FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Advanced techniques in bioimage analysis for the study of neurodegenerative diseases – reconstruction and morphometric analysis of dendritic spines

Gustavo Caetano

Mestrado Integrado em Bioengenharia

Supervisor: Paulo Aguiar

Second Supervisor: Ana Carvalho

September 17, 2018

Resumo

As espículas dendríticas são protrusões encontradas na superfície das dendrites dos neurónios. São os locais pós-sinápticos preferenciais das sinapses excitatórias, e têm sido associadas à sua plasticidade, tendo assim fortes implicações em processos de aprendizagem, memória e neurodegeneração. Como a sua função tem sido relacionada com a sua morfologia e distribuição, a reconstrução e análise digitais têm-se consagrado promissoras no estudo destas estruturas. A maioria da investigação é, apesar disso, desenvolvida em laboratórios sem acesso a técnicas avançadas de análise de imagem. A informação imagética não é eficazmente explorada, e a sua análise torna-se laboriosa, demorada e sujeita e erros subjetivos, uma vez que os investigadores têm de depender de medições manuais para a avaliar. Nesta dissertação, são propostos três métodos computacionais avançados para a caracterização detalhada de espículas em imagens de microscopia fluorescente. O primeiro é um método geométrico capaz de detetar e caracterizar a morfologia das espículas em 3D; o segundo é um método supervisionado baseado em Deep Learning e capaz de detetar espículas em 2D; o terceiro é um conjunto de ferramentas focadas na análise da distribuição e relação das espículas com as sinapses inibitórias. Estas técnicas são inteiramente automáticas e fornecem medidas objetivas da fisicalidade das espículas. As suas metodologias são originais e mostram-se aplicáveis à deteção e estudo morfométrico das espículas, apesar de o serem com algumas precauções. Em relação às distribuições, os resultados sugerem que as espículas exibem uma correlação em número com as sinapses inibitórias em vários neurónios. Ao longo dos ramos dendríticos, os pontos de inibição também apresentam algum grau de periodicidade. Estas considerações contribuem para a caracterização da natureza das espículas, assim como da sua relação no contexto das dendrites dos neurónios.

Abstract

Dendritic spines are membranous protrusions from the surface of neuronal dendrites. They are the preferential postsynaptic sites of excitatory synapses, and have been postulated to contribute to their plasticity, thus having strong implications in processes connected with learning, memory and neurodegeneration. As their function has been associated with their morphology and distribution, digital reconstruction and analysis have become promising approaches in the study of these structures. Most of the research is, however, still performed in laboratories without the expertise for advanced image analysis. The information content on the raw data is often underexplored, and the analysis becomes laborious, time consuming and prone to error, as researchers have to rely on manual measurement to evaluate it. In this Thesis, three advanced computational tools for the detailed characterization of spines from fluorescence microscopy images are proposed. The first is a geometrical method capable of detecting and characterizing the morphology of 3D dendritic spines; the second is a supervised method based on Deep Learning and capable of detecting 2D dendritic spines; the third is a toolbox focused on the analysis of the distribution and relationship of spines and inhibitory synapses. These techniques are fully automatic and provide objective measurements of the physicality of spines. Their novel methodologies are shown to be applicable to the detection and morphometric study of spines, although with some precautions. Regarding distributions, results suggest that spines exhibit a correlation in number to the inhibitory synapses of several neurons. Along the dendritic branches, the inhibitory puncta was also found to express a degree of periodicity. These considerations help to further characterize the nature of spines, as well as their relationship within the dendrites of neurons.

Aknowledgements

I thoroughly thank all the support I have received in the course of this project, from members inside and outside the academia. Whether technical or moral, without it this project would have been faced with drastically lower odds of overcome it. A great amount of appreciation goes to my advisor, whose impact is not only limited by the pages if this dissertation. I also thank my progenitors, friends and close acquaintances, who brought the motivation to surpass the challenges offered by dendritic spines.

Gustavo Caetano

“To make two unequal parts, cut the apple in half."

Unknown

Contents

List of Figures

Abbreviations and Symbols

AMPAR	α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor
APP	Amyloid precursor protein
ASDs	Autism Spectrum Disorders
BG	Bergmann glial
EGFP	Enhanced green fluorescent protein
FXS	Fragile X Syndrome
GABA	gamma-Aminobutyric acid
GVF	Gradient Vector Flow
NMDAR	N-methyl-D-aspartate receptor
PET	Positron Emission Tomography
POI	Points of interest
PSD	Post-Synaptic Density
PSF	Point Spread Function
PSEN1	Presenilin-1
PSEN1	Presenilin-2

Chapter 1

Introduction

Neurons are the basic functional units of the nervous system [1]. They are able to receive, process and transmit electrochemical signals to coordinate all the body activities.

Communication between neurons occurs at specialized junctions named synapses, which can be either electrical or chemical. Electrical synapses physically connect two neurons by means of a gap junction, allowing ions to flow from the pre- to the postsynaptic neuron and alter its membrane potential. In contrast, in chemical synapses, the presynaptic cell releases substances – neurotransmitters - into an extracellular space called synaptic cleft, where they can bind to specific receptors of the postsynaptic cell. This, in turn, initiates ionic exchanges across its membrane and alters the postsynaptic membrane potential as well.

Depending on the ionic concentration of both cells in electrical synapses, and on the type of neurotransmitters in chemical synapses, the resting potential of the postsynaptic cell (around -70mV) can increase or decrease. Synapses which contribute to each of these events are called excitatory and inhibitory, respectively. When their balance causes the postsynaptic neuron to reach a membrane potential above a certain threshold (around -50mV), consecutive shifts in polarity are triggered along its axon. This phenomenon is called an action potential, and ends on the various axon terminals, where new synapses can be formed.

In humans, studies suggest that most synapses are chemical and occur between an axon and a dendrite [2]. Although axons release neurotransmitters only through axon terminals, dendrites can receive them directly through the dendritic shaft, or through specialized structures called dendritic spines.

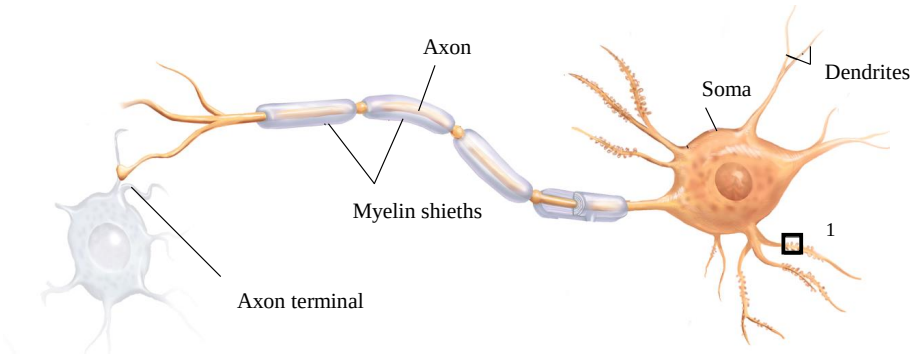
Dendritic spines are membranous protrusions from the surface of neuronal dendrites [3]. Typically, they form one excitatory synapse (more than 90% of excitatory synapses terminate on spines), although some spines may form multiple or even non-excitatory synapses [4]. These structures have been postulated to underlie brain's plastic capabilities since their discovery, implying a center role in processes such as learning and memory. Their abnormality has also been associated with numerous brain disorders, which may enable new methods of study and diagnosis of these diseases.

To support these hypothesis, a growing body of studies has been characterizing these structures under multiple facets, which are presented in the next section as their chemical composition, physical appearance, spatial distribution and how these attributes evolve over time. In addition, their biological purpose is also discussed based on their connections to normal and abnormal neurological processes.

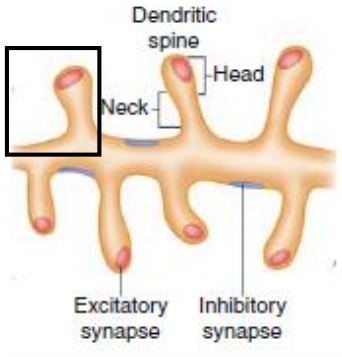
1.1 Biology of Dendritic Spines

1.1.1 Composition

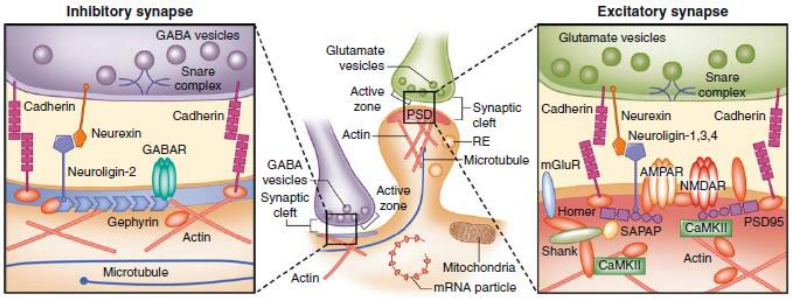
The location of dendritic spines within a neuron, as well as their composition in comparison with shaft synapse sites, is illustrated in Figure 1.1.



(a)



(b)



(c)

Figure 1.1: Representation of a spine in different levels of magnification and activity. Structure of a: a) neuron; b) dendritic segment of square 1; c) inhibitory and excitatory synapse, in the segment of square 2. Adapted from [5].
.

Spines are composed of highly specialized subdomains exerting different functions in transmission and plasticity [4]. The post-synaptic density (PSD) is one of these (Figure 1.1-c). It is a protein-dense region composed by glutamatergic receptors (such as AMPAR and NMDAR), adhesion proteins (such as neuroligin), and signaling systems involved in synaptic transmission. PSD's are not exclusive to spines, but they are more prominent in them, which gives to the excitatory synapse its characteristic asymmetrical appearance. In addition to the PSD, the spine membrane may contain specialized micro-domains for endocytosis or exocytosis. The cytoskeleton in spines is mainly formed by actin filaments (F-actin), which serve as a structural framework, a regulator of protein trafficking and, as described further below, as an active element in spine growth. Mature spines may also contain a smooth endoplasmic reticulum, intracellular membranous structures (e.g. spine apparatus, a specialized form of endoplasmic reticulum), mitochondria and protein synthesis machinery (e.g. polyribosomes), able to translate mRNA molecules from the soma into proteins according to the local needs.

1.1.2 Morphology

Spines are largely heterogeneous in both size and shape [6]. Their shape falls along a continuous spectrum, from short and thick, to long and bulbous. A nomenclature was introduced by Peters & Kaiserman-Abramof to define them, which proposed three classes:

- stubby (lacking an apparent neck);
- thin (containing a small bulbous head and a thin, long neck);
- mushroom (containing a large mushroom-shaped head);

In addition, the term filopodia was also defined for elongated dendritic protrusions that are longer and do not possess distinctive heads. These types of protrusions are illustrated in Figure 1.2.

With respect to size, spines are typically shorter than 2 μm (whereas filopodia are longer than 5 μm), and their head has between 0.001 to 1 μm^3, which typically constitutes more than 80% of their total size.

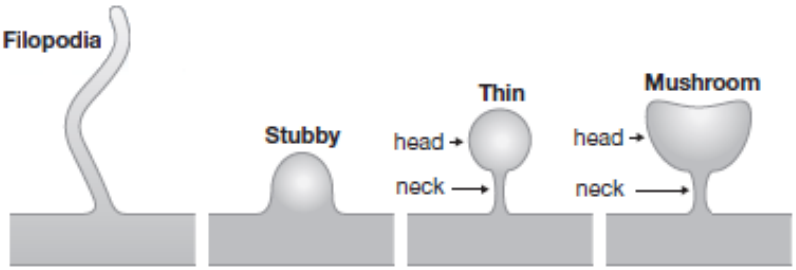


Figure 1.2: Schematic drawings of filopodia and spine morphologies based on the three-category classification.

1.1.3 Distribution

Spines are found on the dendrites of most principal neurons in the brain, including, in the cerebrum: pyramidal neurons (cortex and hippocampus), medium spiny neurons (basal ganglia), bitufted and multipolar neurons (amygdala) and neuron types I, II and III (also in basal ganglia); and in the cerebellum: Purkinje cells [7]. The distribution along the dendrites also varies. The intermediate portion of dendrites is, generally, where spines are most frequent, although there are exceptions, such as the dendrites of the hippocampal CA3 pyramidal neurons, where the proximal regions hold large, complex, branched spines called thorny excrescences. Generally, spine density varies from 1 to 10 spines per 1 μm of dendrite in mature neurons [8]. This number is dependent, as mentioned, on the position along the dendrite, the cell type (e.g. hippocampal CA1 pyramidal and granule cells usually have 2-4 per μm, while Purkinje cells around 10 per μm), and the age, which will be discussed in the next topic of this section.

1.1.4 Development

Spine maturation affects both its composition, distribution, morphology and motility. Over the first week of life, spines are rather thin and elongated and gradually gain a typical mushroom-like structure as the tissue matures [9]. As mentioned above, filopodia density decreases and the general motility becomes less noticeable, except for experience-driven synaptogenesis events. Structurally, immature spines do not contain well-developed organelles such as a spine apparatus and a smooth endoplasmic reticulum. In addition, spines contain more polyribosomes when their formation is at its peak, which suggests that new synapses require a larger number of locally synthesized proteins. Despite some differences between immature and mature spines being known, their origin and growth mechanism are still debatable [10]. There are three hypotheses for the former: the first suggests spines form from filopodia, which are transient by nature; the second that spines grow from existing shaft synapses on dendrites; and an intermediate hypothesis suggests that filopodia connects to an axon terminal and retracts to generate a shaft synapse, where a spine will later develop. Regarding their growth, a recent study [11] showed how G-actin, the

monomers of F-actin which compose the cytoskeletal, may be responsible for the phenomena. These molecules are organized in arrays inside the spine head, as presented in Figure 1.3.

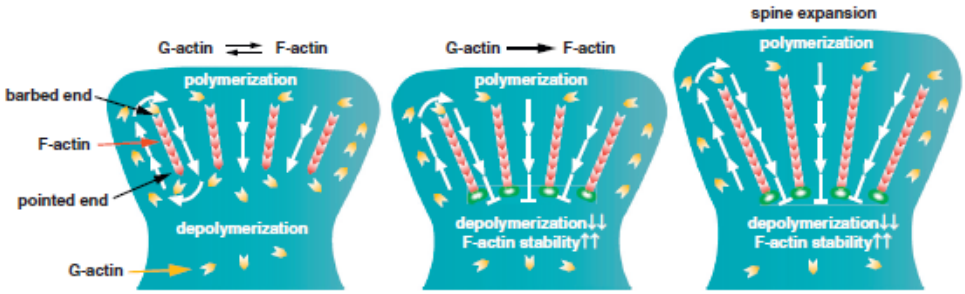


Figure 1.3: Proposed mechanism in [11] for spine expansion. F-actin, a part of the spine cytoskeleton, exhibits treadmilling, a constant migration of actin monomers from one end to the other. In stable spines, this cycle is equilibrated. LTP induction, however, slows down the departure of actin monomers, increasing the size of F-actin and generating the driving force that expands the spine head.

It was found that G-actin undergoes a constant cycle, leaving the pointy end of the array and attaching to the barked end, a behavior called treadmilling. When spines are frequently stimulated, the depolymerization rate at the pointy end decreases, which makes new monomers accumulate at the barked end and increase F-actin length. This would thereby generate a force on the membrane capable of enlarging the dendritic spine. This is also supported by other studies which note that most signaling pathways controlling spine shape seem to directly or indirectly regulate the actin cytoskeleton [10][12].

1.1.5 Function

Although spines have been studied under multiple facets, their impact in networks, and hence their function, is still debatable. Nonetheless, it has been suggested that spines act as an isolated biochemical and electrical compartment, which enables each synapse to be regulated independently [13][14][15] . As a result, individual synapses can be strengthened or weakened, increasing the specificity of the formation of neural networks. The ability of synapses to change their strength in response to their activity is what is meant by the term synaptic plasticity [16], and the reason why they are closely associated with this phenomenon.

The main mechanism thought to promote synaptic plasticity is Long-Term Potentiation (LTP), in which synaptic connections become stronger by a brief period of frequent activation [17]. Frequent activation translates into a frequent arrival of action potentials into a certain axon terminal, which results in a high number of neurotransmitters in the synaptic cleft. In the most studied type of LTP, these neurotransmitters activate the AMPA receptors (AMPAR) of the postsynaptic neuron's membrane (Figure 1.1-c), which depolarizes and enables calcium to flow through its NMDA receptors. The calcium influx initiates a set of intracellular mechanisms that cause more AMPAR

to be inserted in this region of the membrane, which makes it further sensible to ions and thus more likely to be activated in the future.

Spines can affect this process in different ways. To be inserted in the synaptic site, AMPAR must either diffuse laterally through the membrane, or be carried by exocytosis vesicles [13]. It was found that spine necks restrict the lateral diffusion of AMPAR [18][19] [20]. For instance, AMPAR at spines with a distinguishable neck exhibit a twofold slower rate of lateral mobility compared to those without one. Additionally, endocytic and exocytic zones were found within the spines, which can be regulated and thus affect the number of AMPAR in the membrane. Calcium ions can also be controlled by spines, which keeps them in internal stores where they can be released to cause sudden bursts, promoting AMPAR trafficking among other molecular chains. Spines may also function as electrical compartments, capable of modulating the amplitude, kinetics and integration of synaptic potentials [13]. For instance, studies suggest that long spine necks attenuate synaptic potentials between the spine head and the parent dendrite, influencing their contribution on the formation of an action potential [21].

The connections of spine and glial cells are also not well understood. It has been observed that, for instance in the cerebellum, lamellar Bergmann glial (BG) appendages enclose tightly almost every spine of Purkinje cells. This is typical in the beginning of synaptogenesis, when both BG processes and spines are highly motile. When this stage ends, spines are fully ensheathed and the motility of both elements drastically decreases. The hypothesis that glial processes can bound spines to restrain their movement has been studied [22], but still to no conclusion. On the other hand, its has been verified that covered spines usually form large groups, which could mean that glial processes have a role in regulating the number of these structures. Astroglia in the hippocampus has also been observed to participate in the covering of spines. This process seems to happen unevenly across regions inside the hippocampus, and it has been correlated with different physiological states. To inspect if spines are responsible for the selective behavior of glial cells, the morphology of spines has also been studied within an environment populated with astrocites [23]. Recent discoveries found that glial membranes occur substantially closer to the PSDs of thin dendritic spines compared with mushroom spines. Since mushrooms spines have larger heads and are associated with stronger synapses, glial cells may selectively approach more volatile spines to increase their number and promote specific neural pathways, influencing the process of learning.

1.1.6 Dysfunction

Spine's behavior can also reflect several pathologies, such as Alzheimer's Disease, Schizophrenia, Mental Retardation and Autism Spectrum Disorders. These affect spines differently, or vice-versa, as described in the following topics.

1.1.6.1 Alzheimer's Disease

Alzheimer's Disease (AD), the most common cause of dementia, manifests through early memory deficits, followed by the gradual decline of cognitive and intellectual functions. It is characterized

by loss of neurons and synapses in the hippocampus, cerebral cortex and subcortical regions, as well as the formation of beta amyloid (Aβ) plaques and neurofibrillary lesions (e.g. caused by the aggregation of the tau protein). Mutations in three major genes implicated in Aβ metabolism (APP, PSEN1 and PSEN2) have been associated with familial AD [1]. These mutations cause an increased production of pathogenic Aβ oligomers that are responsible for inducing spine alterations and reducing spine density.

In humans, several imaging techniques have been applied to neurological tissues of Alzheimer's Disease patients [24]. The first relevant experiment was done by using electron microscopy on biopsy tissues, and reported a significant loss of synapses compared with cognitively normal controls. Follow-up studies on post-mortem tissue were able to analyze brain regions which are not amenable to biopsy, and reported significantly fewer synapses in the inferior temporal gyrus (which plays an important role in verbal fluency), in the CA1 region, in the dentate gyrus and in the posterior cingulate gyrus (which is a cortical region affected early during the onset of Alzheimer's disease). More recently, a post-mortem study using intracellular injections of Lucifer yellow in the brains of 5 Alzheimer's disease patients revealed that intraneuronal tau aggregates are associated with a progressive alteration of dendritic spines. Further evidence for the loss of synaptic function comes from in vivo PET imaging studies, which by using radionuclide-labeled agonists for specific neurotransmitter receptors, are able to measure the abundance of these receptors in various brain regions and associate them with synapse densities and distributions.

In transgene mouse models, expressing mutant human APP, studies also suggest alterations in spine density and morphology in the hippocampus, even before the development of amyloid plaques. This suggests that Aβ oligomers initiate synapse alterations before neurodegeneration, and thus their analysis may be a powerful diagnosis and evaluation tool in patients with AD [10][12].

1.1.6.2 Schizophrenia

Schizophrenia is a psychiatric disorder involving disturbing thoughts, emotions and perceptions of the reality. It emerges in late adolescence or early adulthood. Postmortem studies in human brains reported grey matter loss and altered spine density in the cortex and hippocampus [25]. Schizophrenia has been constantly associated with selective group of genes. The three most studied in relation to synapse and spine function are the DISC1, ERBB4 and NRG1. In cultured cortical neurons, a DISC1 knockdown[2] reduced spine area, and in mice carrying mutations in this gene, spine numbers in the hippocampus decreased. However, in another transgenic mouse, in which mutations were induced prenatally, spine density in cortical pyramidal neurons was increased. This suggests DISC1 controls spine morphology in a specific developmental and mutation-depended

[1] Familial AD is a form of AD which can be diagnosed in patients under 65 years, and is the result of a genetic predisposition.

