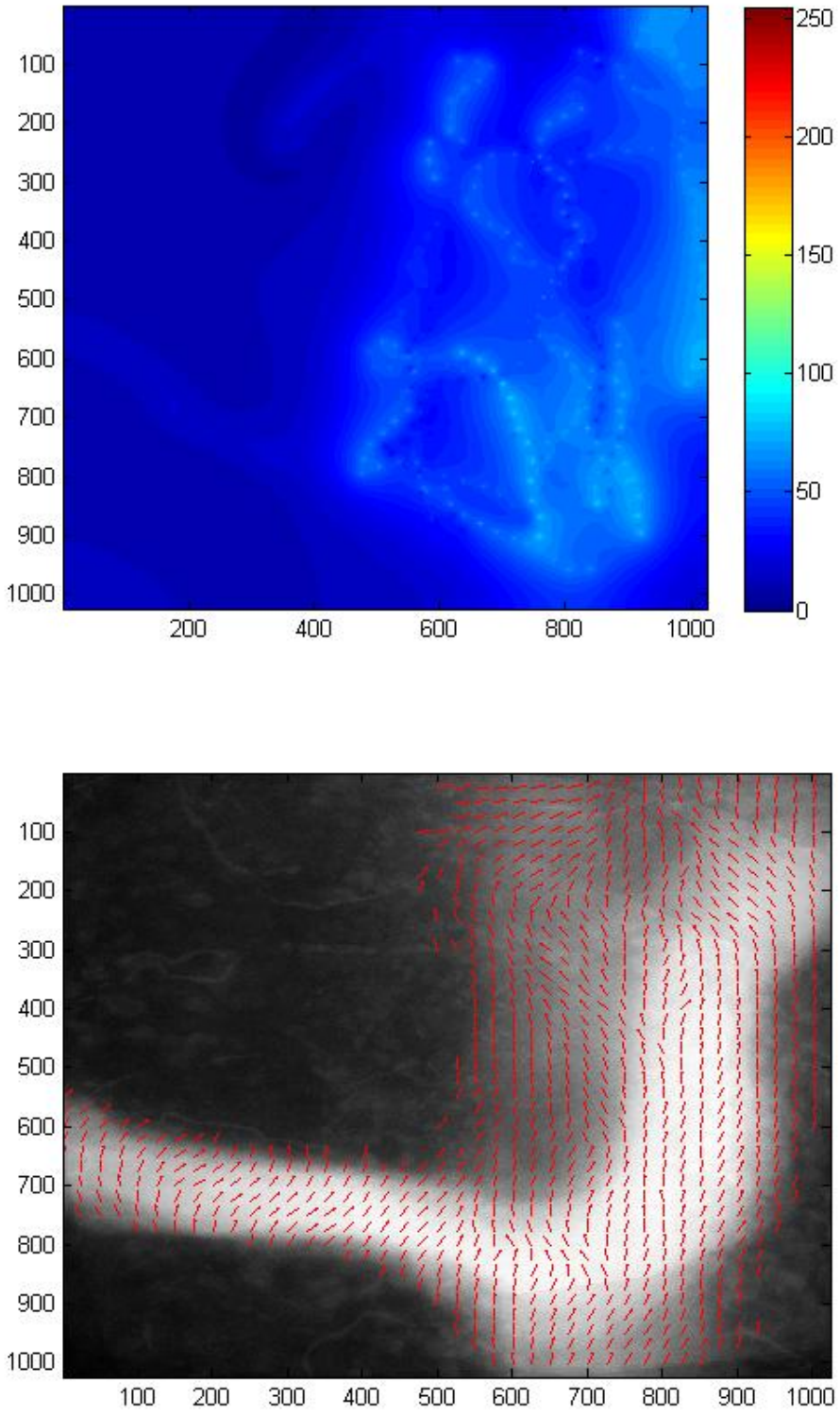


Figure 7.17: Norm (top) and direction (bottom) of the attraction field for the mutant type 2 population.