[2] Gene knockdown is an experimental technique by which the expression of one or more of an organism's genes is reduced.

manner, which can be assessed by the detection of spine numbers and shape in the mentioned brain regions.

1.1.6.3 Intellectual Disability

Fragile X Syndrome (FXS), the most common inherited form of intellectual disability, is characterized by reduced limited cognitive ability, hyperactivity, hypersensitivity, anxiety, impaired visuospatial processing and developmental delay [12]. It is caused by a trinucleotide (CGG) repeat expansion, that by inactivating the FMR1 gene on the X chromosome, leads to the absence of fragile X mental retardation protein (FMRP), resulting in FXS. Studies in patients and animal models of FXS have identified marked alterations in dendritic spine morphology. Several studies report abnormally long, thin and immature filopodia-like spines, which suggests that retardation in neuronal connectivity is the reason for brain dysfunction in these patients [10].

1.1.6.4 Autism Spectrum Disorders

Autism Spectrum Disorders (ASDs) include autism, Asperger syndrome and pervasive developmental disorder-not otherwise specified [8]. They are characterized by impairment of social interactions and adaptation, and are mostly heritable (around 80%), suggesting that the disease is largely determined by genes and not by the environment. Despite the high heritability, the identification of genetic factors in ASDs has been proved difficult, due to their associated genetic heterogeneity. Similarly to FXS, ASD is accompanied by an increase of immature elongated spine density. Its detection and determination of the responsible proteins could unveil important genes in these disorders, improving the current understanding of the diseases.

The main spine associated proteins whose genes and/or gene expression are affected by the described diseases are presented in Table 1 of the Appendix. There are still other pathologies with also affect dendritic spines, such as drug addiction (increased density and larger spine heads) and Parkinson's disease (reduced density by 27% and shortening). It was also found that in the majority of these diseases, treatments that alleviated their cognitive symptoms also have been shown to reverse their respective spine pathologies, indicating the importance of these microstructural perturbations in the neurodevelopment of such disorders.

As shown in this section, spine shape is supposed to be closely related to its function. Size is also considered determinant, as a larger spine volume will imply a larger PSD area, which will, in turn, increase the number of receptors and docked vesicles in the presynaptic bouton. With more neurotransmitters being passed between neurons, the synapse will become stronger, reinforcing a given neural network. Incidentally, altered spine morphology and density have been observed in several neurological disorders. Anatomopathological analysis of brains of patients affected by Alzheimer's disease, schizophrenia, intellectual disabilities and autism spectrum disorders indicate protein anomalies which misregulate these characteristics [10] (in Appendix). These findings

underscore the importance of the physical and spatial behavior of spines in regulating proper brain function, and hence of techniques capable of studying them. As the physical and spatial condition of spines can be visually assessed, imaging and subsequent image analysis are the main methods for studying dendritic spines. They can be designed to handle 2D or 3D images, having each modality its own advantages. 3D imaging and processing benefits from capturing the real geometry of neurites, which can only be approximated by 2D methods. Besides, overlaid neurites in the direction of the microscope lens result in unrealistically merged shapes in 2D, which can not only compromise the accurate silhouette of spines, but also their counting. On the other hand, 2D methods are easier, faster, and have lower equipment requirements to be performed. Both modalities are sought from the research communities, although in recent years 3D methods have been predominant [26]

Considering the intimate relation between dendritic spines characteristics (number, morphology, spatial distribution, etc.) and several pathological conditions, this Thesis aims at the development of advanced computational tools for the detailed characterization of spines from fluorescence microscopy images. In neuroscience laboratories, fluoresce microscopy is still a research workhorse for studying neuronal structures, but not many labs have the expertise for advanced image analysis. The information content on the raw data often ends up being underexplored, and the analysis often become laborious, time consuming and error prone when researchers have to rely on manual counting and measurements. This work contributes with several free, open-source computational tools to aid in the detailed analysis of dendritic spines. The subject-matter was proposed by professors Paulo Aguiar (Advisor) and Ana Luísa Carvalho (Co-Advisor), under a collaboration between Instituto de Investigação e Inovação em Saúde/Instituto Nacional de Engenharia Biomédica and Centro de Neurociências e Biologia Celular in University of Coimbra.

The experimental work developed in this Thesis can be divided into three main branches. The first is a novel geometrical 3D image processing algorithm, implemented in MATLAB, capable of automatically detect, segment and measure dendritic spines in image stacks. The second is a machine learning 2D image processing method, implemented in Python, meant to automatically detect dendritic spines in images. The third is a set of statistical tools, implemented in MATLAB, with the purpose of analyzing and unveiling hidden patterns from a set of measurements of synapses, carried either in spines or in the dendritic shaft. The three branches can also be viewed as means of studying spines under different aspects: for the first, spatially and physically (detection and volumetric measurements); for the second, spatially (detection); and for the third, spatially and physically/chemically (the data comprises information on the distribution of synapses as well as the quantity of certain neurotransmitters, which correlates to synaptic strength and, in spines, to size).

The following chapters are organized as follows. Chapter 2 presents relevant State of the Art techniques in Imaging and Image Processing of dendritic spines performed until the date of writing

this work. Chapter 3 outlines the Imaging procedure used to gather image data to be processed, and describes in detail the three data processing algorithms developed in the course of this Thesis. Chapter 4 presents the results of the three methods and discusses their implication on modern neuroscience. Chapter 5 concludes by outlining the achievements of this work and sharing considerations made by the author.

Chapter 2

State of the Art

Given the biological significance of dendritic spines and how it can be visually assessed, several methods have been devised to image and analyze their characteristics. In this section, microscopy techniques and image analysis algorithms are described, outlining the breakthroughs as well as the limitations of past and current approaches.

2.1 Image Collection of Dendritic Spines

In order to be distinguished, spines (and neurons) must first be stained with an appropriate marker. Golgi staining was the first to be used to this end, and still one of the most reliable histological techniques in the morphological evaluation of dendritic spines . Other techniques include the use of fluorescent dyes (such as Lucifer yellow), antibodies (such as anti-IgA), and fluorescent proteins produced either by controlled mutations or viral vectors (such as lentivirus and rabie virus) [27].

2.1.1 Widefield Microscopy

Golgi staining has been used in conjunction with widefield microscopy. Despite its simplicity, this technique greatly underestimates the number of spines present on a given stretch of dendrite, due to the lack of z-plan resolution and the small volume of spines (the underestimation can be higher than threefold on hippocampal neurons and even higher on Purkinje cells). To offer better resolution, this optical technique is also used with fluorescent dyes, but in return images become blurred, due to the emitted fluorescence being detected not only from in focus photons, but also from molecules excited out of the focal plane.

2.1.2 Confocal Microscopy

Confocal microscopes solve the previous limitation by using a pinhole to selectively collect emission from the focal point. This is a powerful tool to image changes in spine density and plasticity in brain slices and cultured neurons with nearly diffraction-limited resolution. Confocal microscopy

has been used to image fluorescently labeled structures and Golgi staining was not considered suitable for these observations .

2.1.3 Two-photon Microscopy

This technique is able to restrict the fluorescence excitation intensity needed, for instance, in confocal microscopy, and thus reduce photo-damage and photo-bleaching. It can be used *in vivo* through a craniotomy, and shows a high focus range (0.5-0.8 mm) with high resolution.

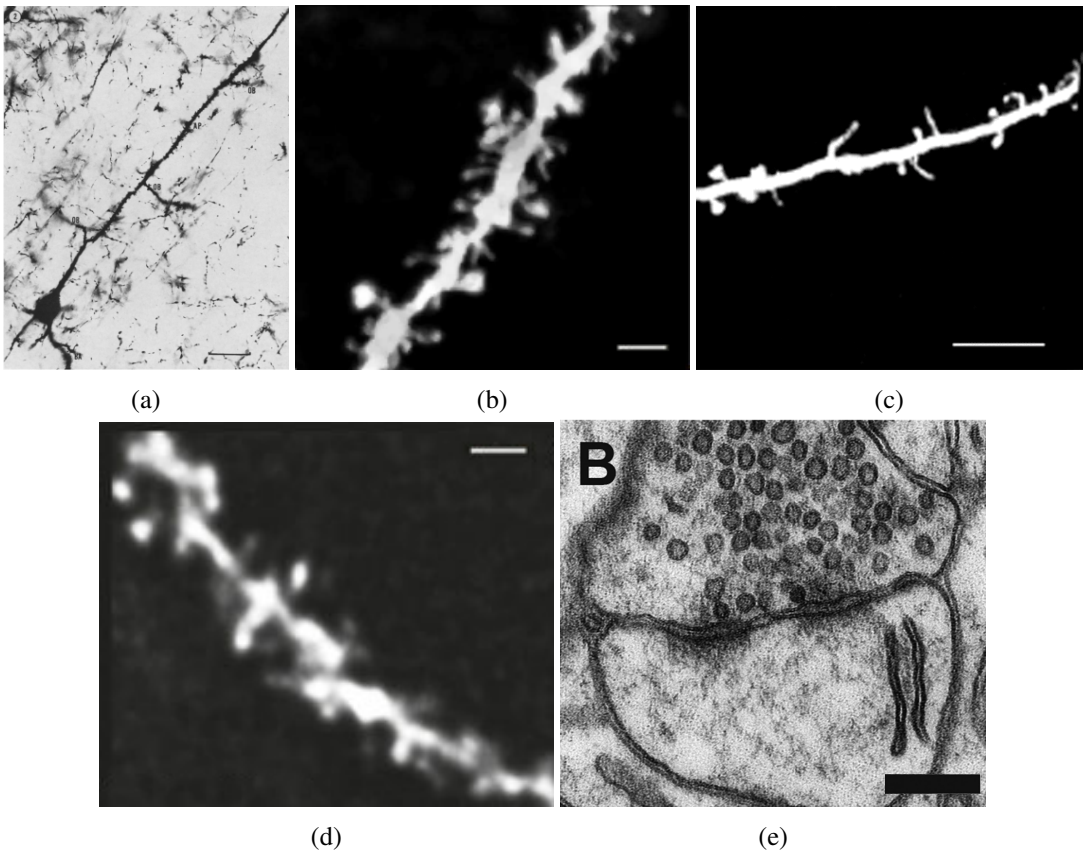
2.1.4 Micro-endoscopy Microscopy

As many brain areas are inaccessible to conventional optical microscopy, to increase focal range even further, an optical micro-probe can the inserted into a specifically targeted deep structure. Adjusting the position of the microscope into the probe, a focal depth of more than 1 cm can be achieved with this highly invasive technique.

2.1.5 Electron Microscopy

This would be the method of election in the detailed study of the composition and activity of spines, since it has a higher resolution than the others. However, to evaluate large amounts of spines (to estimate spine densities as an example), this type of microscopy can be slow, since broad areas have to be run to capture a sufficient amount of spines. Moreover, the required equipment is found more rarely in Neuroscience laboratories than techniques such as confocal microscopy, and for these reasons, the present work focuses on the study of spines from confocal microscopy images.

Images from different types of microscopies are presented in Figure 2.1.



(a) (b) (c)

(d) (e)

Figure 2.1: Neuronal images collected by different microscopy techniques - a) widefield microscopy (scale bar: 25 μm) [28]; b) confocal microscopy (scale bar: 2 μm) [29]; c) two-photon microscopy (scale bar: 10 μm) [24]; d) micro-endoscopy microscopy (scale bar: 2 μm) [30]; e) electron microscopy (scale bar: 200 nm) [31].

2.2 Image Analysis of Dendritic Spines

As current high-resolution imaging techniques allow the collection of massive amounts of three-dimensional anatomical data, the limiting factor in the quality to spine morphology studies has become its accurate quantification and classification [27]. Manual or computer-assisted manual segmentations are slow processes, prone to subjective errors and changes in criteria from study to study. Thus, automated spine detection and characterization algorithms are mainly expected to: remove subjective errors from spine counting; quantify spines based on objectively defined morphological parameters; perform a fast, accessible and automatic analysis on large datasets. Current solutions can be categorized into two approaches: global and local. Global approaches process the initial image in its entirety, to then determine which of the processed pixels/voxels belong to spines. These methods can further be divided into skeleton- and surface-based, according to the region of the neurite they use to distinguish spines from the dendritic shaft. Generally, these methods are computationally intensive, and tend to scale poorly with image size, although there are preprocessing techniques which can reduce significantly the number of operations performed. In contrast, local methods first locate an initial point, usually inside one or multiple spines, and

then exploit local image properties to segment the whole target structure. They process only pixels close to these structures, and so can also be termed "exploratory algorithms". The processing speed is generally higher than global methods, although they require initial seed points, which can either be identified manually, or automatically by a simple heuristic method.

Both approaches start with a preprocessing step, which take the quality of the data set into consideration. Automated image segmentation is hampered by noise (inevitable statistical fluctuations and irrelevant structures), low resolution (in optical microscopes, ultimately limited by light diffraction), inhomogeneous contrast (imperfect distribution of the dye) and background gradients (nonuniform illumination). To some extent, these artifacts can be removed by image processing operations such as smoothing, deconvolution (which corrects the point spread function PSF of the optical system), shading correcting and morphological filtering. When both high magnification and large field of view are required, it is also often necessary to make montages, which integrates multiple images (image registration) and stitching to avoid discontinuities.

2.2.1 Global approach

The general sequence of tasks followed by this approach, after Image Preprocessing, is: Dendrite Extraction, Spine Detection and Spine Classification [32][33]. Dendrite Extraction remains the most challenging task as it must allow the correct identification of their protrusions as spines in order to make the following steps successful. It commonly begins with a binarization step by an adaptive threshold, to yield an initial segmentation of the dendrites (alternative methods based on contour segmentation have also been proposed). Next, a skeletonization algorithm is usually applied, although dendrite centerlines can be obtained by filtering the grayscale images directely with an Hessian of Jacobian operator, or by applying nonmaximum suppression. The result of the skeletonization often contains errors (e.g. detached spines, false protrusions), which can be rectified by geometric methods that take the diameter of the dendrite, the length and/or shape of the spines into consideration. Among the first generation of spine segmentation algorithms, the method demonstrated by Koh et al. (2002) [34] provides automatic detection and quantification of 3D spines, through measures of length, volume, density and shape. The method first extracted the dendritic centerline, and then geometric analysis was performed to detect attached and detached spines according to the shapes of each candidate region. Multiple regions could be then merged into one spine according to their combined shapes. It showed a sensitivity of 95% (percentage of correct spines that were detected automatically compared to manual detection). However, the method is based on geometric constraints of spines and led to false positives dependent on the level of the shape threshold. Of the more recent tools, NeuronStudio, from Rodriguez et al. (2008) [15], proposes to first remove the dendrite, then connect a series of spherical shapes along its centerline, and finally detect spines sufficiently close to it, replacing them also with spheres. The program's number of detections were comparable to the ones from manual spine counting, and also performs at a fast rate. However, it exhibits shortcomings in identifying individual spines from clusters, as well as in lacking image segmentation capability to characterize spines in terms of size, shape, volume, amongst other features. From around the same time (Cheng, Zhang, et.

al., 2007) [35], NeuronIQ is a tool which allows both spine density and morphological features to be determined. After extracting the dendrite centerline, a level-set model is used to segment and detect spines. Its sensitivity was 96%, but had a considerable number of false positives, so it was later improved by the same authors. The most recent implementation of the method (Zhang et. al., 2010) [36], calculates the centerlines by a gradient vector flow (GVF) method. Then, Eigen values of the Hessian matrix for each candidate spine voxel are used to classify if it belongs to any of their distinctive shapes. If so, a level-set method is initiated to segment the final spines. Its sensitivity was of 90% and, in general, it remained too slow for systematic analysis of large neuronal populations. GVF results are displayed in Figure 2.2.

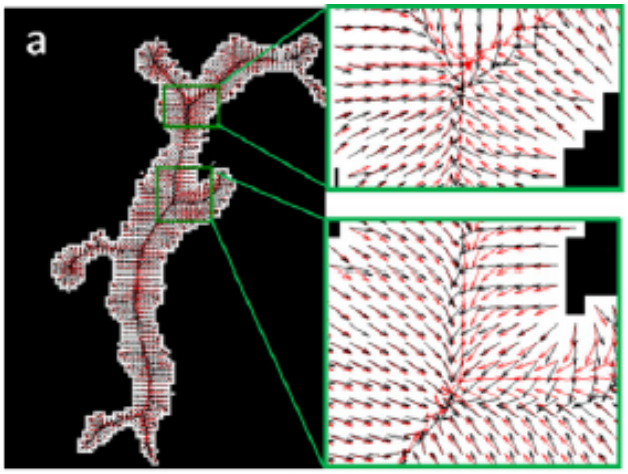


Figure 2.2: GVF method for centerline determination using the default parameters (black arrows) and a strong smoothing criterion (red arrows). Adapted from [26].

Rather than using spine segmentation directly after removing the dendrite, Janoos et al. (2009) [37] proposed to first use a level-set method for segmenting the entire dendrite and spines and then extract its surface. Then, curve-skeletons were extracted using the medial geodesic function. Finally, according to the diameter of the circles, spines were identified from the dendrite, with a sensitivity of 95,3%. As the tips of the spines can be identified by the previous method (correspond to circles with minimal diameter), these points can be used by segmentation methods that start to segment the spine from its tip, as it is the case with one developed by Son et. al. (2011) [38]. This serves to recognize the integrability of some programs and the benefits of their open-source design. Apart from the commercial solutions already mentioned (NeuronStudio and NeuronIQ), there are others that while focused primarily on neuron tracing, are worth referring, such as Neurolucida, FilamentTracer and FARSIGHT. These suffer from some of the same limitations previously described, but are available as a purchasable service/software (Neurolucida and FilamentTracer) or as freeware and open-source (FARSIGHT), which can be an advantage over academic algorithms in a software development context.

2.2.2 Local approach

In direct contrast to the above-mentioned centerline extraction and spine detection pipeline, a second category of spine segmentation explores the image only locally, around relevant structures, rather than processing the entire image. There are at least two advantages of the local over the global approach: the latter (1) usually only works well in uniformly high signal-to-noise ratio images, (2) and it is computationally wasteful (specially in 3D, where only a fraction of the image data contains relevant structures). The lack of efficiency can be compensated by additional methods such as Octree, aimed to reject partitions of the 3D space which are not intersected by the neuronal structure, as shown in Figure 2.3. Nonetheless, the volumes that do not get rejected are usually largely empty (corresponding to the cubes that are not in red in the smallest picture of Figure 2.3), which will increase the computation time unnecessarily in subsequent tasks.

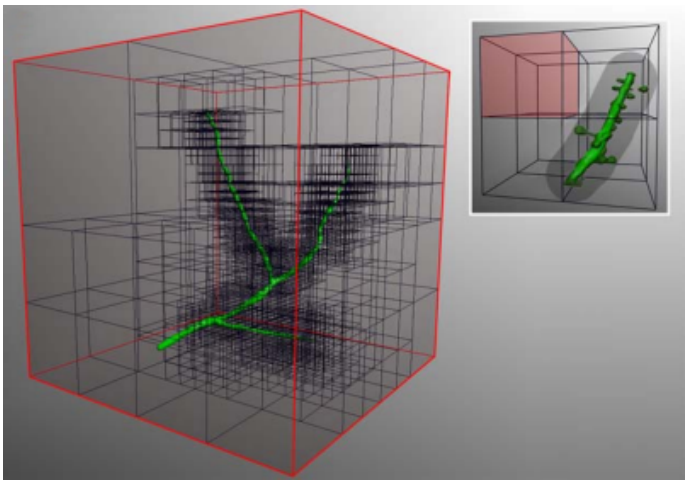


Figure 2.3: Octree calculated for a dendritic section of a neuron to improve processing efficiency. Adapted from [15].

Contrary to global methods, where critical points are usually identified only in last stages, local methods often start with the detection of topological relevant points (either manually or using heuristic automatic detection schemes). He et al. 2012 [39] proposed a method that first detects the tips of the spines, and then uses a region growing or GVF base method to extract the entire spine. The tips are detected by juxtaposing cross sectional areas along the dendrite limited by its membrane or spine membrane, depending if the site has spines. As these sites will generate much greater cross-sectional areas, their most distant point is taken and used to segment the respective spine. This method shown a sensitivity of 97% and is far more computationally efficient than combining centerline extraction and GVF methods. A cross-sectional representation is shown in Figure 2.4.

More robustness can the acquired if the algorithm can be constrained between a starting and ending point defined by the user. In this manner, a cost function can be employed to promote segmentation in intracellular voxels as well as the approximation to the end point [40].

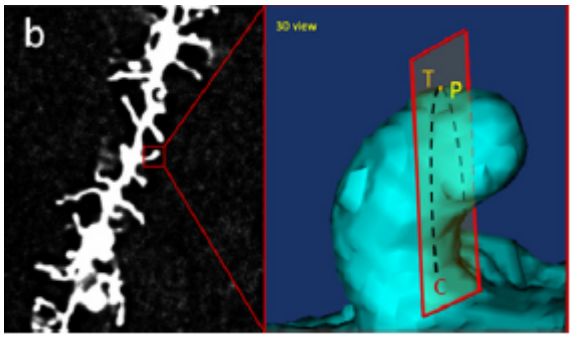


Figure 2.4: Detection of spine tip area using minimal cross-sectional curvature. Adapted from [39].