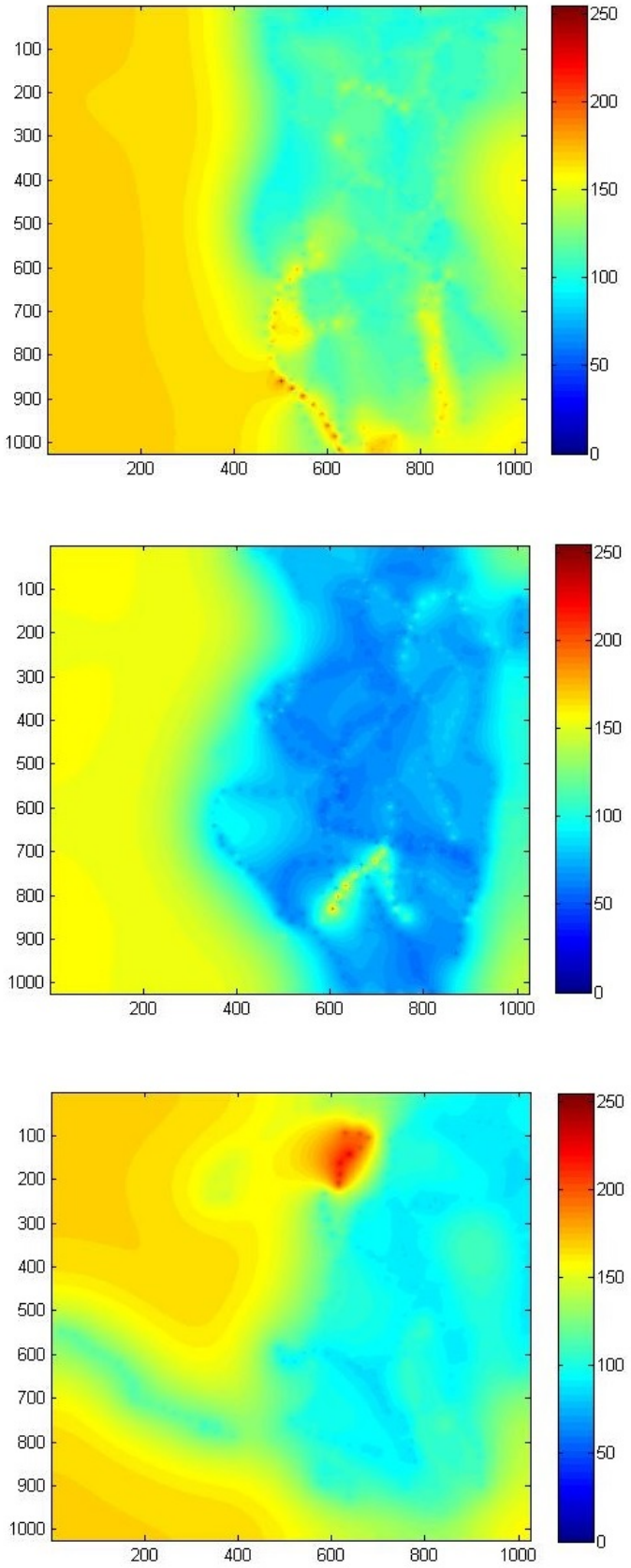


Figure 7.18: E_a/E_b scalar field for the normal (top), mutant type 1 (middle) and mutant type 2 (bottom) populations.

Chapter 8

Conclusion and Perspectives

Contents

Each chapter of this manuscript contains conclusions on the performed work. We now conclude on the main contributions of this Ph.D. Finally, we suggest possible improvements and further development related to this work.

8.1 Main contributions

In this Ph.D thesis we proposed a set of tools for the morphological analysis of axonal trees that covers the entire analysis process. All of the proposed techniques were evaluated on a large database of images of intact adult *Drosophila* fly brains. We have considered both normal and mutated neurons. The mutant neurons were obtained by inactivating certain genes related to neurological diseases in humans.

We started by proposing an automatic segmentation algorithm for confocal microscopy images. The segmentation of the images is the first step necessary to perform the analysis of the neurons. We have compared the performance of our method against two others. Results show that we are able to outperform the other segmentation algorithms is most cases.

Then, we presented a tree comparison method that considers both geometrical and topological properties. The method is an extension of the Elastic Shape Analysis Framework, which was originally developed for simple curves in the Euclidean space. In addition, we proposed a method for the computation of the mean shape of a population of trees. The average shape of a population provides biologists with a synthesized view of the morphological characteristics of a population. We used our approach to compare and classify different populations of axons, and validated it

by comparing its performance with three other techniques. We showed that our method is better at comparing and classifying the different populations of axons.

Finally, we proposed a stochastic model for the generation of axonal trees. This model is defined by a third order Markov Chain and considered several biological processes such as growth and bifurcation. All the parameters (including the chemoattractant attraction field) were estimated for the different populations from the data.

8.2 Perspectives

8.2.1 Segmentation

The results presented in Chapter 5 show that the method is able to produce good tracings that closely match the ground truth and that it outperforms two other state of the art segmentation algorithms. However, our validation was limited to only a subset of the IBV database. In the future, we intend to improve the quantitative evaluation of the method by considering the full image database and by comparing our results with additional algorithms. In addition, we need to develop a method for the automatic selection of the parameters used in each step of the segmentation process. Finally, to make this method accessible to the biological community, we should integrate it in an easy-to-use open software (for example, as a Fiji plug-in).

8.2.2 ESA Distance

First of all, we intend to improve the evaluation of the method by including more classes in the classification and by considering more sophisticated classification algorithms such as Support vector machine.

In addition, the proposed method was applied to the analysis of axonal trees, though the method is not specific to this type of structures. Thus, in the future we will apply our method to the analysis of additional structures such as airways and blood vessels.

Moreover, we need to analyze what parameters (or combination of parameters) better explain the differences between the populations. This should be done in collaboration with the biologists.

Finally, we plan to formulate statistical models of the populations by considering covariance matrices in the tangent spaces. This will allow us to better characterize the differences between the populations.

8.2.3 Stochastic Model

In this work, we modelled the growth of axonal trees in 2D. However, the real biological process occurs in 4D (3D plus time). Thus, we should first extend the model to 3D. Then, by considering 3D time sequences of developing axons, the time component could be included. This would increase the complexity of the model, and particularly, of the optimization.

Finally, additional biological processes could be included in the model, such as the growth cone variation in sensitivity with time to the chemoattractant field.

Appendix A

Introduction - Version Française

Contents

A.1 Contexte

Le cerveau est l'organe le plus complexe en termes structurels ainsi que fonctionnels. D'un point de vue microscopique, il est formé par des milliards de neurones qui s'interconnectent, formant des circuits neuronaux. Ces connections neuronales sont essentielles pour le traitement et le stockage de l'information.

Les neurones communiquent entre eux en utilisant deux types de protubérances (neurites) appelées dendrites et axones. Les dendrites reçoivent l'information qui arrive au neurone, alors que les axones sont en charge de transmettre l'information aux cellules voisines. Bien que les neurones peuvent avoir plusieurs dendrites, ils ont toujours un seul axone. Cette longue structure filamenteuse est capable de connecter un neurone avec des régions distantes du cerveau.

Durant le développement du système nerveux, les neurones étendent leurs axones et forment les circuits neuronaux nécessaires. Les axones sont guidés vers leur cible par une structure spécialisée appelée cône de croissance. Ces structures sont présentes à l'extremité des axones, et sont capables de sentir les signaux chimiques présents dans leur environnement. Le chemin suivi par les cônes de croissance dépend de ces signaux, qui peuvent être attractifs ou répulsifs.