Besides a few limitations already outlined in the description of these methods, there are others, regarding practicality, yet to be overcome. For instance, the support of different image formats is not offered by the majority of these methods, which impedes cooperative work between research teams. A large number of user-settable parameters is also necessary in various applications, which turns the task less automatic and can lead to an arbitrary regulation of the program controls by the researcher. These parameters are suggested to be reduced, or turned into biophysically meaningful and in independent of underlying technical issues. Moreover, although there is a considerable set of metrics these programs are able to analyze, the extent to which they capture essential functions of dendritic spines is questionable. New methods are therefore encouraged, by several authors, to incorporate novel descriptors of spine morphology, be restrained in the number of user-definable parameters and facilitate data interchangeability, while also fulfilling the goal of surpassing manual methods in either efficacy, objectiveness and speed [41].

Chapter 3

Methods

3.1 Image Collection of Dendritic Spines

All images used by the Geometrical and Machine Learning algorithms of this Thesis have been acquired by the researcher group led by Professor Ana Luísa Carvalho, in Centro de Neurociências e Biologia Celular de Coimbra. Five image stacks were collected from organotypic cultures of neurons from the hippocampus of Wistar rat, with ages P6/P7 and genders male/female, using a Confocal Laser Scanning microscope (Zeiss LSM710). The images were acquired with a 63x objective with a 2x optical zoom. All image stacks were created with a spatial resolution of $66nm \times 66nm \times 19nm$ per voxel.

For the Synapse Toolbox, pyramidal neurons from the hippocampus (CA1 region) of rats from age P21 were imaged. Fibronectin intrabodies were used to label inhibitory postsynaptic compartments along the dendrites. Specifically, the intrabody used recognized gephyrin, a scaffolding protein that is essential for GABA receptor clustering at inhibitory synapses, and was fused to tandem-dimer tomato (Geph-FingR-tdT) to label these in red. The delivery of this probe was done by *in utero* electroporation together with a plasmid that expressed cytoplasmic EGFP, to also label the dendrites and excitatory synapses, both in green. To validate this *in vivo* procedure, i.e., to verify that Gphn-FingR-tdTom puncta represent true inhibitory synapses and accurately report the expression levels of endogenous gephyrin, neurons expressing Gphn-FingR-tdTom were immunostained with antibodies against either a presynaptic marker of GABAergic synapses (the vesiclular GABA transporter, vGAT) or endogenous gephyrin. It was found that, in both dissociated hippocampal cultures as well as in cryosections of the hippocapmpus, the intensity of gephyrin expression correlated remarkably well with the intensity of Gphn-FingR-tdTom in dissociated neurons. These results assured that Gphn-FingR puncta reports reliable both the location and the strength of inhibitory synapses along dendrites.

3.2 Image Analysis of Dendritic Spines

This section describes the three algorithms used to analyze images of dendritic spines, either directly by image processing techniques, or indirectly by statistical correlations of data extracted from the images. The first to be introduced is the Geometrical 3D algorithm. It was developed in MATLAB 2018a, and is capable of detecting, segmenting and measuring dendritic spines in 3D image stacks. The second is the Machine Learning 2D algorithm. It was implemented in Python 3.6, using the Anaconda distribution with the Spyder IDE, and focus on the spine detection in 2D images using a deep learning approach. The third is the Synapse Toolbox. It was also developed in MATLAB 2018a and aims at comparing a set of measures from inhibitory and spine synapses. The source code for these three programs and all their methods/algorithms is available on github through the respective URLs: https://github.com/varjak/geom3Dspines; https://github.com/varjak/deep2Dspines; https://github.com/varjak/synToolbox.

3.2.1 Geometrical 3D Algorithm

The proposed geometrical method resembles both global and local approaches for spine segmentation described in Chapter 2. While the entire image is preprocessed and thresholded to obtain two dendrite masks, these masks are not necessarily submitted to any further processing steps. For instance, if spine coordinates were to be manually annotated, they could be used to locally segment spines in these masks, which would deem this algorithm exploratory. On the other hand, a proposed spine detection method can be run to extract the dendritic centerline of one of these masks and find the spine coordinates automatically. This type of operation is typical from global methods, although here only the spine coordinates are used from the centerline analysis, making the overall algorithm independent of the underlying technique, as long as spine coordinates are determined.

The algorithm is organized into modules, which are described in the following sections and illustrated in Figure 3.1, together with the parameters each one uses.

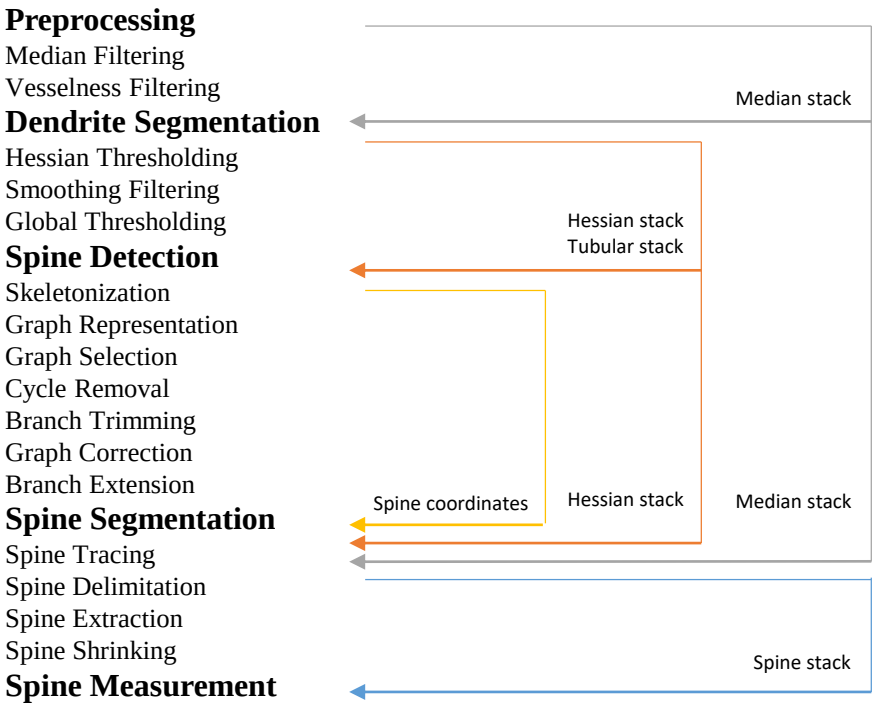


Figure 3.1: Workflow of the proposed geometrical 3D algorithm.

3.2.1.1 Preprocessing

Following image collection, the 3D image stack was deconvolved to compensate for the effect of the optical system (estimated by the point spread function). This was done by the researchers who prepared the image, and thus a deconvolution step was not required for this work. The algorithm begins by loading a set of images and organizing them into a 3D matrix. The images can either be grouped already in a .tif format file, or stored individually in the loading folder as .png or .jpg format files. Then, the color channel (in case the images have more than one, such as in RGB images) with the maximum sum of intensities of all layers is automatically selected from the 3D matrix to convert it to grayscale. The image stack used throughout this section was RGB (with the red channel being the only one with non-zero values) and had a .tif format. Figure 3.2 shows it as a 2D image for visualization purposes. The 2D representation of the stack was generated by a maximum intensity projection along its Z axis. The red region shows the imaged dendrites, and the black the background.

After loading, a $5 \times 5 \times height$ median filter is applied to the grayscale stack. The height of the filter is calculated based on the ratio between the Z and X (or Y) axis resolution of the image stack, which is usually greater than 1 for confocal microscopy images. The calculation is the following (for odd integer lengths or widths)[1]:

$$height = floor\left(floor\left(\frac{length}{2}\right) \times \frac{1}{round(ratio)}\right) \times 2 + 1 \tag{3.1}$$

[1] *floor* denotes the rounding operation of a number to its nearest lower integer, and *round* to its nearest integer.

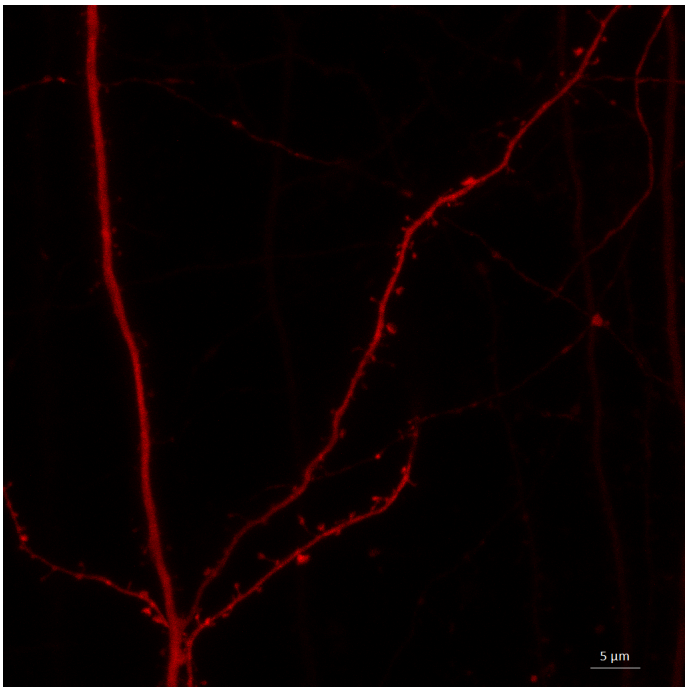


Figure 3.2: Maximum intensity projection of the image stack given to the geometrical algorithm to illustrate its inner works. The maximum projection was also done in MATLAB. The image stack had a size of $1024 pixels \times 1024 pixels \times 46 pixels$, with each voxel measuring $66 nm \times 66 nm \times 19 nm$. The scale bar was inserted based on these values, and may pose as a reference to further image transformations as well.

The filtered image is shown in Figure 3.3 (as a maximum intensity projection). As spines occupy a small portion of the images, turning them hard to see, most of the following figures will depict a small region of interest, shown in the right section to Figure 3.3.

The resultant image is still affected by noise, mainly on the border of the dendrites. The applied median filter uses local statistic information to remove outlier pixels without creating new intensity levels. However, it does not consider the geometry of the foreground, which greatly characterizes both the dendritic shaft and spines. To take advantage of this information, the second order derivative content of the images was analyzed, by first convolving the images with a Gaussian filter, then calculating the gradient of the filtered image, and finally calculating a second gradient for each image produced by the first.

The gaussian second order derivative measures the contrast between the regions inside and outside of a range [-σ, σ], regulated by standard deviation σ of the gaussian kernel, in the direction of the derivative. Similarly to the median filter, the gaussian kernel was also made to have a smaller height than its length and width, expressed by Equation 3.1. In addition, the σ of the Z axis was decreased to span the filter with smaller values along this direction, since, in reality, one filter Z unit corresponds to a larger distance than one X and Y unit. The rectification followed Equation

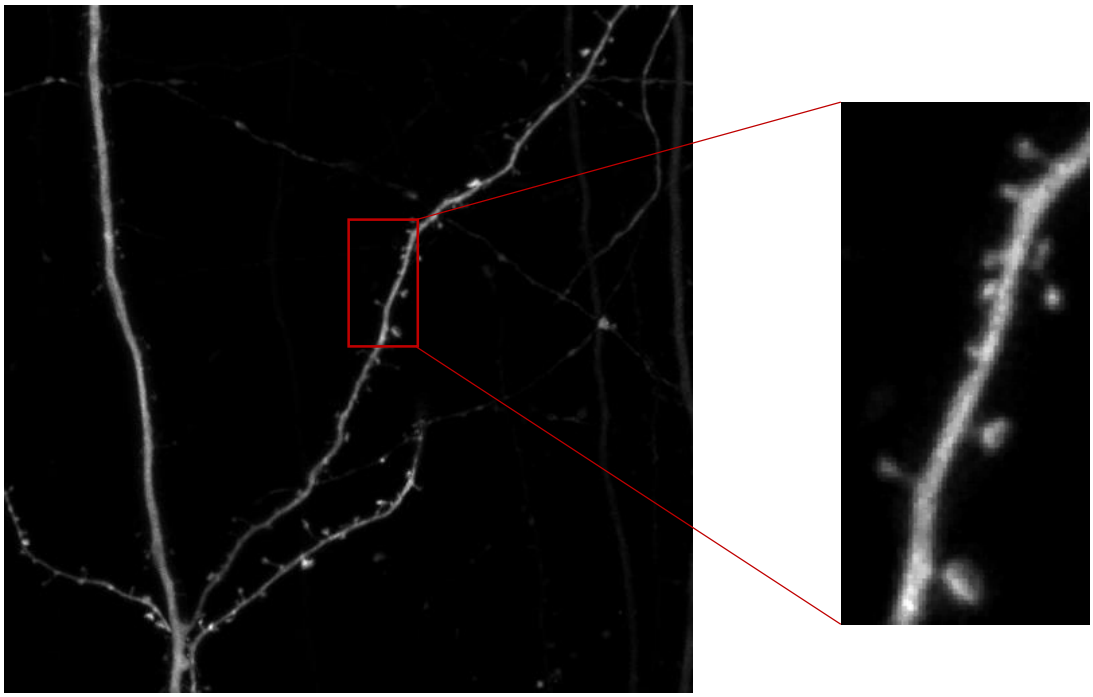


Figure 3.3: Maximum intensity projection of a) - the median filtered stack; b) – a slice of the median filtered stack.

3.2, and its effect is illustrated in Figure 3.4.

$$\sigma_z = \frac{\sigma_x}{ratio} \tag{3.2}$$

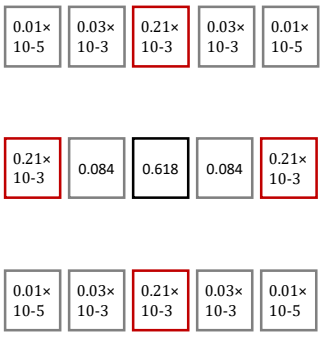


Figure 3.4: A slice of a 5x5x3 gaussian filter with $\sigma = 0.5$ along the XZ plane aimed at an image with a vertical resolution (μm/pixel) of twice of the horizontal resolution. (X axis is horizontal and Z axis is vertical). Given the anisotropy ratio of 2, the sequence of values across Z is a sampled version of the sequence across X, with a sampling rate of 1 value per 2.

The three second order gaussian derivatives, for each axis, are represented in Figure 3.5 for one of the slices of the image stack.

(a) (b) (c)

Figure 3.5: Second-order gaussian derivatives for slice 9 of the image stack: a) – G_{xx} (red) G_{xy} (orange) G_{xz}(yellow) ; b) – G_{yx} (green) G_{yy} (brown) G_{yz} (cyan); c) – G_{zx} (blue) G_{zy} (purple) G_{zz} (pink).

To combine the derivative information into an accurate descriptor of the structure's morphology, the nine derivative values for each voxel were organized into 3×3 matrices called the Hessian matrices. It has been shown [42][43] that the eigenvectors of the Hessian matrix define the principal N directions in which the second order derivatives of an image, with N dimensions, can be decomposed[2]. Each eigenvector is associated with an eigenvalue, which correlates to the curvature found in the direction of its eigenvector. By comparing the magnitude of the eigenvalues, structures with regular curvatures along their axis can be enhanced, such as tubes, blobs and plates. The magnitude relationship associated with each shape is depicted in Figure 3.6 for a 3D image, where λ_1, λ_2 and λ_3 are the eigenvectors sorted by their module.

Structure	λ_1	λ_2	λ_3
	L	L	H-
	L	L	H+
	L	H-	H-
	L	H+	H+
	H-	H-	H-
	H+	H+	H+

Figure 3.6: Structure morphology according to the eigenvalue comparison of the Hessian matrix

The evaluation of the Hessian matrix for each voxel is achieved by Equation 3.3. It combines

[2] In 2D (N=2), for instance, the first eigenvector (the one whose corresponding eigenvalue has the largest absolute value) points in the direction of greatest curvature (second derivative), while the second (the one whose corresponding eigenvalue has the smallest absolute value) points in the direction of least curvature, while also being orthogonal. [44]

three measures of the Hessian matrix - R_a, R_b and S - weighted by three factors - α, β and γ - and outputs the level of resemblance between the neighborhood of the voxel and one of the structures presented in Figure 3.6.

$$O(\lambda_1) = \begin{cases} \left(1 - e^{-\frac{R_a^2}{2\alpha^2}}\right) \times \left(e^{-\frac{R_b^2}{2\beta^2}}\right) \times \left(1 - e^{-\frac{S^2}{2\gamma^2}}\right) & \text{if } \lambda_j < 0 \text{ for } M < j \leq N \\ 0 & otherwise \end{cases} \tag{3.3}$$

Each structure is encoded by the constant M (M C 0,1,2, for blobs, tubes and plates respectively). The constant N refers to the dimensionality of the image (in this case, N=3) and j to one of the N eigenvalues (j C 1, 2, 3). Ra and Rb are ratios comparing different eigenvalues of the Hessian matrix. Each one varies according to the structure M being targeted, as shown in Equation 3.4 and 3.5.

$$R_A = \frac{\prod_{i=M+1}^{N} |\lambda_i|}{\prod_{i=M+2}^{N} |\lambda_i|^{\frac{N-M}{N-M-1}}} \tag{3.4}$$

$$R_B = \frac{\prod_{i=M}^{N} |\lambda_i|}{\prod_{i=M+1}^{N} |\lambda_i|^{\frac{N-M+1}{N-M}}} \tag{3.5}$$

For instance, if tri-dimensional tubes ($N = 3$ and $M = 1$) were to be targeted, these expressions would take the following form:

$$R_A = \frac{\lambda_2}{\lambda_3} \tag{3.6}$$

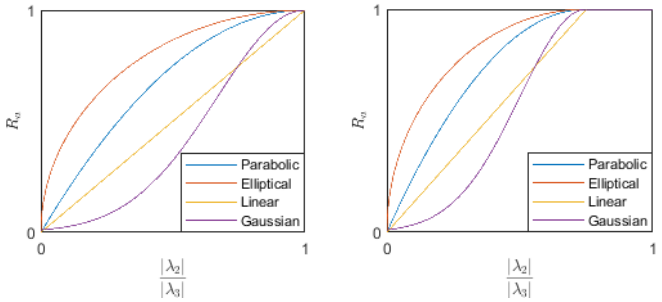
$$R_A = \frac{\lambda_1}{\sqrt{\lambda_2 \lambda_3}} \tag{3.7}$$

As eigenvalues are sorted according to their module, and their values are inversely proportional to the proximity of a border (shown in Figure 3.6), it can be noted that R_A measures the circularity of the cross section of the tube, and R_B the relation between its length and diameter. Both ratios vary along the interval [0,1] and, by Equation 3.3, it can be inferred that the highest tubular response corresponds to the longest and thinnest circular structure [3].

The circularity measure proposed by Equation 3.7 promotes a higher enhancement response for cylinders with a circular cross section. However, dendrites and spines exhibit a range of elliptical cross sections [45] which are not appropriately captured by the previous expression. Moreover, it is not evident that R_a should vary linearly across every value of $(\lambda_2)/(\lambda_3)$. For instance, similar cross sections with high circularity could benefit from having a non-proportional response in

[3] For blobs, the numerator of R_B becomes lambda0, which is not defined, and for plates, the denominator of R_A becomes raised to infinity, leading to an indetermination. Each value is set to zero and infinity, respectively, which cancels their influence in the enhancement function.

relation to sections with low circularity, which could be promptly discarded as non-dendritic material. To allow different response behaviors and expand the range of maximum response, a set of alternative cross section measures were designed and are presented here in both analytical and graphical form. The target_ratio in following MATLAB expressions defines the ratio from which the response is forced to be maximum.



(a) (b)

Figure 3.7: Customable R_a parameters for the calculation of the tube response.

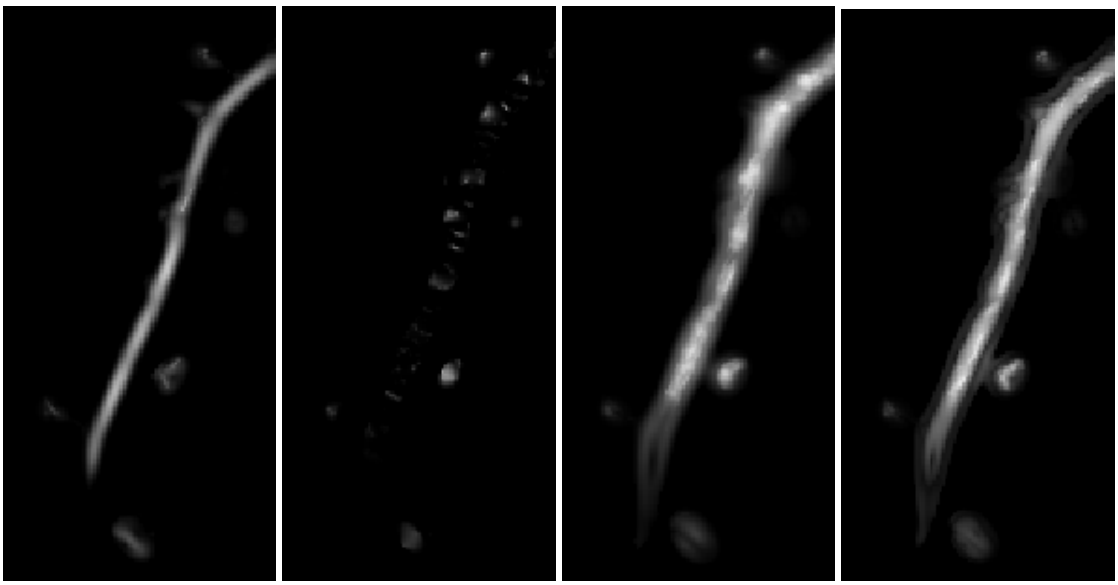
```
   Ra_parabolic = ( -(( (1/target_ratio).*(lambda2/lambda3)-1)).^2+1 ...
);
   Ra_elliptical = (1/target_ratio).*sqrt(target_ratio^2-( (lambda2/lambda3...
)-target_ratio ).^2 );
   Ra_linear = (1/target_ratio).* (lambda2/lambda3);
   Ra_gaussian = exp( -(((lambda2/lambda3)-target_ratio).^2)./(2*(target_ratio...
/3)^2));
```

The final variable in the enhancement function is S, which aims to characterize the intensity level of the region being analyzed. As R_a and R_b are geometrical measures, they would not distinguish small fluctuations in the background as noise, which could yield unexpected results. To discard low intensity regions, characterized by low magnitude derivatives (and hence eigenvalues), the Frobenius matrix was chosen. Following previous recommendations [42] and an empiric trial, all constant weights a B and gamma were fixed at 0.5.

$$S = \sqrt{\sum_{j=1}^{N} \lambda_j^2} \tag{3.8}$$

Independently of how the length of the structures are compared by the arguments of the enhancement function, their range is still limited to the standard deviation chosen for the Gaussian filter introduced before. As the they are unknown and can vary across image regions, a single standard deviation is not feasible to characterize the structures accurately. Therefore, a multiscale Hessian analysis was implemented, which repeatedly calculates the enhancement of every voxel (or a subset of voxels, which would be discussed in the next section) for a defined set of standard deviations. In each iteration, each enhancement value is compared to the previous for the same

voxel, and only the maximum is preserved. Figure 3.8 shows the enhancement results targeting circular tubes, blobs, plates, and elliptical tubes, with an elliptical mapping from eigenvalue ratio into Ra and a maximum response set from a ratio of 0.75.



(a) (b) (c) (d)

Figure 3.8: Enhanced region of interest according to three types of structures: a) Tubes; b) – Blobs; c) - Plates; d) - Elliptical tubes.

All images express a common problem, that is, the lack of several spine necks. Moreover, the circular tube response shortens the diameter of the dendrite shaft and spines, the plate responses enlarges them, and the blob response cuts multiple segments of both parts. The elliptical tube response appears to combine both the circular tube and the plate response, by preserving both elliptical and circular structures, although privileging the latter.

Neither of these images were deemed suitable to proceed into a segmentation phase, due to their inaccurate depiction of dendritic size and spine necks. Thus, a new strategy was adopted, which consisted in using two complementary masks instead of a single realistic mask. One mask would have all spines connected to the dendritic shaft, at the expense of being larger than the dendrite and containing irrelevant structures such as out-of-focus neurites. The other mask would have low noise levels and have all spine heads depicted, at the expense of being smaller than the actual dendrite and lacking on several spine necks. The strategy implied an extensive use of the first mask, enabling the algorithm to detect spines and trace their complete bodies due to the presence of necks. The other mask would be used solely to correct the spine coordinates laid outside the spine heads, in the detection module. In the end, a third segmentation would take place inside the first mask, to identify the spines more accurately. This strategy allows the program to segment spines without an accurate dendrite segmentation, expanding the segmentation options for spine classification. Furthermore, if an accurate segmentation was to be achieved, it could replace each of the above-mentioned masks without changing any other parameter, attesting for

the program's versatility.

The circular tube response features all the characteristics of the second mask, but the intensity values of spines are low, making them hard to segment. To solve this issue, a second version of the circular tube enhancement algorithm was implemented [42]. It yields a higher and more stable enhancement response across tubular structures of different radius, making them easier to segment.

The altered function is presented in Equation 3.9.

$$T(\lambda) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \vee \lambda_3 > 0 \\ \lambda_2^2(\lambda_p - \lambda_2)(\frac{3}{\lambda_2+\lambda_3}) & \text{if } \lambda_2 \leq \lambda_p/2 \\ 1 & otherwise \end{cases} \tag{3.9}$$

λ_2 is unchanged, although λ_3 is normalized into λ_p, as stated in Equation 3.10. This step takes into account the standard deviation of the Gaussian filter, to balance the enhancement response of tubular structures with different radius. The parameter τ controls the intensity of the response, and was set to the maximum value of 1.

$$\lambda_p = \begin{cases} \lambda_3 & \text{if } \lambda_3(x,s) < \tau \times min_x\lambda_3(x,s) \\ \tau \times min_x\lambda_3(x,s) & otherwise \end{cases} \tag{3.10}$$

The enhancement response for the same region of interest is shown in Figure 3.9.

Figure 3.9: Enhanced region of interest according targeting normalized tubular structures.

This image was carried to the segmentation phase, described in the next section, where the creation of the first mask is also detailed.

3.2.1.2 Dendrite Segmentation

The thinner mask was obtained by thresholding the regularized tubular response image of Figure 3.9. Due to its high contrast (as τ was set to its maximum value), a global threshold level was proved sufficient, set to 5% of the image's format range (0.05 on double type images in the range of [0,1]).

The larger mask was calculated based on the Hessian matrices introduced in the last section. As shown, the enhancement of each voxel entailed the construction of an Hessian matrix, the extraction of three eigenvalues, the comparison between them in the form of ratios, and the calculation of the enhancement response. All these steps make up for a computationally expensive process, which is further repeated for every standard deviation used in the gaussian filters. Observing Equation 3.9, it can be noted that an enhancement response is not null only if the following condition is verified.

$$\omega = \lambda_2 < 0 \wedge \lambda_3 < 0 \tag{3.11}$$

The enhancement function then becomes, schematically:

$$T = \begin{cases} response & \text{if } \omega \\ 0 & otherwise \end{cases} \tag{3.12}$$

This implies that if ω could be assessed before calculating the eigenvalues of the Hessian matrix, there would be no need to proceed with further calculations as the enhancement response would be null. In this thesis, one approach was followed [46] to limit the number of voxels requiring further processing in the calculation of the regularized tubular response. The selected subset of voxels forms a mask, which represents a high probability region of finding dendritic voxels. This mask connects all spines and was therefore used as the larger mask throughout this algorithm.

The mask is calculated by first representing the eigenvalues of an Hessian matrix in the polynomial form:

$$p_H(\lambda) = \det(\lambda \times I_3 - H_3) = \lambda^3 + c_1 \times \lambda^2 + c_2 \times \lambda + c_3 \tag{3.13}$$

Where, in MATLAB code,

```
C1 = - (fxx + fyy + fzz);

C2 = fxx .* fyy + fxx .* fzz + fyy .* Gzz - Gxy .* Gxy - Gxz .* Gxz...
- Gyz .* Gyz;

C3 = Gxx .* Gyz .* Gyz + Gxy .* Gxy .* Gzz + Gxz .* Gyy .* Gxz - Gxx...
.* Gyy .* Gzz - Gxy .* Gyz .* Gxz - Gxz .* Gxy .* Gyz;
```

c_1, c_2 and c_3 are the coefficients of the polynomial, calculated by a combination of Hessian elements (second derivatives), and the eigenvalues are the roots of the polynomial. c_1 has also the

following property:

$$c_1 = -\sum_{k}^{N} \lambda_k \tag{3.14}$$

To obey ω, the two roots of highest magnitude of the polynomial must be negative. This can be inspected according to two Theorems:

Theorem 1: If the real parts of all roots of p are negative, then all coeficients c are null or positive.

Theorem 2 (Routh-Hurwitz criterion): The number of roots of p which positive real parts, is equal to the number of sign chances in the Routh array of p.

The Routh array, for $p_H(\lambda)$, is composed of $[c_1, c_2, c_3]$. Following these statements, three conditions can be derived that do not satisfy ω:

1. $c_1 \leq 0$
2. $c_2 \leq 0 \wedge c_3 = 0$
3. $c_1 < 0, c_3 > 0, \wedge c_1 \times c_2 < c_3$

Condition (1) holds because $c_1 \leq 0$ implies $\lambda_1 + \lambda_2 + \lambda_3 > 0$. The three lambdas are sorted by their modules in a growing order, so if λ_2 and λ_3 are negative (as ω requires), then $\lambda_1 + \lambda_2 + \lambda_3 < 0$, which contradicts the above expression. Condition (2) holds because $c_3 = 0$ yields $p_H(\lambda) = \lambda \times (\lambda^2 + c_1 \times \lambda^+ c_2)$. By the zero-factor property, the lambda outside the parenthesis is equal to zero, implying the lambdas inside must be negative to obey ω. However, if c_2 is negative, by Theorem 1 the polynomial inside parenthesis does not have only negative roots, which contradicts Omega.

Finally, condition (3) holds because it implies there are two sign changes along the Routh array. By Theorem 2, this means there are at least two null or positive lambdas in p, which opposes ω as well

Thus, after the computation of the Hessian matrix for each voxel, only the ones where neither of these conditions is met are subjected to the eigenvalue analysis. Their coordinates form a Hessian mask, which together with the mask of the tube response, are utilized in the subsequent steps of this algorithm. Both are presented in Figure 3.10.

The two masks were also processed to enhance their qualities. The tubular response mask was dilated with a spherical structure element of radius 3, expanding the volume of its spines, which will improve the spine coordinate correction in the next module. The processed tubular mask is shown in Figure 3.11.

The Hessian mask, in turn, was blurred by convolving it with an average kernel of 3x3x3, and thresholding it with a value of 30% of its range. This resulted in a smoothed version of the mask, which will facilitate the spine tracing in a later module. The effect of the smoothing is depicted in Figure 3.12.

(a) (b)

Figure 3.10: Slice nº 9 of the unprocessed binary masks from the a) – tubular response; b) – Hessian analysis.

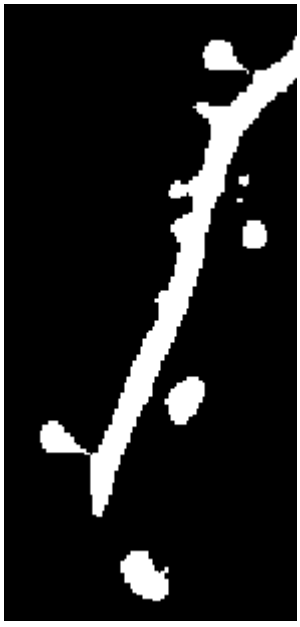


Figure 3.11: Slice nº 9 of the processed tubular binary mask.

It's worth mentioning two last remarks regarding the Preprocessing and Segmentation sections. The first is that the segmentation of the Hessian mask is actually performed during the Preprocessing function of the program. This happens as the calculation of the Hessian matrices already takes place in the preprocessing step leading to the tubular image, and this mask was already implemented to avoid computational overload. It was described here solely due to the nature of

(a) (b)

Figure 3.12: Scatter plot of the Hessian mask a) before and b) after the smoothing operation.

its formation. The second is that more combinations of Preprocessing and Segmentation methods were experimented during the course of this work, but none of them produced reliable results. This obstacle motivated the choice of the describe strategy as a mean to overcome it. Following the workflow of Figure 3.1, both masks are carried to the next stage.

3.2.1.3 Spine Detection

The detection of spines was performed on the Hessian mask. The mask was transformed into a skeleton, using the *bwskel* command available in MATLAB 2018a. As it is a part of MATLAB's repertoire, it is optimized and does not suffer from the time issues reported in other articles [41], regarding the skeletonization of dendrites. The end- and branchpoints of the skeleton (connected to one, and three or more pixels, respectively) were determined by the *bwmorph3* command, introduced in the same version. The skeleton and both types of points are shown in Figure 3.13.

The general idea behind the detection method is to first identify the dendritic shaft in the skeleton of Figure 3.13, then considering each of its branches as one spine. Bifurcated branches are trimmed to their first branchpoint, and the end of every branch (after trimming) is intersected with the tubular mask, to guarantee they belong to the foreground. Finally, each spine candidate is pushed away from their correspondent branchpoint on the dendritic shaft, until they are near the edge of the foreground, to facilitate spine tracing. This approach implies a different set of operations on distinctive points based on their relationship with others. Thus, it becomes necessary to find the connections between adjacent points of interest (branchpoints and endpoints). Resorting to Graph Theory, the relationships between points of interest (POI) were represented with a graph, a structure meant to model pairwise relations between objects. A graph can be created from an adjacency matrix, which is a square symmetrical matrix establishing the connection of an object i to an object j, by assigning a nonzero value to the position (i,j). This matrix can be binary, representing solely the connected and unconnected objects, or real valued, expressing also the weight of each connection. Two adjacency matrices were created: one connecting branchpoints, and another connecting branchpoints with endpoints. To populate them, the branchpoints of the skeleton were

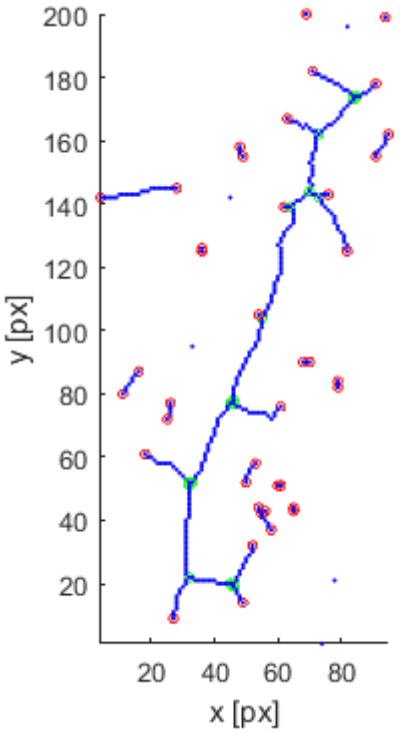


Figure 3.13: Skeleton of the region of interest. Green points represent branchpoints and red points represent endpoints

removed, leaving its "bones" unconnected. Each bone was identified with MATLAB's command bwlabeln and analyzed individually. Figure 3.14 presents the colored bones of the considered region of interest.

For each bone, a pair of new endpoints was calculated, and for each new endpoint a direct neighborhood was defined. Every branchpoint coordinate triplet was checked to whether it belonged to the neighborhood of each extremity of the bone (upper right square of Figure 3.14). If the two neighborhoods were matched to two branchpoints, then these points were found to be connected, and the length of the bone was inserted into the branchpoint adjacency matrix, according to the position of both branchpoint coordinates in the coordinate vector. The neighborhood was also matched against the old endpoints, to populate the branchpoint-endpoint adjacency matrix. A similar matching process was performed on the neighborhood of branchpoints (instead of the extremities of the bones), to connect adjacent branchpoints which otherwise would not be associated (lower right square of Figure 3.14). Based on the branchpoint adjacency matrix, the connections between branchpoints were represented in graph form. The graph is shown in Figure 3.15, with nodes disposed similarly to branchpoint (solely for visualization purposes) and edges with the length of their connections.

The next step in the algorithm is the selection of a single graph from the multiple graphs that could possibly be generated until now. The condition for selection was defined to be the number of endpoints connected to each graph. As most endpoints are included in branches rising from the dendritic shaft, and each branch will be regarded as a candidate spine, the number of endpoints

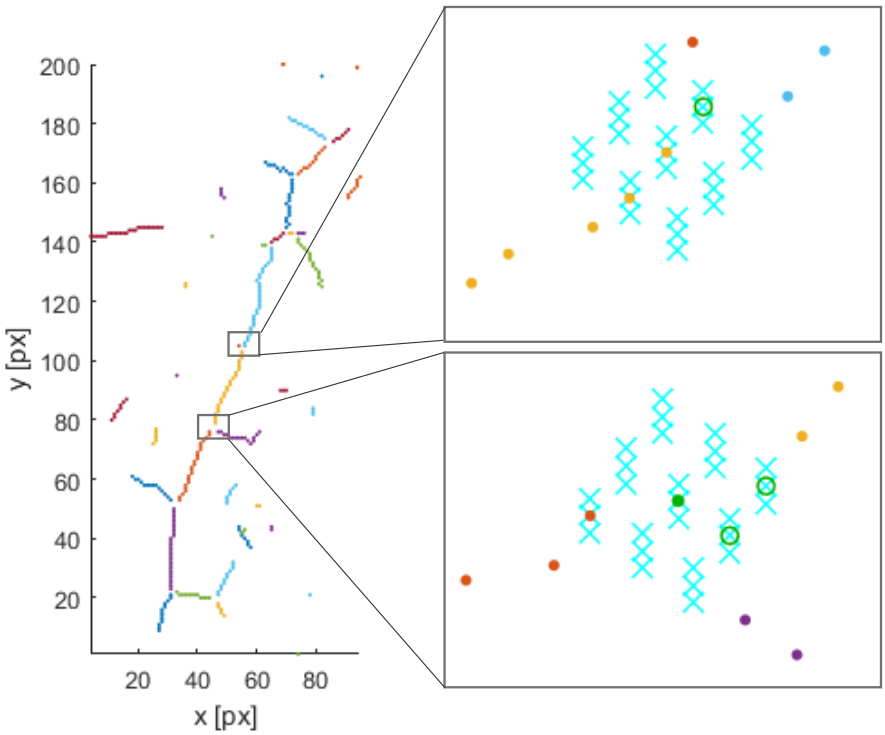


Figure 3.14: Bones of the skeleton. The presence of branchpoints and endpoints is inspected in the extremities of each bone, as well as surrounding the branchpoints themselves.

was regarded as an estimation of the number of spines. The graph with the larger number of endpoints was selected, to allow the detection of as much spines as possible. If multiple dendrites are present in the original image, this process can be repeated for the consecutive "spiniest" graphs to increase the number of coordinates generated. In the considered region of interest, there are no other graphs, so the one presented was chosen by default. In practice, the selection method was written as follows. For every branchpoint, the number of connected endpoints was counted (corresponding to the number of nonzero values in each row of the branchpoint-endpoint matrix) and summed across branchpoints of the same graph. Isolated graphs were distinguished by the command conncomp, and the one with the largest endpoint count was selected. As shown in Figure 3.15, the graph includes a series of cycles which correspond to adjacent branchpoints found by the skeletonization algorithm. Figure 3.14 (lower right) depicts one of these cases. A cyclic graph can hinder the computation of several tasks, such as finding the longest path. Moreover, some of their nodes could pose as false branches and be classified as spines, leading to the generation of incorrect spine coordinates by the end of this module. Hence, a cycle removal step was written, capable of finding all cycles below a specified number of nodes and reducing them into a single node, with the connections of all other nodes in the cycle. To find the cycles, an opensource function credited to J. Jeffry Howbert was incorporated and altered for this project. It works by finding every triplet of nodes in a graph, selecting one of its noncentral nodes, and iterating over the remainder of the graph, using a depth search approach, to find the other. If found, the cycle is

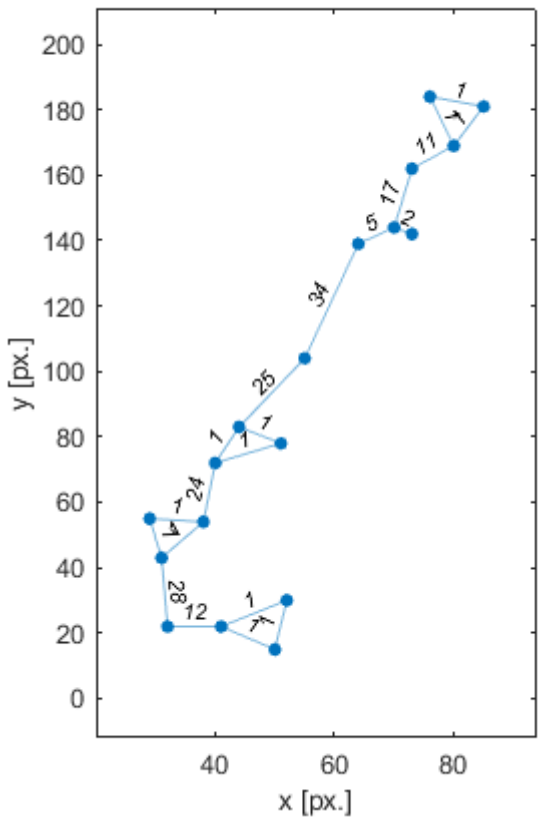


Figure 3.15: Graph connecting the branchpoints of the skeletonized dendrite.

counted as well as its number of elements. The algorithm is illustrated in Figure 3.16.