Une fois l'axone arrive à sa destination, il commence à générer des branches pour pouvoir se connecter à plusieurs cibles. Toutes ces cibles recevront le même signal provenant du neurone. Il existe différentes configurations de ramification, chacune adaptée à différents besoins. Certains axones peuvent avoir toutes leurs branches

dans la section distale de l'axone, alors que d'autres peuvent avoir plusieurs branches qui poussent tout au long de l'axone.

Certaines caractéristiques morphologiques sont fortement liées à la capacité des neurones à fonctionner de façon correcte. Par exemple, le nombre de branches dans un axone est lié au nombre de connections qu'un neurone peut avoir. Donc, ces caractéristiques morphologiques peuvent permettre de caractériser des états pathologiques. Il est connu aussi que certaines maladies neurologiques (comme le syndrome de l'X fragile) sont liées a des caractéristiques morphologiques inusuelles dans les neurones. Par conséquent, l'analyse morphologique des neurones, et en particulier des axones, est d'une grande importance.

Pour analyser ces caractéristiques morphologiques, les chercheurs utilisent des images de neurones qui peuvent être acquis en utilisant différentes modalités. Une des techniques la plus adaptée est la microscopie confocale. Cette technique d'image offre plusieurs avantages sur la microscopie conventionnelle, et lorsqu'elle est utilisée avec le phénomène de fluorescence, elle est capable de donner des images 3D d'échantillons qu'il seraient impossible d'étudier autrement.

Pendant cette thèse nous avons travaillé en collaboration avec des biologistes de l'Institut de Biologie de Valrose. Ils nous ont fourni une base de données d'images d'axones chez la *Drosophile*. Nous avons considéré des neurones normaux et pathologiques. Les neurones pathologiques ont été obtenus en mutant certains gènes liés à différentes maladies chez l'homme. Ces images font l'objet de l'étude pendant ce thèse, et nous aideront à valider les différents algorithmes proposés.

Etant donné un ensemble d'images, nous devons les traiter pour extraire l'information présente. La première étape nécessaire pour leur traitement est la segmentation. Actuellement, il existe une grande variété de méthodes de segmentation, chacune adaptée à différentes situations. Etant donné qu'aucune de ces méthodes permetent d'obtenir des résultats satisfaisants sur nos images, nous proposons une nouvelle méthode.

Une fois l'information morphologique extraite des images, nous devons l'analyser et quantifier les différences entre les populations d'axones. Plusieurs méthodes ont été proposées. Certaines sont fondées sur une distance qui considère l'information topologique ou géométrique (ou les deux), alors que d'autres comparent directement certaines caractéristiques morphologiques (longueur, nombre de branches, etc.). Nous proposons une nouvelle méthode pour la comparaison d'arbres fondée sur la théorie des formes élastiques, que nous avons utilisée pour comparer et classifier différentes populations d'axones. De plus, nous proposons une méthode pour calculer la forme moyenne d'une population d'arbres. Cette forme moyenne fournit aux biologistes une représentation visuelle pour analyser les caractéristiques morphologiques

d'une population.

Une approche différente pour l'analyse morphologique est l'utilisation des neurones segmentés comme données d'un modèle de croissance axonale. Un tel modèle peut donner des informations sur les processus et les paramètres biologiques impliqués dans la croissance axonale. Nous proposons un nouveau modèl stochastique discret qui incorpore plusieurs processus biologiques (comme la croissance et la ramification). Tous les paramètres du modèle sont estimés à partir des images. Ces paramètres fournissent un autre outil pour caractériser et classifier les différentes populations.

La chaîne d'analyse morphologique complète est représentée sur la Figure A.1.

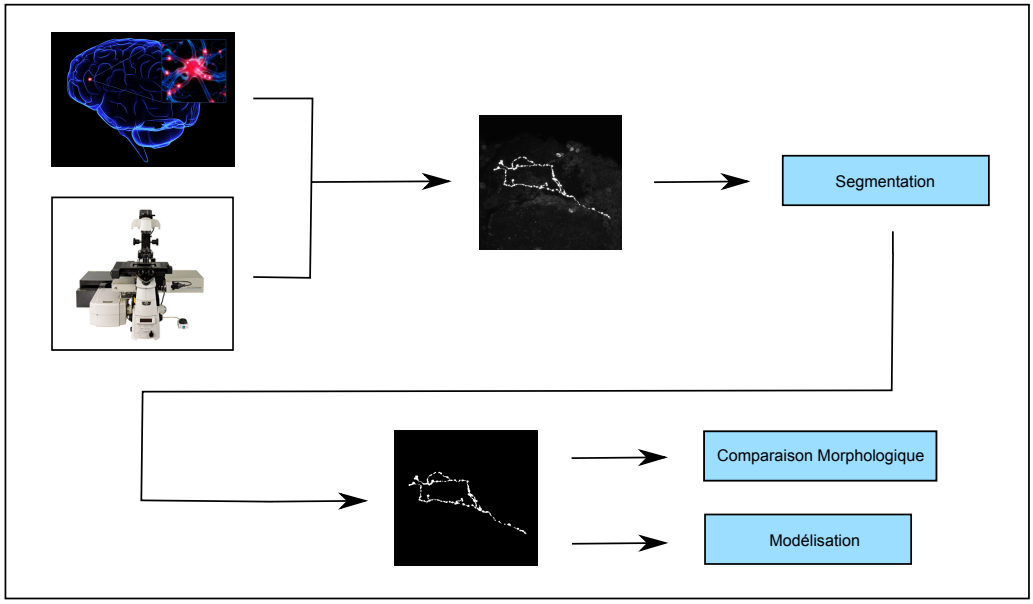


Figure A.1: Diagramme schématique de la chaîne d'analyse morphologique. (Images: http://scientopia.org/, http://www.nikoninstruments.com/ and http://www.wadsworth.org/).