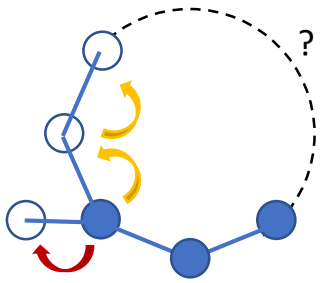
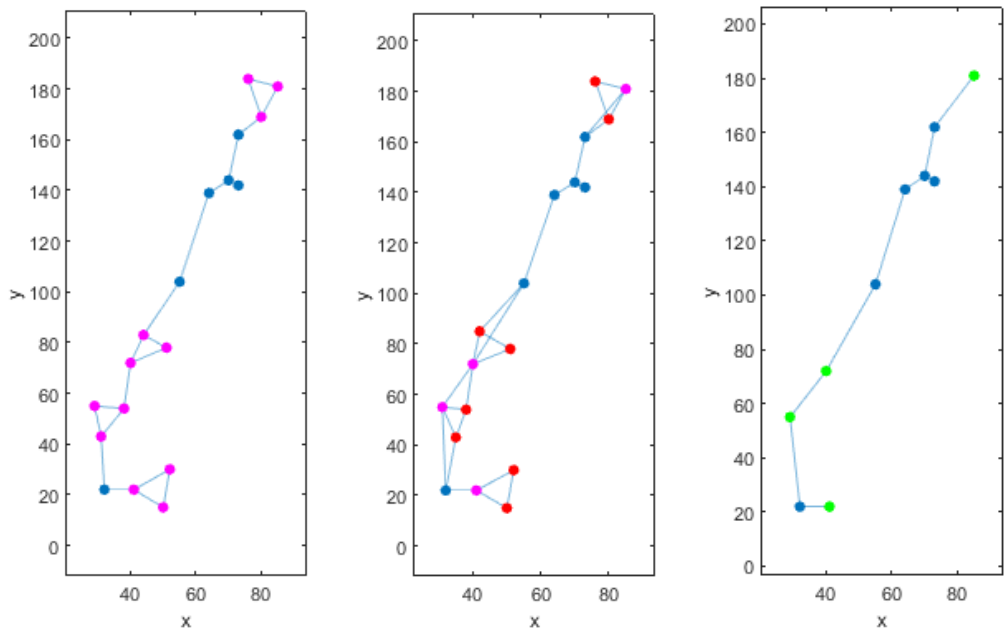


Figure 3.16: Schematic of the algorithm followed by the function cycleCountBacktrack, credited to J. Jeffry Howbert. The blue filled circles represent one of the triplets found by the algorithm, and the white filled circles the nodes that are checked from one side of the triplet. Dead ends are abandoned, and the algorithm stops when (1) the other side of the triplet is found; (2) there are no more nodes explore; (3) or the length of the path exceeds a predefined limit.

As the function did not outputted the sequence of nodes composing each cycle by default, this change was implemented, in the form of a variable that saves the correct nodes during the process. After finding the cycles, the branchpoint adjacency matrix is altered by adding to the row and column of the first node of each cycle, the rows and columns of the rest within the same cycle, and

deleting them afterwards. Figure 3.17 shows the different phases of the cycle removal process.



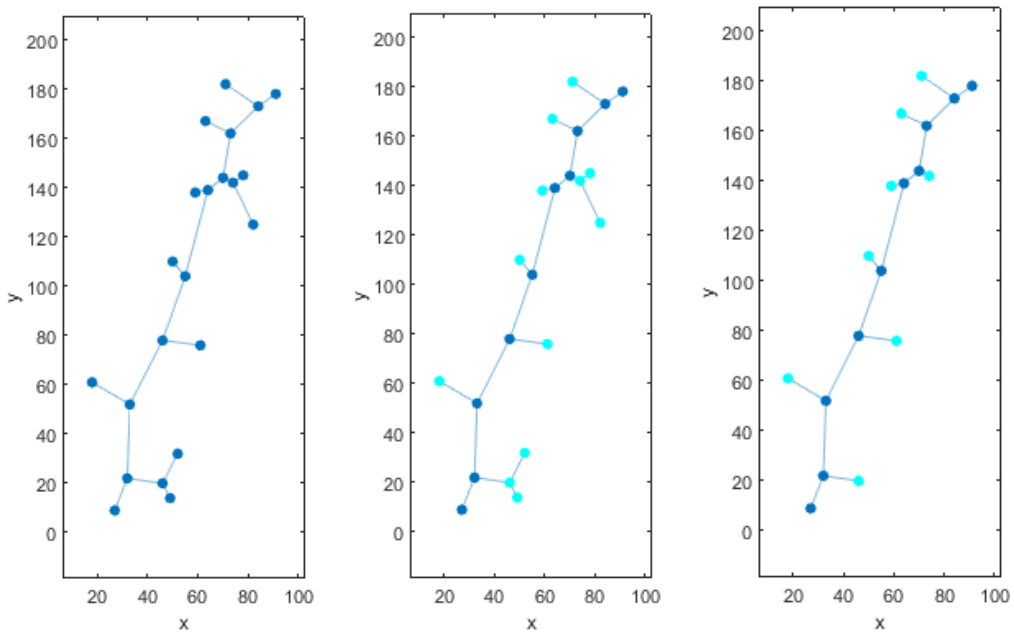
(a) (b) (c)

Figure 3.17: The cycle removal process: a) identification of cycles; b) connection transfer to a single node within a cycle; c) result after removal.

The branchpoint adjacency matrix was then merged with the branchpoint-endpoint matrix. They were kept separated until this point because endpoints do not form cycles, and they would just increase the computational time of the cycle removal phase. The global graph is shown in Figure 3.18 a). To identify branches of the graph as candidate spines, its shaft needs first to be calculated. The shaft is distinguished from the branches by its larger length. MATLAB offers a function named *graphallshortestpaths* which calculates the length of the shortest path between every pair of nodes in a given graph. This function was used to determine the two extremities of the shaft, by choosing the pair with the highest distance in between. However, the sequence of nodes from one extremity to the other is not outputted by the function, and so the extremities of the shaft were fed into a second function, named *shortestpath*, to get the shortest path between them as a sequence of nodes. The dendritic shaft is shown highlighted in the graph in Figure 3.18 b)[4].

Knowing the shaft of the graph, the branches were trimmed by determining the neighbors of each node of the shaft, with the function neighbors, and keeping only the ones not belonging to the shaft itself. These, together with the shaft nodes, make the graph of a trimmed dendritic skeleton, with all spine candidates identified. Figure 3.18 c) shows the result of the trimming process.

[4] Note that since both branchpoints and endpoints are used in the calculation of the longest path (the dendritic shaft), the ends of the dendrite will be cataloged as the ends of the shaft and not erroneously as spines.



(a) (b) (c)

Figure 3.18: The branch identification and trimming process: a) the graph containing both branchpoint and endpoints; b) – the graph with identified branches [5]; c) the graph with trimmed branches.

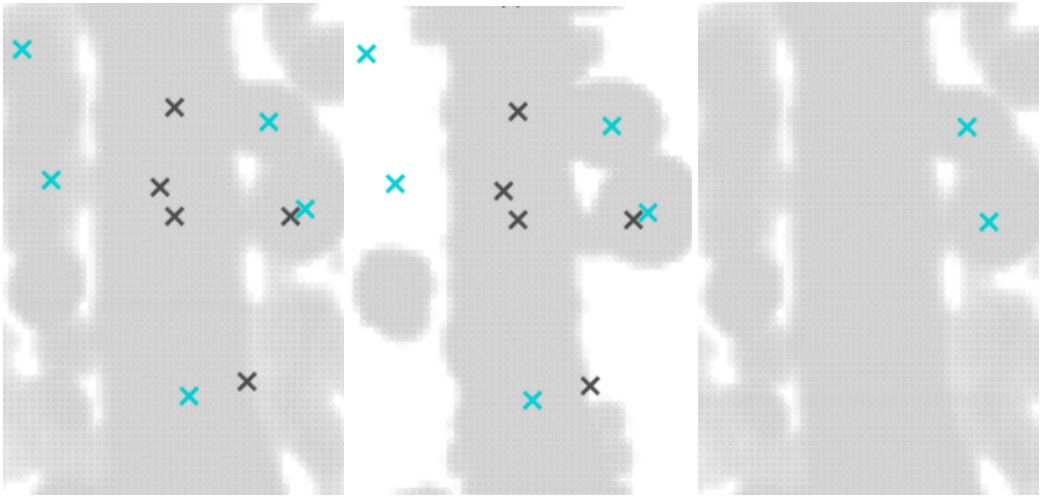
The spine candidate coordinates were then subjected to a correction phase. Both spine and branchpoint coordinates were compared with the foreground of the tubular mask. Every spine that didn't belong to the foreground or was connected to a branchpoint that did not, was removed. This reduced both spines marked outside the dendrite and inside the dendritic shaft, in cases when the branch point was incorrectly marked in the periphery of the shaft. The correction phase is illustrated in Figure 3.19.

The last phase of the detection module aims at improving the position of the spines which passed the correction phase. The tracing module has a higher success rate if it starts from near the spine tip, and so the spine points were pushed away from the respective branchpoints in an attempt to reach this region. They were positioned three pixels from the edge of the foreground, as shown in Figure 3.20.

From lower to upper branchpoint, it can be observed that the first spine point did not move, due the fact of being already close to the surface of the dendrite; the second spine point moved into a position where the tracing has higher odds of succeeding; the third spine point moved into a position far from the tip of the spine [6], although considerably better than the original; and the forth

[5]The weight of the lower and upper branches was manually assigned to a high value to exclude them of being classified as branches. This was done for visualization purposes, as the region being presented cuts the original dendrite, producing two artificial segments whose small length would trick the branch classification, if not compensated.

[6]The region actually belongs to a combination of two spines, which were too close to be distinguished by the segmentation module. The tracing phase will attempt to segment it as a whole, which is facilitated by the spine point reallocation.



(a) (b) (c)

Figure 3.19: Correction of spine candidates (marked in blue): a) – Hessian mask with all spine and branchpoints (marked in dark gray); b) – Tubular Mask with the same points; c) Hessian mask with the corrected spine coordinates.

Figure 3.20: Reallocation of corrected spines into the neighborhood of an edge. Branchpoints, corrected spines and reallocated spines are represented with dark gray, cyan and red crosses, respectively.

spine approached the tip, although it was already well positioned. To improve the location of spine points such as the first and third, an alternative procedure was written. In addition to moving the points until the last three voxels of the dendrite are reached, the alternate method also estimates de direction of movement, by scanning the dendrite around the branch point, with equally distant

segments of a large length. The scanning around the third branchpoint is illustrated in Figure 3.21. This process is also called Rayburst and will be extensively used in the tracing phase.

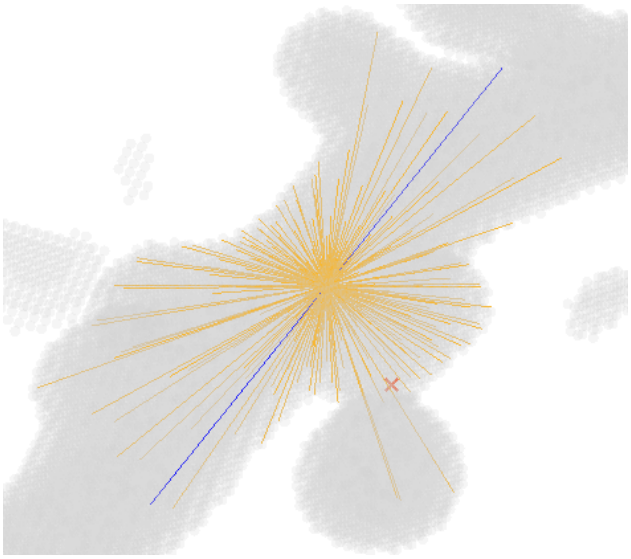


Figure 3.21: The scanning of the region around the branchpoint by multidirectional rays.

The rays were projected from the branch point until they reached a surface or the length limit, and were saved as vectors spherical coordinates in reference to the branchpoint. A 2D interpolation was performed to increase the number of points and fit them into a surface in a spherical coordinate system, which is presented in Figure 3.22.

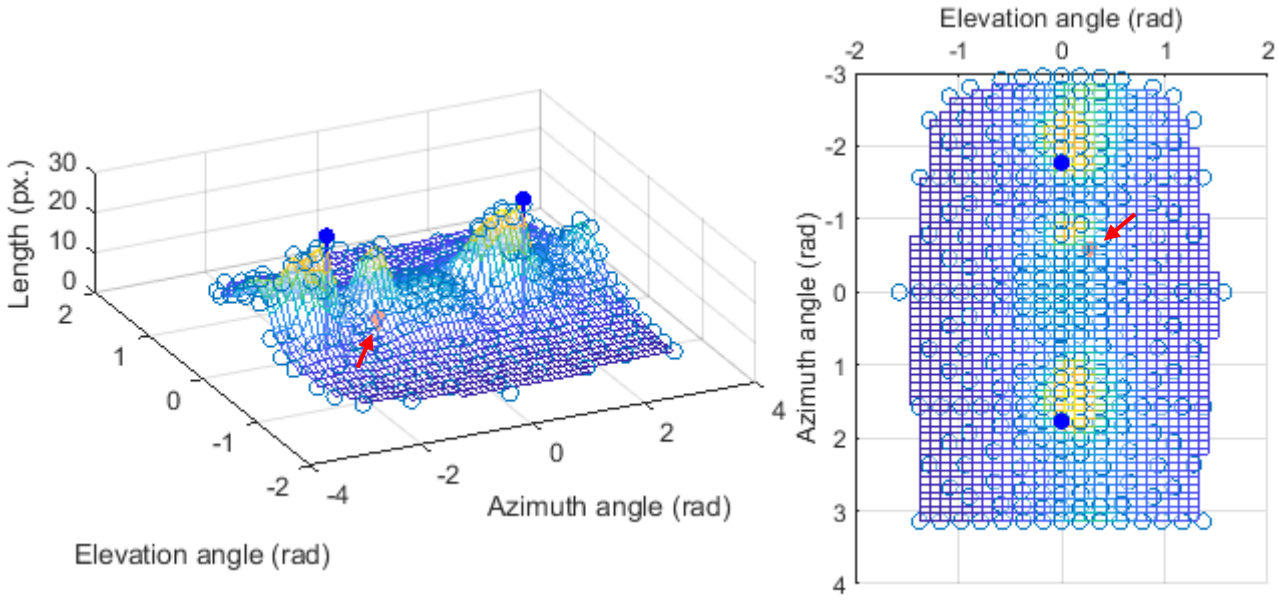


Figure 3.22: Final points of every ray represented by spherical coordinates and connected with a surface. The empty spheres represent the original points, the blue-fill marks those corresponding to a local maximum, and the red sphere represents the spine point determined by the last phase.

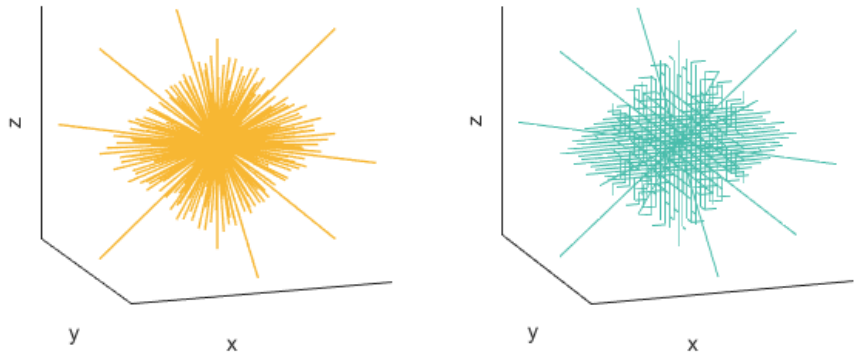
There are three main peaks regarding vector length. Two of them correspond to the opposite directions of the dendritic shaft (detected by the two blue rays in Figure 3.21), and one to the spine close to the branchpoint. The aim of the alternate reallocation phase was to move the spine point along the direction of its nearest peak. For this case, the spine point would be moved successfully to the tip of the spine. However, for cases where the branchpoint is marked close to the tip of the spine, such as the upper point of Figure 3.20, the nearest ray length peak would not the along the direction of the tip, but of the neck or even outside the spine. By making the position of the spine points insignificant, this method would have to be more elaborate to account for every possible relationship between the branchpoint and the dendrite. As so, it was chosen to consider the spine coordinates from the skeletonization phase instead, and apply a simpler method to move the spine points towards the tip of the structures. These were regarded as the final coordinates of the detection phase and carried to the spine segmentation module.

3.2.1.4 Spine Segmentation

The spine segmentation phase aims at separating each spine from the dendritic shaft of the Hessian mask. As spines are connected to the shaft by their neck, the region of separation should be the beginning of the neck (its proximal region), which has first to be identified. To this end, an exploratory algorithm was developed, able to trace the spine from its detected coordinates to the proximal neck region, based on the Rayburst tracing technique [15]. Once the region is identified, a delimitation phase draws the border between shaft and spine, and an extraction step isolates spines by preserving their voxels while removing the rest. The Tracing, Delimitation and Extraction phases are presented individually through the remainder of this module.

Tracing The concept of the Rayburst tracing relies on estimating the orientation of a tubular structure by projecting multiple rays from one of its interior points onto its surface. The rays are equally distant and can be applied in several dimension spaces. To use them in the dendritic images, a core of 129 equally distant 3D rays were discretized into paths so they could be intersected with the image. In this section, one ray will be regarded as a line cast without parallel, i.e., the two segments projected in opposite directions from the center of the core will be regarded as a single ray. The continuous and discrete cores are shown in Figure 3.23.

These cores are used to estimate the topology of the structure and find the tracing direction. For instance, in a perfect infinite 2D cylinder, the shortest ray cast from any given point inside it corresponds to a line with the diameter of the cylinder. The centerline of the strucure can then be traced by moving a point from the center of the shortest ray in the direction orthogonal to the ray. In 3D, the shortest ray may not correspond to the diameter of the cylinder, but will be perpendicular to a line that will. A 2D Rayburst can be performed after it, slicing the first shortest ray in half, and calculating its own shortest ray, which will be a line with the diameter of the cylinder. The centerline can be traced analogously as before, by moving a point from the center of the second ray in the direction orthogonal to the plane formed by the two shortest rays. This procedure is illustrated in the left graph of Figure 3.24 with a cross section of a perfect 3D cylinder.



(a) (b)

Figure 3.23: Continuous and discrete cores of 129 multidirectional 3D rays.

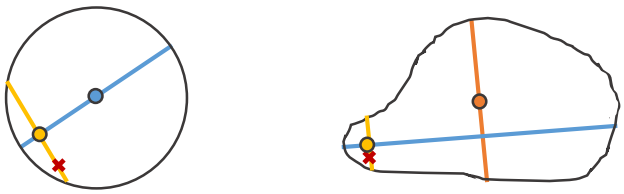


Figure 3.24: Shortest rays and center points of a 3D Rayburst viewed in the cross section of a) a regular cylinder, and b) an irregular structure. The red cross represents the starting position of the process. The yellow line represents the shortest ray of the 3D Rayburst centered on the starting position, and the yellow circle its middle point. The blue line represents the shortest ray of the 2D Rayburst centered on the yellow point, and the blue circle its middle point, which is also the estimated center of the cylinder. For the atypical structure, the estimated center is corrected by applying a 1D Rayburst parallel to the shortest ray of the 3D Rayburst, and calculating its middle point.

However, dendrites and spines are not regular cylinders. In these cases, besides the 3D and 2D cores, a "1D core" (a single ray) is cast parallel to the first shortest ray, and from the center of the second, to correct the position of the tracing point due to the structure's irregularity. The method is more robust and further facilitates the tracing of spines on the smoothen Hessian mask. The right graph of Figure 3.24 illustrates it on the cross section of an atypical surface.

The proposed method was applied for each detected spine. The core of 129 rays, saved as a matrix with 129 vectors pointing equally distributed directions, was converted to a *cell* [7] with the coordinates of generic discrete paths for each ray. The paths were then centered around the spine coordinates, and the correspondent intensity values of the image were sampled in the same order as the coordinates of the path. Counting from the center, each half of the intensity vectors was scanned for its first zero, which corresponds to the background voxel next to the surface in one of the directions of the ray. The elements after the first zero were removed, and the Euclidian distance between both ends of the trimmed path was calculated and saved in the same cell. The shortest ray was found by locating the path with the minimum Euclidian distance, and saved as

[7] A MATLAB container able to save vectors with different lengths.

the 1D ray core. The center point of the path was located, and the process was repeated for the 2D (composed by all coplanar rays of the 3D core) and 1D cores. After estimating the center point of the spine's cross section, the cross product between the shortest rays of both the 3D and 2D Rayburst was calculated, to determine the direction of tracing. A unit vector along this direction was then added to the center point, to determine the next position where the Rayburst should be applied, repeating the process. The algorithm is exemplified in Figure 3.25, on one of the spines belonging to the selected region of interest.