A.2 Organisation de la Thèse

A.2.1 Contributions Principales

Dans cette thèse, nous décrivons quatre contributions principales:

- Un algorithme de segmentation d'images axonales en microscopie confocal. La méthode combine différents algorithmes pour rehausser les filaments, binariser, squeletiser et boucher les trous a fin d'extraire les axones des images.

- Une nouvelle distance pour comparer des arbres fondeée sur la théorie des formes élastiques. La distance considère l'information topologique et géométrique de l'arbre.
- Un algorithme pour le calcul de la forme moyenne d'une population d'arbres à partir de la distance définie.
- Un modèle stochastique discret de la formation d'arbre axonal. Ce modèle est défini par une chaîne de Markov d'ordre trois qui incorpore plusieurs processus biologiques. En particulier, un champ chimique d'attraction qui influence les cônes de croissance est considèré.

A.2.2 Organisation du Manuscrit

Ce manuscrit est organisé selon les publications sur lesquelles il est basé.

Le chapitre 2 présente les motivations de cette thèse et décrit les processus biologiques impliqués dans la morphologie axonale.

Le chapitre 3 décrit la base d'images utilisée pendant cette thèse pour évaluer les différents algorithmes proposés. Nous expliquons aussi le process d'acquisition d'images.

Dans le chapitre 4 nous continuons avec une description des différentes étapes nécessaires pour accomplir l'analyse morphologique des axones: segmentation, comparaison et modélisation. Nous décrivons brièvement les différentes méthodes de la littérature pour chaque étape. Ces méthodes seront utilisées ultérieurement comme référence pour évaluer nos algorithmes.

Le chapitre 5 détaille notre première contribution: l'algorithme de segmentation qui a été proposé dans [Mottini 2012].

Dans le chapitre 6 nous présentons les deux principales contributions de cette thèse, que nous avons proposé dans [Mottini 2013, Mottini 2014a]. Nous commençons par présenter notre méthode de comparaison d'arbres. Finalement nous présentons une méthode capable de calculer la forme moyenne d'une population d'arbres. Bien que la méthode puisse être utilisée pour la comparaison de différents types d'arbres, nous l'utilisons uniquement pour la comparaison d'arbres axonaux.

Le chapitre 7 décrit le modèle stochastique discret pour la simulation d'arbres axonaux que nous avons proposé dans [Mottini 2014b]. Il considère deux processus principaux: la croissance qui modélise l'élongation et la forme des neurites, et la ramification qui modélise la génération de branches. Le modèle est défini par une chaîne de Markov d'ordre trois. Tous les paramètres sont estimés à partir d'images de microscopie confocale. Ces paramètres peuvent être utilisés pour analyser les propriétés morphologiques des différentes populations.

Finalement, nous concluons en résumant les contributions clés développées pendant cette thèse et discutons des perspectives de ce travail dans le chapitre 8.

A.3 Liste de Publications

- **A. Mottini**, X. Descombes and F. Besse. From Curves to Trees: A Tree-like Shapes Distance Using the Elastic Shape Analysis Framework. *Neuroinformatics*, under review, 2014.
- **A. Mottini**, X. Descombes, F. Besse and E. Pechersky. Discrete Stochastic Model for the Generation of Axonal Trees. *In Proc. IEEE EMBC*, Chicago, USA, 2014.
- **A. Mottini**, X. Descombes and F. Besse. Axonal Tree Classification Using an Elastic Shape Analysis Based Distance. *In Proc. IEEE ISBI*, Beijing, China, 2014.
- **A. Mottini**, X. Descombes and F. Besse. Tree-like Shapes Distance Using the Elastic Shape Analysis Framework. *In Proc. British Machine Vision Conference*, Bristol, UK, 2013.
- **A. Mottini**, X. Descombes and F. Besse. Axon Extraction from Fluorescent Confocal Microscopy Images. *In Proc. IEEE ISBI*, Barcelona, Spain, 2012.

Appendix B

Conclusions et Perspectives - Version Française

Contents

Chaque chapitre de ce manuscrit contient une conclusion sur le travail fait. Ici, nous concluons sur les contributions principales de cette thèse. Finalement, nous suggérons de possibles améliorations et développements additionnels pour notre travail.

B.1 Contributions Principales

Dans cette thèse nous proposons un ensemble d'outils pour l'analyse morphologique d'arbres axonauxqui couvre tout le process. Toutes les techniques que nous avons proposées ont été validées sur une grande base d'images d'axones chez la *Drosophile*. Nous avons considéré des neurones normaux et mutants. Les neurones mutants ont été obtenus en modifiant certains gènes liés à différentes maladies chez l'homme.

Nous avons commencé par proposer un algorithme de segmentation automatique d'images de microscopie confocale. La segmentation d'images est la première étape nécessaire pour analyser les neurones. Nous avons comparé les performances de notre méthode avec deux autres méthodes. Les résultats montrent que notre méthode est capable de mieux segmenter nos images dans la plupart des cas.

Ensuite, nous avons présenté une méthode de comparaison d'arbres qui considère l'information géométrique et topologique. Cette méthode est une extension de la théorie des formes élastiques, qui a été développée originalement pour comparer des courbes simples dans l'espace euclidien. De plus, nous avons proposé une

méthode pour le calcul de la forme moyenne d'une population d'arbres. La forme moyenne permet aux biologistes d'avoir une représentation visuelle des principales caractéristiques morphologiques d'une population. Nous avons utilisé nos méthodes pour comparer et classifier les différentes populations d'axones, et nous avons validé nos résultats en comparant avec trois autres techniques. Nous avons montré que notre méthode est plus performante.

Finalement, nous avons proposé un modèle stochastique pour la génération d'arbres axonaux. Ce modèle est défini par une chaîne de Markov d'ordre trois, et considère plusieurs processus biologiques comme la croissance et la ramification. Tous les paramètres (y compris le champ d'attraction chimique) ont été estimés pour chaque population à partir des images.

B.2 Perspectives

B.2.1 Segmentation

Les résultats présenté dans le Chapitre 5 montrent que notre méthode produit de bons résultats comparables aux segmentations manuelles et que notre méthode fait mieux que deux autres méthodes de la littérature. Pourtant, notre validation a été limitée à un sous-ensemble de la base d'images IBV. Dans le futur, nous devons améliorer l'évaluation quantitative de la méthode en considérant la base complète. De plus, nous devons développer une méthode pour estimer les meilleurs paramètres pour chaque étape de l'algorithme. Finalement, pour rendre cette méthode accessible à la communauté biologique, elle devrait être intégrée dans un logiciel libre.

B.2.2 Distance ESA

Tout d'abord, nous devons améliorer l'évaluation de la méthode en ajoutant plusieurs classes à la classification, et en considérant des algorithmes de classification plus sophistiqués comme les séparateurs à vaste marge.

En plus, la méthode a été appliqué exclusivement à l'analyse d'arbres axonaux, même si elle peut être utilisée pour d'autres types de structures. Donc, dans le futur nous voudrions l'appliquer à l'analyse de structures additionnelles comme les vaissaux sanguins.

En outre, nous devons analyser quels paramètres (ou commbinaison de paramètres) explique mieux les différences entre les populations. Cela devrait être fait en collaboration avec les biologistes.

Finalement, nous voudrions formuler des modèles statistiques des populations en considérant les matrices de covariance dans les espaces tangentiels. Cela nous

permettrait de mieux caracteriser les différentes populations.

B.2.3 Modèle Stochastique

Dans cette thèse, nous modélisons la croissance axonale en 2D. Cependant, le process biologique se produit en 4D (3D plus temps). Donc, nous devons d'abord étendre le modèle à la 3D. Ensuite, si nous considérons des séquences 3D d'axones qui se développent dans le cerveau, nous pourrions ajouter le temps aussi. Cela augmenterait la complexité du modèle, notamment pour l'estimation des paramètres.

Finalement, d'autres processus biologiques pourraient être ajoutés au modèle, comme la variation de la sensibilité des cônes de croissance au champ d'attraction chimique.

Bibliography

[Andersson-Engels 1997] S. Andersson-Engels, C. af Klinteberg, K. Svanberg and S. Svanberg. *In vivo fluorescence imaging for tissue diagnostics.* Phys. Med. Biol, vol. 42, pages 815–824, 1997. (Cited on page 31.)

[Ascoli 2007] G.A. Ascoli, D.E. Donohue and M. Halavi. *NeuroMorpho.Org: a central resource for neuronal morphologies.* J Neurosci., vol. 27, pages 9247–9251, 2007. (Cited on pages 35 and 104.)

[Bassell 2000] G.J. Bassell and S.T. Warren. Principles of neural science, 4th ed. 2000. (Cited on page 22.)

[Basu 2010] S. Basu, A. Aksel, B. Condron and S.T. Acton. *Tree2Tree: Neuron segmentation for generation of neuronal morphology.* In Proc. International Symposium on Biomedical Imaging: From Nano to Macro, 2010. (Cited on pages 41, 43 and 61.)

[Basu 2011] S. Basu, B. Condron, and S.T. Acton. *Path2path: Hierarchical Path based Analysis for Neuron Matching.* In Proc. IEEE International Symposium on Biomedical Imaging, 2011. (Cited on pages 48, 87 and 94.)

[Besag 1974] J. Besag. *Spatial Interaction and the Statistical Analysis of Lattice Systems.* Journal of the Royal Statistical Society, Series B, vol. 36, no. 2, pages 192–236, 1974. (Cited on page 124.)

[Brown 2008] K.M. Brown, T.A. Gillette and G.A. Ascoli. *Quantifying neuronal size: summing up trees and splitting the branch difference.* Semin Cell Dev Biol., vol. 19, pages 485–493, 2008. (Cited on pages 26 and 53.)

[Bruckert 2012] H. Bruckert. Caractérisation d'hrp48, une protéine de liaison aux arns, lors de la morphogenése axonale chez la drosophile. 2012. (Cited on pages 24, 32, 33 and 34.)

[Burnette 2008] D.T. Burnette, L. Ji, A.W. Schaefer, N.A. Medeiros, G. Danuser and P. Forscher. *Myosin II activity facilitates microtubule bundling in the neuronal growth cone neck.* Dev Cell, vol. 15, pages 163–169, 2008. (Cited on page 23.)

[Campbell 2001] D.S. Campbell and C.E. Holt. *Chemotropic responses of retinal growth cones mediated by rapid local protein synthesis and degradation.* Neuron, vol. 32, pages 1013–1026, 2001. (Cited on page 23.)

[C.H. Wu 2012] N. Ticozzi P.J. Keagle P.C. Sapp K. Piotrowska P. Lowe M. Koppers D. McKenna Yasek D.M. Baron J.E. Kost P. Gonzalez Perez A.D Fox J. Adams F. Taroni C. Tiloca A.L. Leclerc S.C. Chafe D. Mangroo M.J. Moore J.A. Zitzewitz Z.S. Xu L.H. van den Berg J.D. Glass G. Siciliano E.T. Cirulli D.B. Goldstein F. Salachas V. Meininger W. Rossoll A. Ratti C. Gellera D.A. Bosco G.J. Bassell V. Silani V.E. Drory R.H. Brown C.H. Wu C. Fallini and J.E. Landers. *Mutations in the profilin 1 gene cause familial amyotrophic lateral sclerosis.* Nature, vol. 488, pages 499–503, 2012. (Cited on page 34.)

[Chabrier 2004] S. Chabrier, H. Laurent, B. Emile, C. Rosenburger and P. Marche. *A comparative study of supervised evaluation criteria for image segmentation.* 2004. (Cited on pages 73 and 74.)