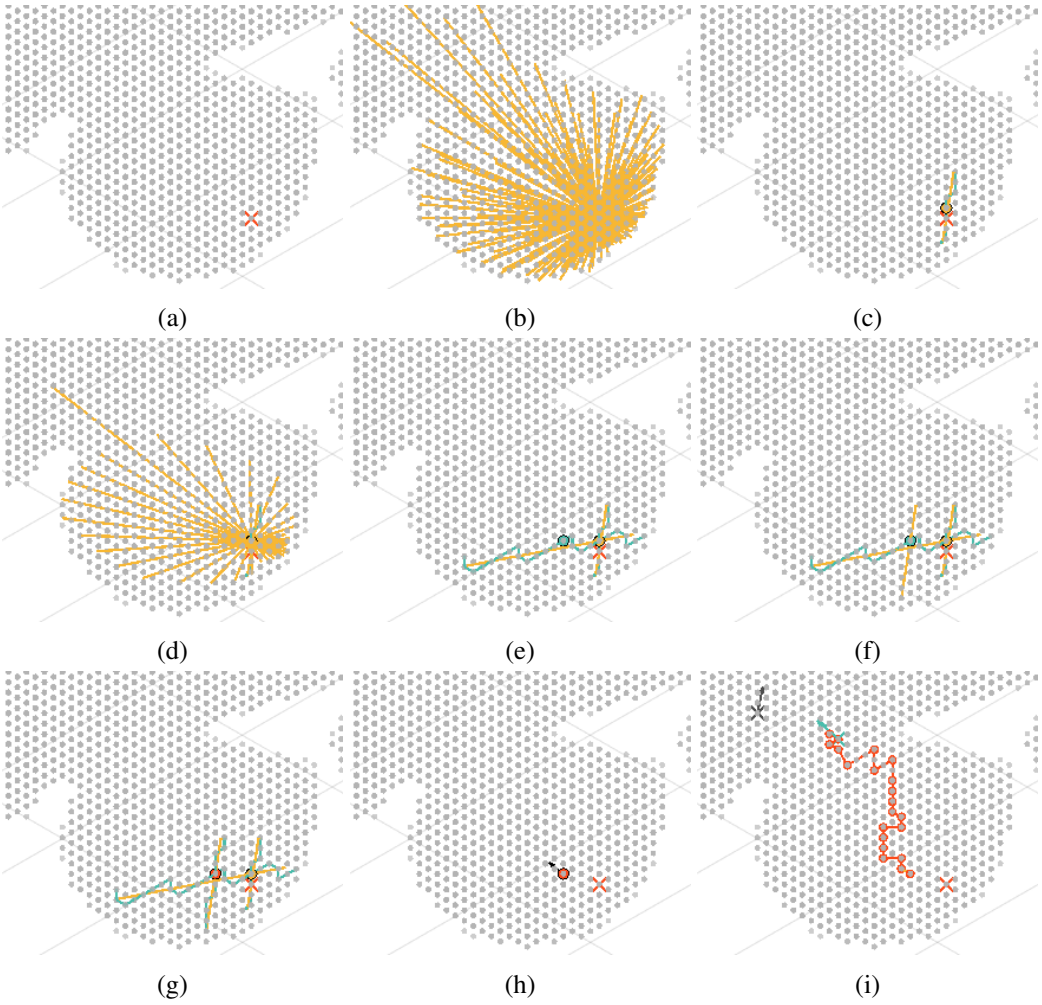


Figure 3.25: The proposed Rayburst algorithm applied to one spine: a) - the initial position calculated by the detection module; b) – The 3D Rayburst cast on the initial position, shown with continuous rays to improve the visualization; c) – The shortest continuous (yellow) and discretized ray (blue) of the 3D Rayburst and its center point (yellow); d) – The 2D Rayburst cast on the 3D Rayburst's center point; e) - The shortest continuous (yellow) and discretized ray (blue) of the 2D Rayburst and its center point (blue); f) - The 1D Rayburst cast on the 2D Rayburst's center point; g) - The single continuous (yellow) and discretized ray (blue) of the 1D Rayburst and its center point (red); h) – The final point with the direction of tracing (black arrow); - i) The final point of the second chain of Raybursts connected to the first; j) – The final tracing path with the estimated neck point and its direction along tracing (blue cross and arrow), and the last tracing point and direction classified as not belonging to the spine (black cross and arrow).

The direction of tracing, represented by the black arrow of Figure 3.25 h), is calculated by the cross section between two shortest rays. Depending on the order of the calculation, the resultant vector may point to one of two opposing directions. The direction leading to the neck of the spine is not known by the algorithm beforehand, so two different tracings are carried per spine, each in one of the opposing directions. The double tracing allows the algorithm to not depend on any user-defined initial vector, as it is usually required [47], having the initial seed as its only requisite. The final tracing is shown in Figure 3.26[8] [9].

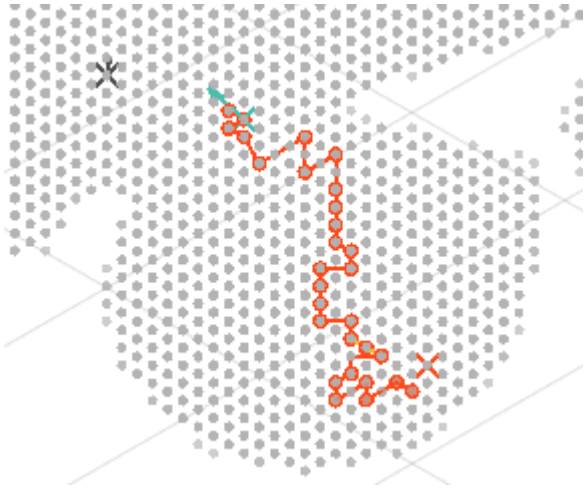


Figure 3.26: The completed double tracing for a chosen spine.

The aim of the tracing phase it to find a point close to the neck of the spine, so that a border can be marked and the spine can be distinguished from the dendrite by being on one side of the border. Therefore, the tracing phase must properly stop when it reaches the region of the neck.

The stopping conditions are several and encompass different scenarios which can arise during tracing. The tracing cannot leave the mask bounds for instance, and will stop when a background pixel is encountered. The same is valid for the image bounds. The tracing is also limited by a number of steps. (set to 15). This condition is useful to reduce computational costs when the tracing begins on a spine point incorrectly detected inside the dendrite shaft. The last two conditions regard the detection of a neck point, and were defined as the distance between consecutive tracing points, and the angle between consecutive moving directions. When one step performed by the tracing is larger than a predefined threshold (set to 3.6 voxels, more than the tripe of a normal voxel step), the tracing stops since it assumes the step was made to the outside region of the spine,

[8] The final tracing path was chosen not to be connected to the initial position as a visualization preference. Since the tracing steps begin in the point calculated by the first Rayburst chain, the path was chosen to only be represented from that point on..

[9] The Rayburst can also be used to estimate the topology of a structure with the aim of reconsctruting it. For each tracing point, the shortest rays were saved and used to form ellipses fitting the cross section of the spines. Each ellipse was then connected by a meshing strategy which generated a fixed number of points on the circumference of the ellipse and united them with triangular faces. The results for the selected region of interest are presented in the Appendix. In spite of approximating the surface, the meshes were deemed inaccurate, which led to the method described in this section.

where the shortest rays will be drastically larger than the previous ones, and the consecutive re-centering steps performed on each ray may detract the tracing point from the path. Alternately, a large difference of tracing directions (set to 45º or more) is also interpreted as the entrance on the dendritic shaft, since its axis is nearly perpendicular to the axis of dendritic spines, and may deviate the orientation of the shortest rays. The conditions were based on two hypothesized scenarios that could indicate the departure from a spine. These are illustrated in Figure 3.27.

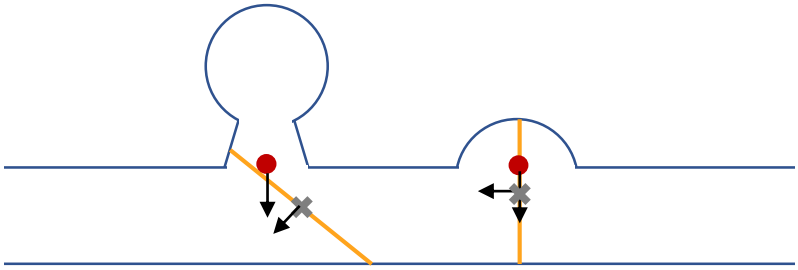


Figure 3.27: Two scenarios where the tracing point leaves the spine. The gray crosses represent the last tracing point, the yellow lines the shortest ray of the second Rayburst, the red circles the second to last tracing point, and the black arrows the tracing directions.

As shown in Figure 3.27, the conditions consider two distinct tracing behaviors which depend mainly on the length of the spine. In the left case, the second Rayburst of the last tracing point casts its shortest a ray between the neck (or its neighborhood) and the dendritic wall, diagonally to the previous moving direction. The ray cannot reach closer regions of the wall since the spine tip, in their opposing direction, is too distant from the tracing point. Hence, the point will be centered far from the previous, while the direction will be moderately tilted. In the right case, the short length of the spine enables the shortest ray to terminate in a closer region of the wall, positioning the tracing point close to the previous but with a drastically different orientation.

In every step of the tracing, this set of conditions is verified and if one is violated, the process ceases. Still, because two tracing paths are completed for each spine, it remains to know which one ended on the neck. To determine it, the conditions that were violated by both are compared and the path which infringed the ones most typical of a spine departure is chosen. Additionally, the condition analysis also serves to discard tracings that did not reach a spine's neck, or encountered a condition that invalidated both tracings, such as reaching the image bounds. The actions held by the algorithm for each pair of violated conditions is organized in Figure 3.28, where arrows point to paths deemed to have ended on the neck, based on its infringements.

The simultaneous infringement of the angle and distance conditions was considered the most typical of tracings leaving the neck, which should correspond to similar scenarios to the one presented in the left part of Figure 3.27. It is chosen between all combinations except when the other path reached the image bounds, which indicates an incomplete spine. For the same reason, the image bounds infringement invalidates every other path independently of its infringement. Angle and distance are interchangeable, and the path is chosen according to the longest distance travelled between both. The step limit was privileged against angle and distance to account for long spines that had one path terminated erroneously, and the other not reaching the neck due to insufficient

	Angle & Distance	Angle	Distance	I. Bounds	M. Bounds	Steps
Angle & Distance	**longest**					
Angle	↑	**longest**				
Distance	↑	**longest**	**longest**			
I. Bounds	✖	✖	✖	✖		
M. Bounds	↑	↑	↑	✖	✖	
Steps	↑	←	←	✖	←	✖

Figure 3.28: Decision table based on tracing infringements

number of steps. Such situation is depicted in Figure 3.29 (left). Steps are also preferential to the image bounds condition for long spines where the seed point is near the tip (promoted by the detection module), shown in Figure 3.29 (middle left). If both paths terminate by the step limit, however, neither is chosen, since they most probably traveled through both directions of the dendrite shaft, due to an incorrectly marked seed point (Figure 3.29, right). Finally, if the tracing was made in a direction perpendicular to the axis of the spines, and both paths terminate by mask bounds, the tracing is discarded as well (Figure 3.29, middle right). These scenarios approximate the reasons which make the tracing stop, but may not always be verified for every image and region. The table can therefore be adjusted to account for the most frequent reasons associated with the end of tracing in the neck region.

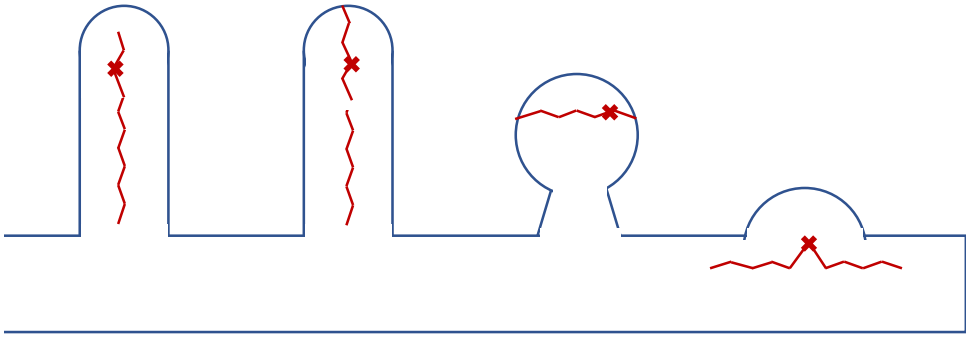


Figure 3.29: Atypical tracing stopping scenarios handled by the condition comparison.

The tracings were also made invulnerable to the angle and distance condition in their first two steps. Since the seed point may be anywhere inside the spine and lead to unexpected first steps both in distance and direction, this feature prevents cases similar to the one depicted in the left end of Figure 3.29,, while making the minor assumption that it is possible to give at least two steps inside the spine from its seed point.

After each tracing, a tracing report is issued to the console presenting the condition that led to its termination. For the paths ended by the distance and/or angle conditions, the values that

triggered them are also specified. Figure 3.30 shows the report for the spine of Figure 3.26.

```
Tracing Report - Spine 2
 > Forward path
  - Stopped by: Distance and Angle
    Distance report: 6.7823 <? 3.6
    Angle report: 52.9219 <? 45
 > Backward path
  - Stopped by: Mask bounds
>>
```

Figure 3.30: Tracing Report for the spine of Figure 3.36.

Once every double tracing has ended and the paths terminating on the neck are chosen, their second to last tracing points and directions are saved (or last in case the infringement was on the step limit) and carried to the delimitation process, described below.

Delimitation The delimitation of spines is performed by positioning a plane in the neck point, normal to its tracing direction. As the neck points and directions may not be perfectly positioned and oriented, the plane must be rectified to ensure all the voxels connected to one of its sides belong to a spine.

Geometrically, if a rectangle of limited length is positioned in an arbitrary point within a 3D image with both foreground and background, one of the following three cases will occur: (A_0) all of its border points will intersect the foreground; (B_0) some of its border points will intersect the foreground; (C_0) none of its border points will intersect the foreground;

In the neck region of spines, case (A) may occur if the rectangle is not large enough to intersect the surface of the dendrite. This case does not provide any information on the topology of the dendrite, since it can occur nearly anywhere in the foreground. If, however, the rectangle is made large compared to the diameter of the dendrite, case (A) is mitigated and the three possibilities are reduced to (B) and (C). Case (B_0) may arise from two different situations. If the plane intersects the dendrite longitudinally, its opposite border points will share the same approximate behavior, either by being both on the foreground (the right and left side of the rectangle, for instance) or outside the foreground (the top and bottom sides, for instance). If, however, the plane intersects the dendrite diagonally, and it is not large enough to surround it, more pixels of one side will intersect the foreground than the other. Case (C_0) occurs when the plane is positioned transversely inside the spine. It can also take form when it's positioned transversely to the dendrite, although this is rare since the neck points from the tracing that passed the condition analysis are seldomly inside the dendrite and oriented along its axis.

Since case B_0 may correspond to two different conformations, it was divided in case A and B, which correspond to the intersection of one border (approximately) with the dendrite, and the intersection of two borders (approximately) with the dendrite, respectively. Case C_0 was retitled to C. For each case, an algorithm was written to adjust the initial plane into the transition between

dendrite and spine, facing the spine. Figure 3.31 shows a schematic of each case and of the sequence of steps to perform accordingly.

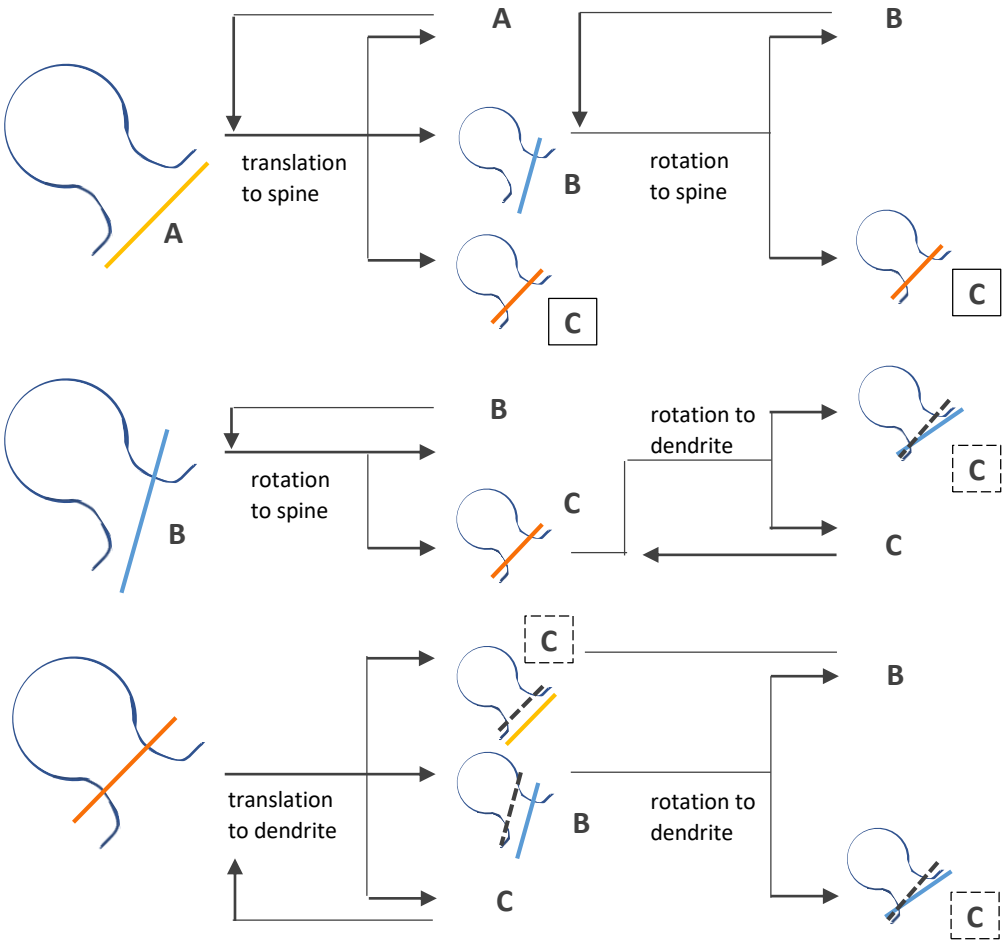
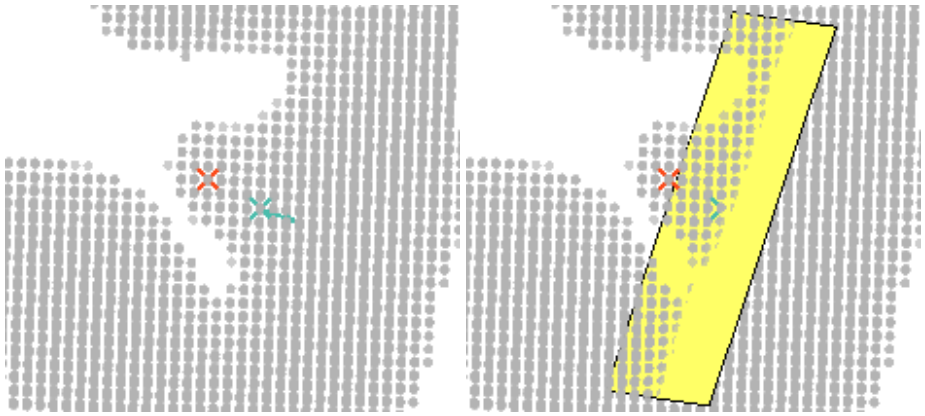


Figure 3.31: The plane rectification process. Cases (A), (B) and (C) are represented by letters and drawings if different from the last. The possible results are denoted with a contour around the letter. Dashed contours indicate that the result corresponds to the plane of the previous iteration, also drawn with a dashed line. The transformation between planes is written.

After the plane is placed in the tracing point, the case it corresponds to is inspected. In 2D, as shown in Figure 3.31, the cases can be distinguished by the position of the line ends alone. In 3D however, the borders of the plane have to be analyzed to determine if they are evenly intersecting the structure, or if the intersection is prevalent in one side [10] of the rectangle. Figure 3.32 exemplifies the positioning of an initial plane in one of the neck points of the region of interest.

To analyze the intensity profile of the rectangle, one slice with the same size, position and orientation is extracted from the image. The extraction is held by an opensource function named *extractSlice* and credited to Pangyu Teng. The function extracts specific voxels and turns them into a 2D matrix of intensities and coordinate indexes. The code was modified so that the top values of

[10] By side, what is meant is not an actual edge of the rectangle, but a broad direction from its center point. The intersection can be found on one corner for example, referred as one side of the shape.



(a) (b)

Figure 3.32: Positioning of the initial rectangle in the neck point of a spine.

the matrices correspond to the top region of the slice in the 3D volume, which was not always de case.

After the extraction, only the central connected component is kept from the intensity matrix, corresponding to the cross section in contact with the neck point. This allows the rectangle to have any length above the diameter of a spine, since any structure unconnected to the neck point, and thus unrelated with the spine in question, is discarded.

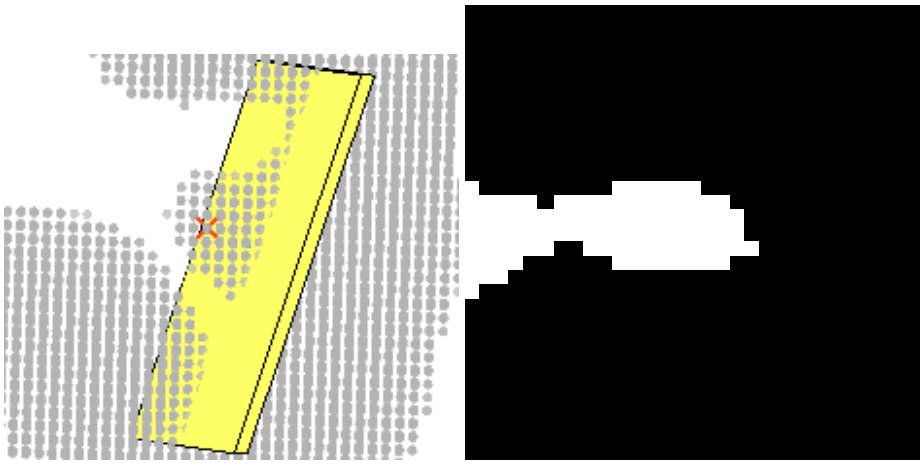
To determine if there is any kind of symmetry between the edges of the slice, all pixels outside the edge are removed (by intersecting the image with a binary square ring), and the centroid between the edge points is calculated. If the centroid falls into a central square, with 3/5 of the length of the slice, the border points are deemed symmetric and the plane follows case (A). If the centroid falls into the 1/5 of the slice closer to an edge, the border points are considered asymmetric and the plane follows case (B). If the slice does not possess border points, then the plane is classified as case (C). Figure 3.33 shows the slice, border points and centroid for the example of Figure 3.32.