[Colon-Ramos 2009] D.A. Colon-Ramos. *Synapse formation in developing neural circuits.* Curr Top Dev Biol, vol. 87, pages 53–79, 2009. (Cited on page 25.)

[Crittenden 1998] J.R. Crittenden, E.M. Skoulakis, K.A. Han, D. Kalderon and R.L. Davis. *Tripartite Mushroom Body Architecture Revealed by Antigenic Markers.* Learn Mem, vol. 5, pages 38–51, 1998. (Cited on page 32.)

[Donnelly 2011] C.J. Donnelly, D.E. Willis, M. Xu, C. Tep, C. Jiang, S. Yoo, N.C. Schanen, C.B. Kirn-Safran, J. van Minnen, A. English, S.O. Yoon, G.J. Bassell and J.L. Twiss. *Limited availability of ZBP1 restricts axonal mRNA localization and nerve regeneration capacity.* EMBO J, vol. 30, pages 4665–4677, 2011. (Cited on page 34.)

[Donohue 2011] D.E. Donohue and G.A. Ascoli. *Automated reconstruction of neuronal morphology: an overview.* Brain Res Rev, vol. 67, no. 1-2, pages 194–102, 2011. (Cited on page 42.)

[Feragen 2010a] A. Feragen, F. Lauze, P. Lo, M. de Bruijne and M. Nielsen. *Geometries on Spaces of Treelike Shapes.* In Proc. 10th Asian conference on Computer vision, vol. 2, pages 160–173, 2010. (Cited on pages 51 and 86.)

[Feragen 2010b] A. Feragen, F. Lauze and M. Nielsen. *Fundamental geodesic deformations in spaces of treelike shapes.* In Proc. International Conference on Pattern Recognition, 2010. (Cited on pages 51 and 86.)

[Feragen 2013] A. Feragen, P. Lo, M. de Bruijne, M. Nielsen and F. Lauze. *Toward a Theory of Statistical Tree-Shape Analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, pages 2008–2021, 2013. (Cited on page 51.)

[Frangi 1998] A.F. Frangi, W.J. Niessen, K.L. Vincken and M.A. Viergever. *Multiscale vessel enhancement filtering.* In Proc. Medical Image Computing and Computer Assisted Intervention, 1998. (Cited on pages 64 and 67.)

[Gallo 2011] G. Gallo. *The cytoskeletal and signaling mechanisms of axon collateral branching.* Dev Neurobiol, vol. 71, pages 201–220, 2011. (Cited on page 24.)

[Geman 1984] S. Geman and D. Geman. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 6, pages 721–741, 1984. (Cited on page 124.)

[Gillette 2011] T.A. Gillette, K.M. Brown and G.A. Ascoli. *The DIADEM Metric: Comparing Multiple Reconstructions of the Same Neuron.* Neuroinformatics, vol. 9, pages 233–245, 2011. (Cited on page 87.)

[Gobson 2011] D.A. Gobson and L. Ma. *Developmental regulation of axon branching in the vertebrate nervous system.* Development, vol. 138, pages 183–195, 2011. (Cited on page 24.)

[Goshtasby 2012] A.A. Goshtasby. Image registration: Principles, tools and methods. 2012. (Cited on page 121.)

[Grider 2006] M. Grider, Q. Chen and H. Shine. *Semi-Automated Quantification of Axonal Densities in Labeled CNS Tissue.* Journal of Neuroscience Methods, vol. 155, pages 172–179, 2006. (Cited on pages 45 and 46.)

[Grigorescu 2002] S.E. Grigorescu, N. Petkov and P. Kruizinga. *Comparison of Texture Features Based on Gabor Filters.* vol. 11, pages 1160–1167, 2002. (Cited on page 62.)

[Heisenberg 2003] M. Heisenberg. *Mushroom body memoir: from maps to models.* Nature reviews on Neuroscience, vol. 4, pages 266–275, 2003. (Cited on page 33.)

[Heumann 2009] H. Heumann and G. Wittum. *The Tree edit distance, a Measure for Quantifying Neuronal Morphology.* Neuroinformatics, vol. 7, pages 179–190, 2009. (Cited on pages 47 and 86.)

[Hirokawa 2010] N. Hirokawa, S. Niwa and Y. Tanaka. *Molecular motors in neurons: transport mechanisms and roles in brain function, development, and disease.* Neuron, vol. 68, pages 610–638, 2010. (Cited on page 22.)

[Jefferis 2007] G.S. Jefferis, C.J. Potter, A.M. Chan, E.C. Marin, T. Rohlfing, C.R. Maurer and L. Luo. *Comprehensive maps of Drosophila higher olfactory centers: spatially segregated fruit and pheromone representation.* Cell, vol. 128, pages 1187–1203, 2007. (Cited on page 104.)

[Kabaso 2009] D. Kabaso, P. Coskren, B. Henry, P. Hof and S. Wearne. *The Electrotonic Structure of Pyramidal Neurons Contributing to Prefrontal Cortical Circuits in Macaque Monkeys Is Significantly Altered in Aging.* Cerebral Cortex, vol. 19, pages 2248–2268, 2009. (Cited on page 27.)

[K.M. Brown 2011] A.J. Canty V.De Paola J.A. Hirsch G.S. Jefferis J. Lu M. Snippe I. Sugihara K.M. Brown G. Barrionuevo and G.A. Ascoli. *The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions.* Neuroinformatics, vol. 9, no. 2-3, pages 143–157, 2011. (Cited on page 41.)

[Kobayashi 2010] T. Kobayashi, K. Terajima, M. Nozumi, M. Igarashi and K. Akazawa. *A stochastic model of neuronal growth cone guidance regulated by multiple sensors.* Journal of Theoretical Biology, vol. 21, pages 712–722, 2010. (Cited on pages 55 and 112.)

[Koene 2009] R.A. Koene, B. Tijms, P. van Hees, F. Postma, A. de Ridder, J.A Ramakers, J. van Pelt and A. van Ooyen. *NETMORPH: A Framework for the Stochastic Generation of Large Scale Neuronal Networks With Realistic Neuron Morphologies.* Neuroinformatics, vol. 7, pages 195–210, 2009. (Cited on pages 56 and 112.)

[Kruskal 1983] J.B. Kruskal. *An Overview of Sequence Comparison: Time Warps, String Edits, and Macromolecules.* SIAM Rev, vol. 25, page 201âĂŞ237, 1983. (Cited on page 97.)

[Kuhn 1955] H.W. Kuhn. *The Hungarian Method for the Assignment Problem.* Naval Research Logistics Quarterly, vol. 2, page 83âĂŞ97, 1955. (Cited on page 94.)

[Laborde 2011] J. Laborde, A. Srivastava and J. Zhang. *Structure based RNA Function Prediction Using Elastic Shape Analysis.* In Proc. IEEE International Conference on Bioinformatics and Biomedicine, 2011. (Cited on page 101.)

[Lee 1994] T.C. Lee, R.L. Kashyap and C.N. Chu. *Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms.* Graphical Models and Image Processing, vol. 56, pages 462–478, 1994. (Cited on page 68.)

[Liu 2008] W. Liu, A. Srivastava, and E. Klassen. *Joint Shape and Texture Analysis of Objects Boundaries in Images Using a Riemannian Approach.* In Proc. Asilomar Conference on Signals, Systems, and Computers, 2008. (Cited on page 89.)

[Liu 2010] W. Liu, A. Srivastava, and J. Zhang. *Protein structure alignment using elastic shape analysis.* In Proc. First ACM International Conference on Bioinformatics and Computational Biology, 2010. (Cited on page 89.)

[Lowery 2009] L.A. Lowery and D. Van Vactor. *The trip of the tip: understanting the growth cone machinery.* Nat Rev Mol Cell Biol, vol. 10, pages 332–343, 2009. (Cited on page 22.)

[Lu 2009a] B. Lu, K.H. Wang and A. Nose. *Molecular mechanisms underlying neural circuit formation.* Curr Opin Neurobiol, vol. 19, pages 162–167, 2009. (Cited on page 26.)

[Lu 2009b] J. Lu, J.C. Tapia, O.L. White and J.W. Lichtman. *The interscutularis muscle connectome.* PLoS Biol., vol. 7, 2009. (Cited on page 104.)

[Medioni 2014] C. Medioni, M. Ramialison, A. Ephrussi and F. Besse. *Imp promotes axonal remodeling by regulating profilin mRNA during Drosophila brain development.* In press, Current Biology, 2014. (Cited on page 34.)

[Mottini 2012] A. Mottini, X. Descombes and F. Besse. *Axon Extraction from Fluorescent Confocal Microscopy Images.* In Proc. IEEE International Symposium on Biomedical Imaging, 2012. (Cited on pages 18 and 142.)

[Mottini 2013] A. Mottini, X. Descombes and F. Besse. *Tree like Shapes Distance Using the Elastic Shape Analysis Framework.* In Proc. British Machine Vision Conference, 2013. (Cited on pages 18 and 142.)

[Mottini 2014a] A. Mottini, X. Descombes and F. Besse. *Axonal Tree Classification Using an Elastic Shape Analysis Based Distance.* In Proc. IEEE International Symposium on Biomedical Imaging, 2014. (Cited on pages 18 and 142.)

[Mottini 2014b] A. Mottini, X. Descombes, F. Besse and E. Pechersky. *Discrete Stochastic Model for the Generation of Axonal Trees.* In Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2014. (Cited on pages 18 and 142.)

[Murty 1968] K.G. Murty. *An Algorithm for Ranking all the Assignments in Order of Increasing Cost.* Operations Research, vol. 16, pages 682–687, 1968. (Cited on page 95.)

[Otsu 1979] N. Otsu. *A threshold selection method from gray level histograms.* IEEE Trans. Sys. Man. Cyber., vol. 9, pages 62–66, 1979. (Cited on page 66.)

[Ozaki 1997] S. Ozaki and W.D. Snider. *Initial trajectories of sensory axons toward laminar targets in the developing mouse spinal cord.* J Comp Neurol, vol. 380, pages 215–229, 1997. (Cited on page 24.)

[Paddock 2014] S. Paddock, T. Fellers and M. Davidson. *Confocal Microscopy: Basic Concepts.* Rapport technique, 2014. (Cited on page 30.)

[Pawlik 2011] M. Pawlik and N. Augsten. *Rted: A Robust Algorithm for the Tree Edit Distance.* PVLDB, vol. 5, pages 334–345, 2011. (Cited on page 105.)

[Pedersen 2008] C.R. Pedersen, L.R. Nielsen and K.A. Andersen. *An Algorithm for Ranking Assignments Using Reoptimization.* Computers and Operations Research, vol. 35, page 3714âĂŞ3726, 2008. (Cited on page 95.)

[Peng 2010] H. Peng, Z. Ruan, D. Atasoy and S. Sternson. *Automatic Reconstruction of 3D Neuron Structures Using a Graph Augmented Deformable Model.* Bioinformatics, vol. 26, page i38âĂŞi46, 2010. (Cited on pages 41, 42, 60, 73 and 77.)

[Pentico 2007] D.W. Pentico. *Assignment Problems: A Golden Anniversary Survey.* European Journal of Operational Research, vol. 176, page 774âĂŞ793, 2007. (Cited on page 94.)

[Pont-Tuset 2013] J. Pont-Tuset and F. Marqués. *Measures and Meta-Measures for the Supervised Evaluation of Image Segmentation.* 2013. (Cited on page 73.)

[Risser 2008] L. Risser, F. Plouraboue and X. Descombes. *Gap filling of 3D microvascular networks by tensor voting.* IEEE Trans Med Imaging, vol. 27, pages 674–687, 2008. (Cited on pages 69 and 71.)