(a) (b) (c)

Figure 3.33: The calculation of border symmetry in the extracted slice. A) the slice; b) its border points; c) their centroid and the translational region

Since the centroid falls into the central square, Figure 3.32 depicts a case (A), where the plane

is located inside the dendritic shaft. As so, the first action is to translate it in the direction opposed to the tracing, leading it to the spine. For every translation, a new slice is extracted and the border symmetry is verified to check if the plane transitioned to another case. The first translation is shown in Figure 3.34, along with the extracted slice.



(a) (b)

Figure 3.34: First translation. A) The new plane in the 3D volume; b) the extracted slice.

Since the border points are distributed asymmetrically along the edge, the plane has transitioned to case (B), and the translation process stops. The border asymmetry indicates that the plane is not intersecting the spine equally, and therefore has to be rotated. To do so, a rotational axis is calculated in the extracted slice, and positioned in the correspondent site of the 3D plane. The calculation begins by finding the cartesian line equation between the border centroid and the center point of the slice. As the centroid estimates the region where the plane intersects the dendrite, the rotation will be performed to carry the centroid in the direction of the spine and align both sides of the plane. In the opposite side, the region intersecting the spine correctly should not move, and hence the rotation axis should be positioned adjacently to this intersection, making this region nearly static while rotating the centroid region. For every pixel, the subcoordinates of their isolated corners are calculated, and the corner further away to the centroid is determined. The rotational axis is positioned in this corner, perpendicular to the line whose equation was found before, and the vector connecting the central point and the closest point to the axis (rotation arm) is calculated. The coordinate matrix outputted by the extractSlice function is used to create a local coordinate system for the slice within the global coordinate system of the image. The transformation matrix between spaces is calculated, and used to orientate the rotational axis and arm in the 3D space. Since the center of the slice is the translated neck point, the vector arm is cast from this point in the 3D space, yielding the position where the rotational axis will the centered. The centroid, central and corner points, as well as the rotational arm and axis are marked in Figure 3.35 for the previous slice.

Each rotation was set to 3º degrees, after which a new slice is extracted to check if the plane

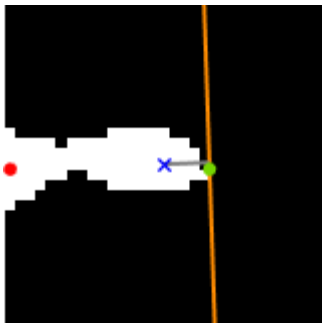
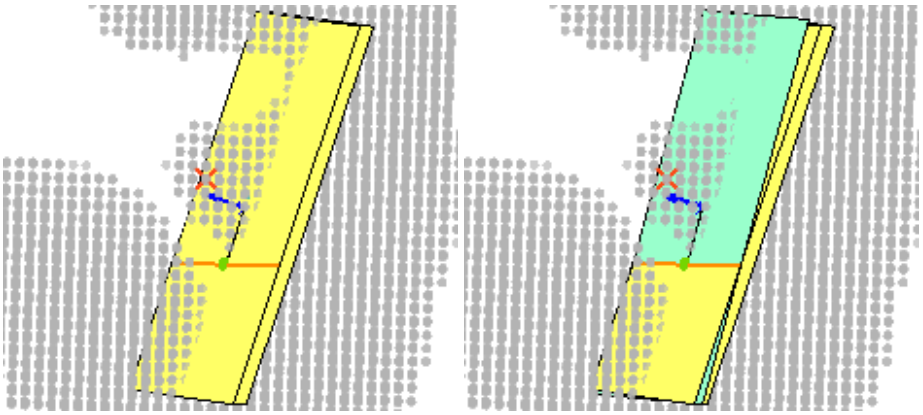


Figure 3.35: The major points and vectors calculated for the rotation of the plane. The centroid, central and corner points correspond to the red point, blue cross and green point, respectively. The arm vector is shown in gray, and the rotational axis in orange.

transitioned from cases. The rotation in the 3D space [11] is shown in Figure 3.36, with the same elements of the slice of Figure 3.35 (minus the centroid). The slice coinciding with the rotated plane is shown in Figure 3.36.



(a) (b)

Figure 3.36: Rotation of the plane in the direction of the spine.

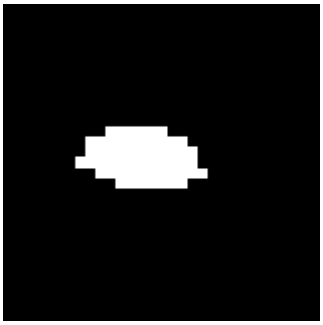


Figure 3.37: The slice extracted after the first rotation.

[11] The rotation is applied on the arm vector around the rotational axis. This operation is achieved by a publicly available function named *rodrigues_rot* which is credited to Ismail Hameduddin.

Since there are no more border points, the plane is only intersecting the spine neck, specifically in the region immediately after the dendrite shaft. Therefore, the rotation process stops, and the center position and orientation of the last plane are saved. The final plane and normal direction is presented in Figure 3.38.

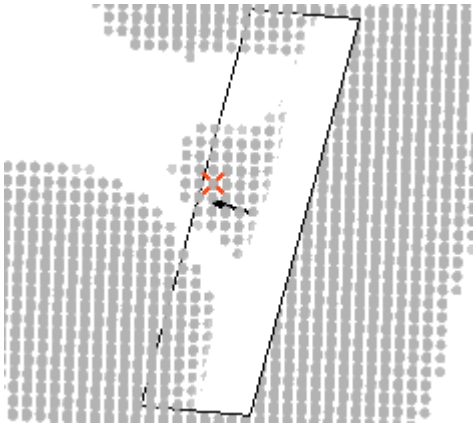


Figure 3.38: The final plane positioned in the proximal region of the neck towards the spine.

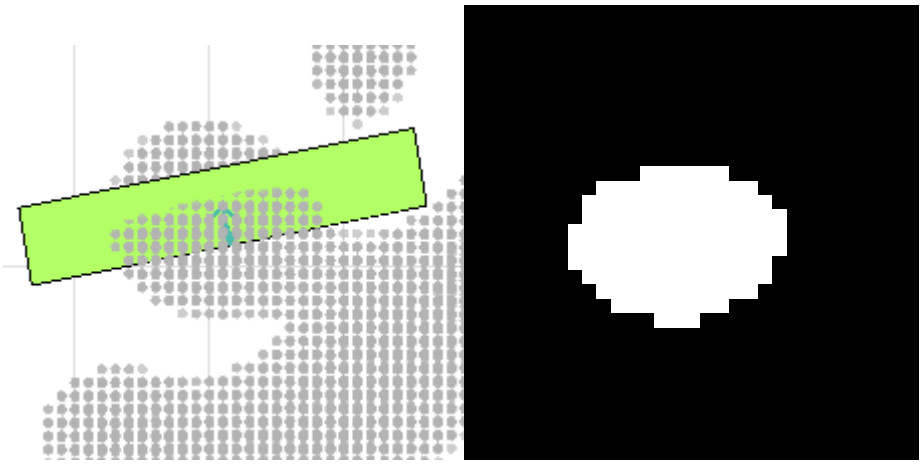
The sequence of transformations followed Figure's 3.31 strategy for planes beginning in case (A). For these cases, there is a shortcut to the final plane that is triggered if the final translation gives rise to a plane of case (C). If so, as the translation steps are small (1 voxel), the translation phase ends making a perfect slice, without the need of a rotation step and thus ending the process.

The strategy handling initial case (C) planes has opposite direction translations and rotations to the one exemplified, but features two new characteristics, one of them hinted by the dashed dots surrounding some plane letters. Hence, it will be demonstrated as follows. The strategy for initial case (B) planes is an intermediate version of both other strategies and will not be exemplified to avoid redundancy.

In an initial case (C), the plane is already surrounding the spine, although with an expected orientation and location. Figure 3.39 exemplifies this scenario with a 3D plane and the correspondent slice.

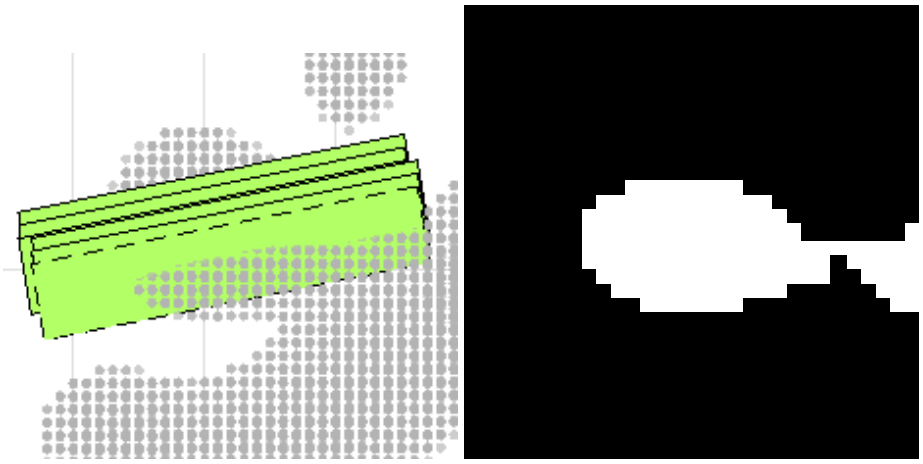
Since the distance from the dendrite shaft is unknown, the first action to apply on the plane is the translation along the tracing direction, until it becomes either a case (A) or (B). The sequence of translations are shown in Figure 3.40.

From the symmetry analysis of the slice of Figure 3.40, the last translational plane is categorized as a case (B). If it were to be rotated directly as in the last example, the edge intersecting the dendrite would never leave the shaft, since it would correspond to the rotational axis. The rotational axis must be always outside the surface of the dendrite or spine, the prevent the cross section of expanding to shaft territory. Therefore, when the plane is adjusted from the outside of the shaft, the last plane is discarded, represented by the dashed contours, and the rotation process is applied to the one before it. As the rotation axis requires an edge centroid to be calculated, and the previous slice does not possess any border points since it is a case (C), the edge centroid of the discarded plane is used to this effect. This allows the rotation to proceed in the correct direction,



(a) (b)

Figure 3.39: Initial case (C) plane (a) and extracted slice (b).

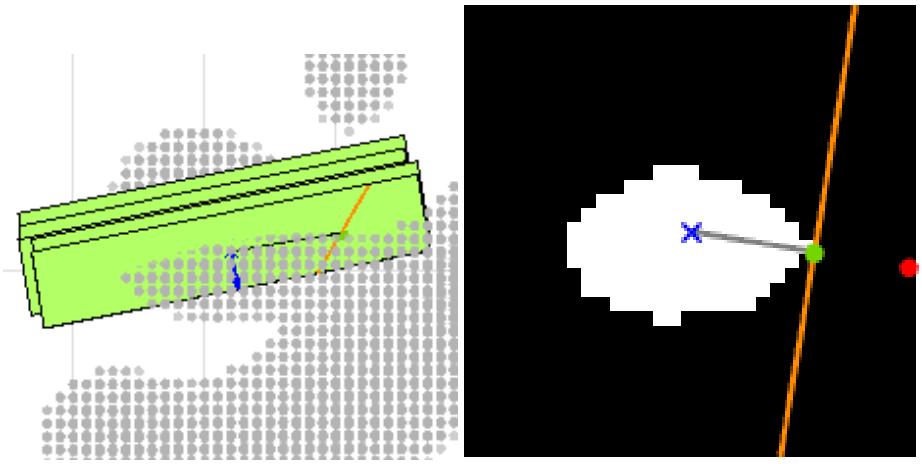


(a) (b)

Figure 3.40: The translational process applied to the initial plane: a) the plane instances; b) the slice of the last plane (dashed).

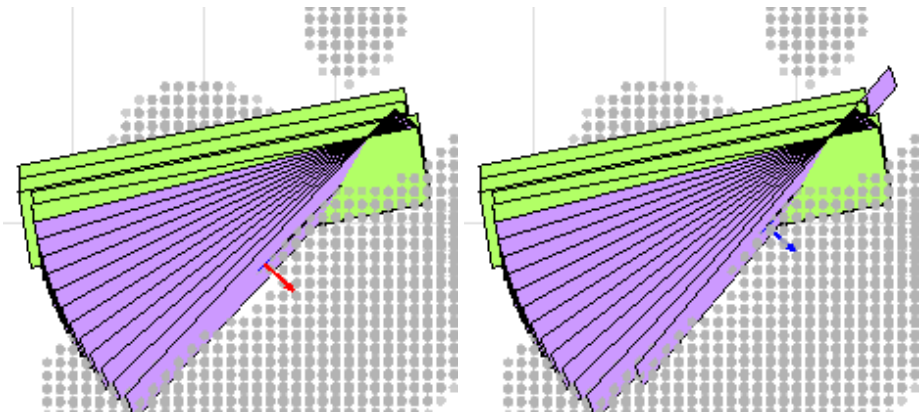
on a plane with insufficient information to determine it. Figure 3.41 shows the rotational apparatus on both the plane and slice.

For each rotation, a slice is extracted to evaluate if the plane transitioned to a case (B). The asymmetry analysis is based on the central component of the slice, which can be compromised if the center being rotated leaves the foreground. This is particularly uncertain for long rotational processes, such as the one being exemplified. To prevent an escape from canceling the delimitation, when the intensity of the central point turns to zero, the rotational arm is set to half of its previous length. As the escaped central point and the rotational axis limit the cross section of the spine, halving the rotational arm will adjust the central point exactly to the middle of the section, and allow the rotation to proceed its course. Figure 3.42 depicts this event and the response by the program.



(a) (b)

Figure 3.41: The plane (a) and slice (b) prior to their first rotation towards the dendrite.



(a) (b)

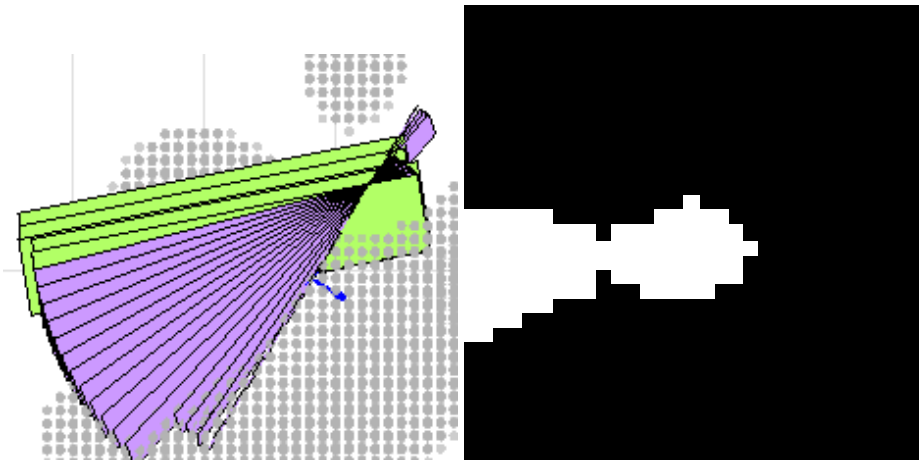
Figure 3.42: Adjustment of the plane triggered by the escape of its center point: a) – the rotational plane centered around a background pixel; b) – the corrected rotational plane, re-centering it to the foreground.

With the rotated point re-centered, the process continues until it reaches case (B). This transition implies that the side being rotated towards the dendrite has passed its surface by one rotational step. This event is depicted in Figure 3.43.

As so, the last plane is discarded, similarly to the last translational plane, and the position and orientation of the one before are saved as the final boundary. Figure 3.44 presents the final plane and slice.

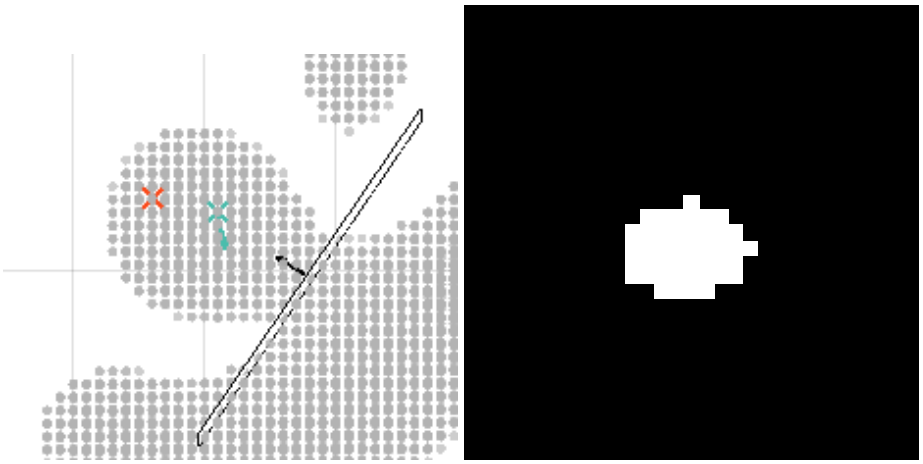
After the final plane is positioned, a rectification report is issued to the console, outlining the number of transformations and transitions the initial plane went through. The report for the last example is presented in Figure 3.45.

The final planes are saved as a pair of position and direction vectors, and the data is carried to the spine extraction phase.



(a) (b)

Figure 3.43: Last rotational plane and slice.



(a) (b)

Figure 3.44: Last rotational plane and slice.

```
Retification Report - Spine 9 (neck 1)
Slice  Types: C(B)Ce(B)C
Translations to spine: 0/30
Translations to dendrite: 5/30
Rotations to spine: 0/30
Rotations to dendrite: 15/30
>>
```

Figure 3.45: Rectification report for the illustrated C case. The upper case letters represent the cases experienced by the planes, and the lower case 'e' the escape of the center of the slice. The transformations are shown along with their number and limit.

Extraction In this phase, the spines of the masks are segmented with a region growing method, limited by both the mask and the planes. A function named *regionGrowing* and shared publicly

by Daniel Kellner was used and modified to consider both plane points and directions previously calculated. This routine starts by saving the seed point into a vector and checking if its direct neighbors fulfill a set of conditions. The seed point is removed when its neighbors are checked and the ones that pass the conditions are added. These are also marked as white voxels in a matrix with the side of the image. The process repeats for each point present in the vector, until it becomes empty. The conditions examine if the points belong to the image limits, if they were already considered, and if their intensity is within a certain with respect to the seeding point. The range was set to a value below one, since the routine is being applied to the foreground of a binary mask. To check if the points belonged to the side of the plane facing the spine, the dot product between the normal of the plane and the vector connecting the plane to a considered point was calculated. The positive sign of the result implies the point is on the correct side. As the spine points from the detection phase can be on a different side of the plane than the actual spine, as shown in the example of Figure 3.32, the seed point is instead calculated as the voxel attached to the center of the plane, in the direction of its normal. A maximum distance to the seed is also checked, to prevent over segmentations in cases the planes are not positioned correctly. The algorithm runs iteratively for every delimited spine, and produces a binary matrix containing all spines. This matrix is shown in Figure 3.46 for the chosen region of interest.

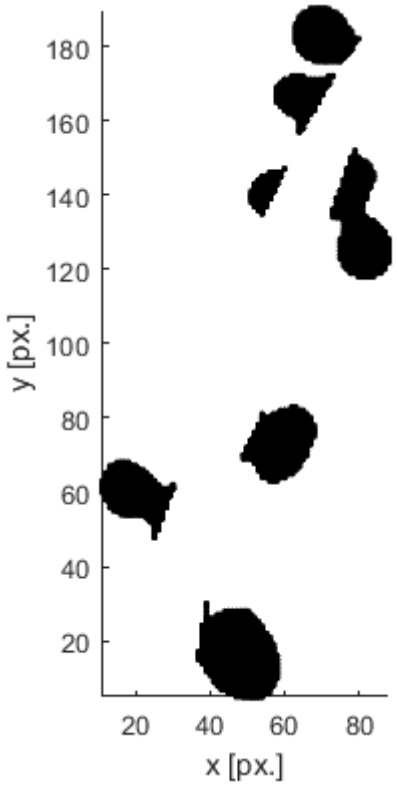


Figure 3.46: Extracted spines by region growing.

The matrix can then be corrected in the shrinking phase, or passed directly with the neckpoints to the Measurement module. The shrinking phase is described in the following section.

Shrinking This phase aims at reducing the size of the extracted masks. These are unaltered parts of the Hessian_stack, which overestimated the location of voxels belonging to dendrites in the original image. To correct them, they were intersected with the median stack delivered by the Preprocessing module, which yielded their intensity profiles.