[Rodriguez 2003] A. Rodriguez, D. Ehlenberger, K. Kelliher, M. Einstein, S.C. Henderson, J.H. Morrison, P.R. Hof and S.L. Wearne. *Automated reconstruction of three-dimensional neuronal morphology from laser scanning microscopy images.* Methods, vol. 30, no. 1, pages 94–105, 2003. (Cited on page 44.)

[Rong 2011] J. Rong, L. Qiang, L. Quan and L. Shenquan. *A proposal for the morphological classification and nomenclature of neurons.* Neural Regeneration Research, vol. 6, pages 1925–1930, 2011. (Cited on page 53.)

[Sakoe 1978] H. Sakoe and S. Chiba. *Dynamic Programming Algorithm Optimization for Spoken Word Recognition.* IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 26, pages 43–49, 1978. (Cited on page 97.)

[Santiago 2010] L.F. Santiago, E.G. Rocha, C.L. Santos, A.Jr. Pereira, J.G. Franca and C.W. Picanco Diniz. *S1 to S2 hind and forelimb projections in the agouti somatosensory cortex: axon fragments morphological analysis.* J Chem. Neuroanat., vol. 40, pages 339–345, 2010. (Cited on page 104.)

[Saul 1998] L.K. Saul and M.I. Jordan. *Mixed memory Markov models: decomposing complex stochastic processes as mixtures of simpler ones*, 1998. (Cited on page 114.)

[Segev 2000] R. Segev and E. Ben-Jacob. *Generic modeling of chemotactic based self-wiring of neural networks.* Neural Networks, vol. 13, pages 185–199, 2000. (Cited on pages 54 and 112.)

[Selkow 1977] S. Selkow. *The tree to tree editing problem.* Information Processing Letters, vol. 6, page 184âĂŞ186, 1977. (Cited on pages 47 and 86.)

[Semwogerere 2014] D. Semwogerere and E. Weeks. *Confocal Microscopy Tutorial.* Rapport technique, 2014. (Cited on pages 29 and 30.)

[Sokolova 2009] M. Sokolova and G. Lapalme. *A Systematic Analysis of Performance Measures for Classification Tasks.* Information Processing and Management, vol. 45, page 427âĂŞ437, 2009. (Cited on page 106.)

[Srivastava 2011] A. Srivastava, E. Klassen, S.H. Joshi and I. Jermyn. *Shape Analysis of Elastic Curves in Euclidean Spaces.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, pages 1415–1428, 2011. (Cited on page 87.)

[Tai 1979] K. Tai. *The tree to tree correction problem.* Journal of the Association for Computing Machinery, vol. 26, page 422âĂŞ433, 1979. (Cited on pages 47 and 86.)

[Thirion 1998] J.P. Thirion. *Image matching as a diffusion process: an analogy with maxwellâĂŹs demons.* Medical Image Analysis, vol. 2, no. 3, pages 243–260, 1998. (Cited on page 121.)

[Uitert 2007] R. Van Uitert and I. Bitter. *Subvoxel precise skeletons of volumetric data based on fast marching methods.* vol. 34, 2007. (Cited on page 67.)

[Walter 2007] C. Walter, B.L. Murphy, R.Y. Pun, A.L. Spieles-Engemann and S.C. Danzer. *Pilocarpine induced seizures cause selective time dependent changes to adult generated hippocampal dentate granule cells.* J Neurosci., vol. 27, pages 7541–7552, 2007. (Cited on page 104.)

[Wang 2007] J. Wang, X. Zhou, J. Lichtman, S.F. Chang and S. Wong. *Dynamic Local Tracing from 3D Axon Curvilinear Structure Detection from Microscopic Image Stack.* In Proc. International Symposium on Biomedical Imaging: From Nano to Macro, 2007. (Cited on page 60.)

[Wearne 2005] S.L. Wearne, A. Rodriguez, D.B. Ehlenberger, A.B. Rocher, S.C. Hendersion and P.R. Hof. *New Techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales.* vol. 136, pages 661–680, 2005. (Cited on pages 41, 44 and 73.)

[Weiner 2013] J.A. Weiner, J.D. Jontes and R.W. Burgess. *Introduction to mechanisms of neural circuit formation.* Front Mol Neurosci, vol. 6, 2013. (Cited on page 22.)

[Wilkinson 2001] M.H. Wilkinson and M.A. Westenberg. *Shape Preserving Filament Enhancement Filtering.* In Proc. Medical Image Computing and Computer Assisted Intervention, 2001. (Cited on pages 64 and 67.)

[Wills 1999] Z. Wills, L. Marr, K. Zinn, C.S. Goodman and D. van Vactor. *Profilin and the Abl tyrosine kinase are required for motor axon outgrowth in the Drosophila embryo.* Neuron, vol. 22, pages 291–299, 1999. (Cited on page 34.)

[Winkler 2006] G. Winkler. Image analysis, random fields and markov chain monte carlo methods: A mathematical introduction (stochastic modelling and applied probability). 2006. (Cited on pages 123 and 124.)

[Zhang 1989] K. Zhang and D. Shasha. *Simple fast algorithms for the editing distance between trees and related problems.* SIAM Journal of Computing, vol. 18, pages 1245–1262, 1989. (Cited on page 86.)

[Zhang 1996] K. Zhang. *Aconstrained edit distance between unordered labeled trees.* Algorithmica, vol. 15, page 205âĂŞ222, 1996. (Cited on pages 47 and 86.)

[Zhang 2008] Y. Zhang, X. Zhou, J. Lu, J. Lichtman, D. Adjeroh and S.T. Wong. *3D Axon Structure Extraction and Analysis in Confocal Fluorescence Microscopy*

Images. Neural Computation, vol. 20, pages 1899–1927, 2008. (Cited on pages 41, 43 and 60.)