In confocal miscroscopy, one of the main noise sources is Poisson (or statistical) noise. This is proportional to the number of photons recorded in a given pixel, which in turn correlates to intensity [48]. It has also been suggested that the size of errors introduced by confocal imaging make up 5–7% of the mean intensity values of a given region [49]. Based on these considerations, a relative threshold was set to 2% of the maximum intensity found inside a given mask. The low value accounts for the image being already median filtered, and serves to exclude the zero pixels inside each mask. The threshold was not applied directly to the spine volume, but as a condition in the region growing routine previously introduced. This allows it to be applied only on voxels connected to the seed point, preventing a result containing isolated voxels. However, it was observed that the grayscale region growing may degrade the masks irregularly in some regions, producing results more atypical than the ones from the extraction phase. Since it does not account for any shape descriptors, as the Hessian analysis did (with the second derivatives), the size reduction does not seem to compensate the regularity loss of the Hessian mask, and so it was turned into an optional feature. It is also reasonable to make this process facultative to allow other masks to be used in the program, without altering their morphology by default. In either case, the final spine masks and neck points are then passed to the Measurement module for their quantification.

3.2.1.5 Spine Measuring

In this module, each spine mask is identified by the *bwlabeln* command, and iterated over to calculate an individual set of measurements. Spines are plotted in 3D with a number on his top, and the measurement data for each is compiled into a table, linking each row with the number of each spine. The set of measurements is the following:

- Volume
- Surface Area
- Solidity
- Centroid proximity

Volume is equal to the number of voxels of each spine, and thus evaluates its size, which in turn is correlated with synaptic strength. The surface area can be used with the volume to estimate the resemblance of spines with certain shapes, and aid in the classification of their type. Solidity is defined by the ratio of voxels between a structure and its convex hull (the minimally convex volume encapsulating the structure). It is a measure of concavity, and can be used to evaluate the length and thickness of the neck in comparison with the head. The centroid proximity is calculated

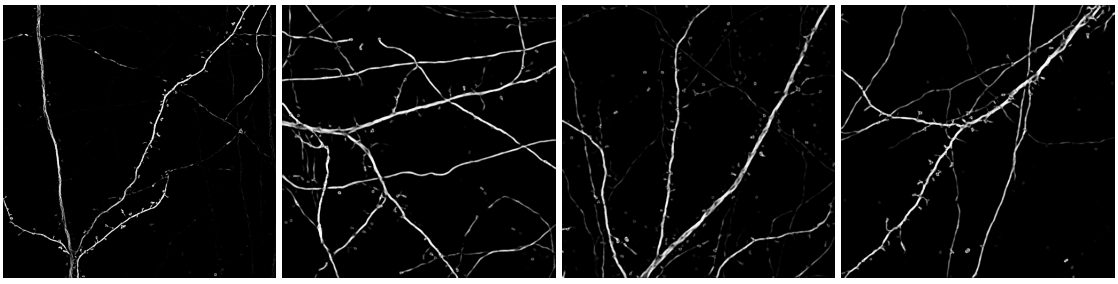
with respect to the neck, using the plane points from the previous module, and it offers a way of evaluating how far the head, typically including the center of mass of the structure, is from the dendrite shaft.

The first three measures are directly calculated by MATLAB's command *regionprops3*. The centroid is also calculated by this command, and the Euclidian distance is taken between it and the plane point, marking the beginning of the neck.

All results are presented in the Results and Discussion section of this work.

3.2.2 Machine Learning 2D Algorithm

The 2D image data used in the development of the supervised detection method was created by applying a maximum projection to four 3D image stacks. The images were then preprocessed using the tubular response similarly to what was detailed in the last section, although applied in 2D to adapt to the data format. These preliminary steps were done in MATLAB, and the results are shown in Figure 3.47.



(a) (b) (c) (d)

Figure 3.47: Preprocessed 2D dendritic images for the Machine Learning detection algorithm

The algorithm designed to identify the spines in these images uses a Convolutional Neural Network (CNN), a variant of Deep Learning networks capable of extracting features automatically from image data and learning the underlying patters of the structures required to be identified. The program was developed in Python, using the Anaconda distribution, and the network was built using the Keras library run on top of the GPU version of the Tensorflow backend.

As required by supervised methods in general, a dataset containing images both with and without the target structures has first to be created. Spines occupy small regions of the available images, and so the dataset for this method is formed by equally sized windows extracted from the four images. The dataset is then used to train the algorithm to identify spines and evaluate the performance on a part of the dataset not used for training. After the training, a model is generated in .h5py format, and used to predict the presence of spines in new windows in form of a probability. Since the aim of this method is to identify spines contained in the images with both spine and non-spine material, the model is used iteratively across every pixel of the preprocessed image, to evaluate if a neighborhood, of the same size of the dataset windows, contains spines. The output is an array of probability values, which is then reshaped into a grayscale image with

the size as the original. A post-processing step is then applied to threshold the image and eliminate some noise created by the classification phase.

3.2.2.1 Creation of the Dataset

The creation of the dataset is done by manually classifying small windows from the original images. This process can suffer from the subjectivity of the evaluator, since their location can be chosen to facilitate the classification and disregard more ambiguous regions which will be encountered by the algorithm in the prediction phase. If the choice of the location was to be random, however, as spines are in a very small proportion compared to the rest of the image, the creation of the dataset would take impractically long periods of time and the majority of images would correspond to the background of the dendrites. To make the image annotation impartial, yet producing relevant information in the shortest amount of time possible, a method resembling an interactive quiz was devised, which selects relevant windows from the images and presents them to the user. The user is then invited to choose to classify them as spines, non-spines, to pass if he can't determine confidently which class to attribute, or to exit the process. The quiz is illustrated in Figure 3.48, in the console of the Spyder IDE.

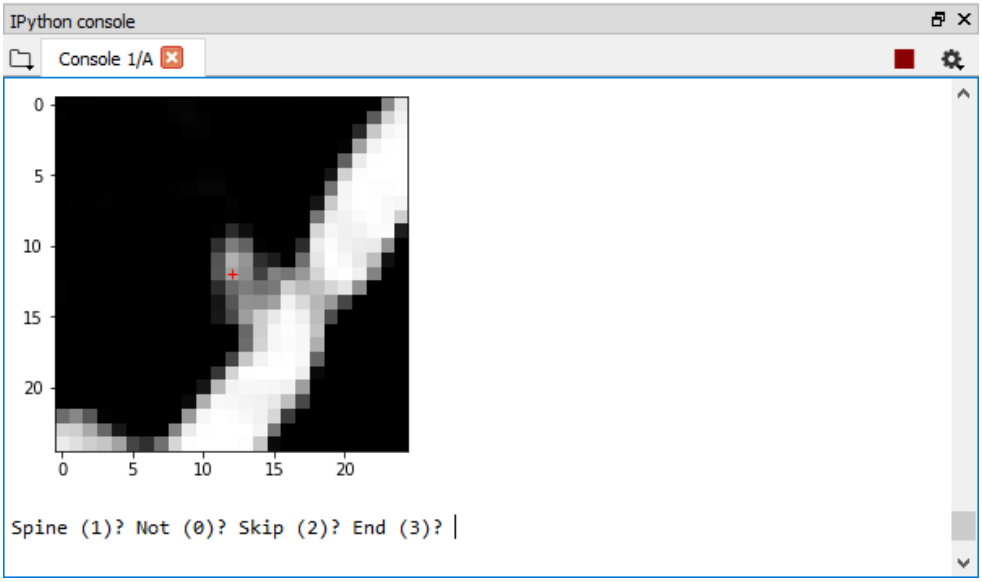


Figure 3.48: Interactive quiz for the creation of a dataset.

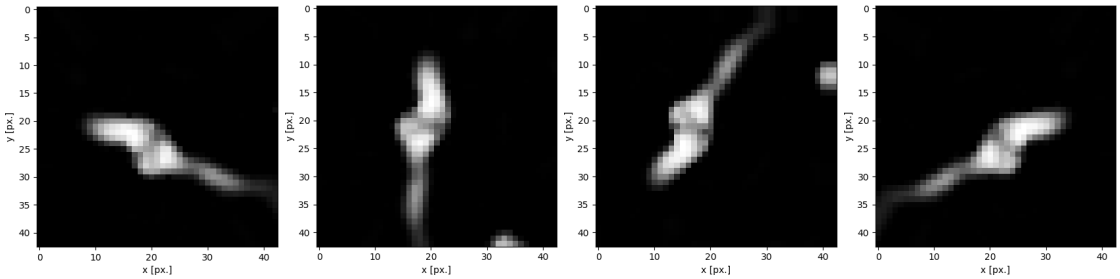
The algorithm begins by loading the preprocessed images and applying a threshold of 0.1 (for the images of type double and range [0,1]) to locate their pixels above and below this level. As the images are the outputs of a tubular response, their signal-to-noise ratio is high, which makes the pixels above the threshold to most probably belong to the dendrite structure. The coordinates of both high and low intensity pixels are saved into separate arrays. The program then chooses one coordinate from one of these arrays, with a strong bias (0.9 to 0.1) to the high intensity pixels, to increase the chances of capturing a dendritic spine. A window of size set to 25 pixels is then

extracted around the chosen location, and presented to the user, which is requested to classify it. Depending on the input, the program either saves the image into a 3D array and indicates its class on a binary vector of the same length, or ignores it. If the user chooses to ignore it by skipping, this process is repeated, otherwise it stops. In this case, since the windows classified as spines will be less than other cases, even with the intensity bias (since dendrites shafts are predominant above the threshold), the algorithm deletes a number of non-spine dendrites so that the final arrays for both classes have equal length. Then, each array is split into two parts, one spanning ¼ths of the length and the other ¾ths. The longest arrays are saved for the training phase, while the shortest for the testing phase. The arrays are saved in *.npy* format, to be used either by the Replication module or directly to the Training and Testing module.

3.2.2.2 Replication of the Dataset

Neural networks often require great amounts of data to yield reliable results (some reaching millions of samples [50]). Since the dataset is composed of four images with a limited amount of spines[12], the classified arrays can be expanded artificially to enhance the performance of the algorithm. This was done with a Keras class named *ImageDataGenerator* which, based on parameters (such as rotation range, resize level, and others), produces a defined number of randomly transformed copies of a given image. The only transformations allowed in this phase were rotation, set to range of 360º, and random horizontal flips. Size was not changed and the window was not translated since the center could escape the target structure. Although the horizontal flips simply alter the disposition of the image, rotations create artificial zero-valued regions around the corners, which would be problematic as they do not depend on the structure being targeted and would be nonetheless learned in the training phase. To solve this issue, although the window size presented in the quiz is set to 25 pixels, the actual windows saved are extracted from the image with a higher length. In the Replication phase, these windows are rotated randomly by the generator, and cropped around the center, yielding a final window of 25 pixels. The extra length is calculated such that a 45º rotation and the following cropping do not create any zero-value pixels. The replication number was set to 30 copies after experimentation. The augmented dataset was then saved similarly as before, to be loaded in the Training and Testing phases. An example of three transformations made on one of the windows is shown in Figure 3.49.

[12] From the four images used, 70 spines were collected. Neural networks performance typically scale with the amount of data used, which can reach hundreds of thousands of samples. The replication step was written to remedy the absence of more images, although it should not substitute additional original images in case they are available.



(a) (b) (c) (d)

Figure 3.49: Original window (a) and three replications from the thirty generated

3.2.2.3 Training

In this phase, the data is used to teach a neural network how to identify spines on images with the size of the windows generated. A convolutional neural network is a sequence of filters and routines that are applied to an initial image, to yield a probability score of how similar it is to a certain class. The network has dynamic parameters, called weights, that are repeatedly convolved with the initial image and its transformations along the network. The weights are organized into small matrices called filters, and slide through the image multiplying each of the region values with the correspondent weight. After each slide, a new image is formed, and multiple filters are often used to form stacks of multiple images. The stacks can then be passed through an activation function, which alters their elements based on their value. The activation function is typically a non-linear function, to increase the possible element combinations after the convolution phase. After the activation phase, the matrices are normally subjected to a pooling phase, in which their elements are sampled following a predefined rule, and the matrices are reduced in size. This process repeats to form deeper nets, which are considered to capture more complex patterns present in the images. When the network reaches the last pooling layer, the matrices are converted into an array, and passed through a different activation function, which outputs a value for every class of the classification problem. The values are considered to express the similarity between the image and each class. As each image of the dataset has a label identifying its class, the calculated values are compared with the values from the label by a loss function (the label takes the form of a binary vector, populated with zeros except in the position correspondent to its class), and a disparity measure (loss) is determined. An optimization routine then takes this measure and variates the weights of the convolutional phases to lead the output of the next convolutional process into a value with lower loss. The variation is often applied in the direction of the larger loss decrease, which is achieved if the optimization method is set to Gradient Descent. Each convolutional process is referred to as an epoch and usually, the higher their number the more accurate are the estimation in the training images. This can be disadvantageous, however, since the network can become too familiarized with the dataset and produce unpredictable results in newer data. This is what is termed by overfitting, and strategies to reduce it include decreasing the number of epochs, blocking a set of weights from varying (called dropout), or dividing the training set to produce a validation set and selecting the networks which perform well on the validation

set instead on the training images. In this phase, different architectures were considered and one was chosen based on the results on the windows of the test phase, and on the visual analysis of the network's ability to predict the position of spines in the complete preprocessed images. The chosen architecture is depicted in Figure 3.50.

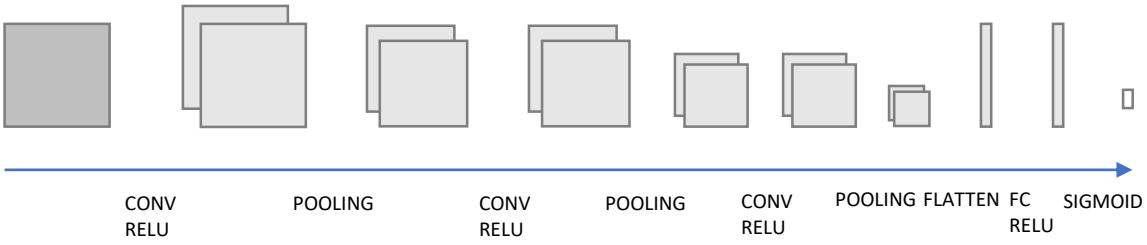


Figure 3.50: Architecture of the chosen network for the spine classification problem.

The number filters the three convolution steps (also referred to as layers) was set to 32, 32 and 128, respectively in its order (following the common practice of setting them to a power of two), and their shape maintained constant with 3x3 weights. Every activation function following the convolutions was chosen to be ReLu, which preserves the value of the positive elements while setting to zero those which are negative. The pooling phases followed the maximum value rule, downsampling the matrices by substituting the pixels from a movable region (set to measure 2x2) with their maximum value. After the last pooling phase, the matrix is converted into an array and each element is multiplied by weight to produce a reduces array with length equal to the number of classes. This layer is called fully connected layer (FC) as it uses all values from the array to calculate each probability value (connects all inputs to an output), contrarily to the convolutional layer that only considers local values for each calculation. The values are finally passed to the sigmoid function which outputs the score of the image for each class. To calculate the loss, the cross entropy is calculated, which is based on the logarithmic difference between the score and the label. The optimizer set to alter the weights that produced the best results was Adam, an approach to Gradient Descent which varies how much each weight is altered. The dataset was divided into a training and validation group, and the process until the creation of the model was repeated in a 2-fold cross validation. The network accuracy and losses for the training and validation datasets were plotted for each epoch, and considered in conjunction with the accuracy from the testing phase.

3.2.2.4 Testing

The testing phase uses the training model to predict the classes of the set of testing windows. The model is a Keras object and has a method named predict for this effect. The algorithm outputs an accuracy which represents its ability to classify spines in unknown windows of 25x25 pixels. After considering this results, the model was finally used to predict the location of spines in a complete image, which is performed by the Prediction module.

3.2.2.5 Prediction of spine locations

The projection of the same image presented in the geometrical method component was chosen to be subjected to the prediction capabilities of this algorithm. Recurring to the same class method of the testing phase, the model was used in each 25x25 pixel window of the image to output the probability result of it containing a spine. Instead of extracting a new window iteratively to predict its class, the windows were first grouped into a 3D *numpy* array, which fastens the prediction in exchange of requiring an extra space in the disk (around 1GB for the 1024x1024 image). The probability array was reshaped to be of the size of the preprocessed image, which is presented in Figure 3.51. It was then carried to a post-processing phase to be filtered and segmented.

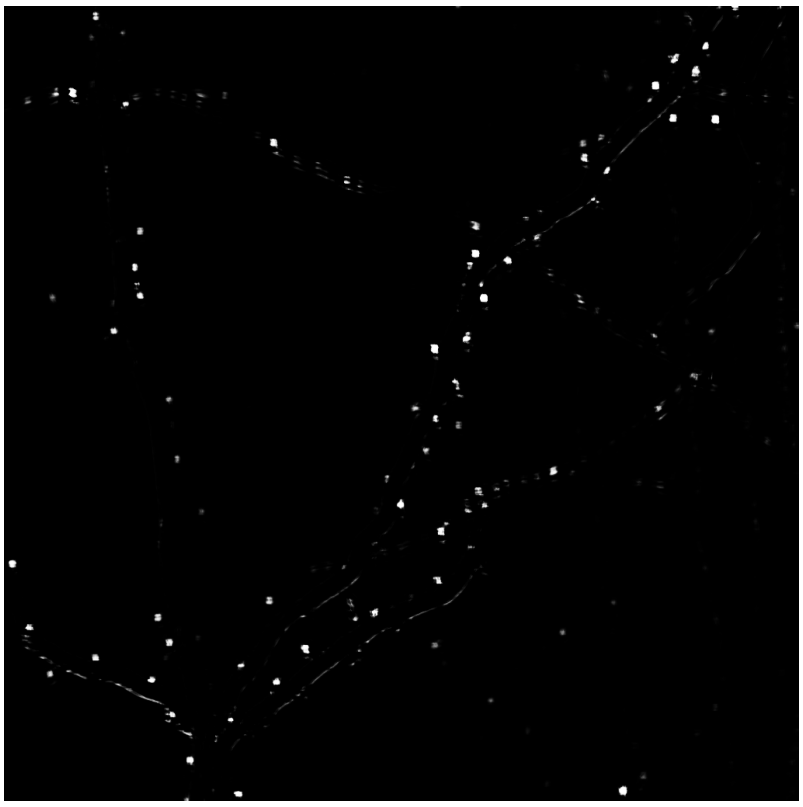


Figure 3.51: Spine predictions in form of probabilities, for the first image used in the dataset.

3.2.2.6 Post processing

To preserve the high probability regions of the scores image, a threshold slightly beneath half the image range was set (0.4 for the double type [0, 1] image). Then, the skimage.measure library was important to identify the connected components with the function label. By visual observation, the predictions image contains the identification of spines as circular objects, while some boundaries of the dendrites are also presented but with an elongated shape. Hence, the circularity of every connected component was measured by the regionprops function of the same library. Only the objects with high circularity (above 0.9) were selected. The objects components with an area less

than 3 pixels were also discarded to further eliminate misclassifications. The results are presented in the Results and Discussion chapter.

3.2.3 Synapse Toolbox

The data collected for the Synapse Toolbox was stored in .csv format files, each containing measurements from all inhibitory or spine synapses from a single dendritic branch. Each branch was associated with a specific dendritic region (basal or oblique) from a specific cell, as multiple neurons and regions were used for the measurements. For instance, the data from the inhibitory synapses of branch 1 of the oblique dendrites of cell 1 would be in a file designated by:

Cell_001_Oblique_branch_001_shaft.csv

The type of synapses is quoted in each file as ‘shaft’ or ‘spine’, referring to their location. These terms will be used interchangeably with ‘inhibitory’ and ‘excitatory’ in the remainder of this section, since inhibitory and excitatory synapses are usually located in those respective regions. However, exceptions exist, as 10% of excitatory synapses are estimated to reside in the dendritic shaft, and 30% of inhibitory synapses have been recently attributed to dually innervated spines [51], which carry both an excitatory and an inhibitory synapse. As the measurement method technically identified inhibitory synapses and spine synapses, and no two types were found to coincide in the same location, the two nomenclatures will be used without distinction.

Each file is consistently formatted, with a header spanning two rows followed by a set of rows representing synapses, each with three measurements: distance from beginning of branch, synaptic strength and branch length. The shaft data from branch 1 of the basal region of cell 20 is shown in Figure 3.52. All files, corresponding to 9 cells and 52 branches in total, were stored in a single folder.

	A	B	C	D	E
1	Sum_Shafts[][0],Sum_Shafts[][1],Sum_Shafts[][2]				
2	0,1,2				
3	3.576,0.54306,41.537				
4	13.067,1.6464,41.537				
5	21.283,0.512665,41.537				
6	24.159,0.956434,41.537				
7	32.173,1.42756,41.537				
8					

Figure 3.52: Data from Cell_020_Basal_branch_001_shaft.csv

The MATLAB program designed to handle these files is divided into functions, described below.

3.2.3.1 Load Data

Function:

```
Load_Data()
```

The first function is meant to read the desired files and store their data into a struct object, to be used by all later functions. Its call in the main script is done with the following line:

```
dataStruct = Load_Data();
```

This function starts by opening a search window to allow the user to select the folder containing all the .csv files. Then, it reads the name of every file and decomposes them into cell number, region and branch number. Based on them, it asks the user to type the desired values, so it can selectively load the data from the .csv files into the dataStruct, as illustrated in Figure 3.53.

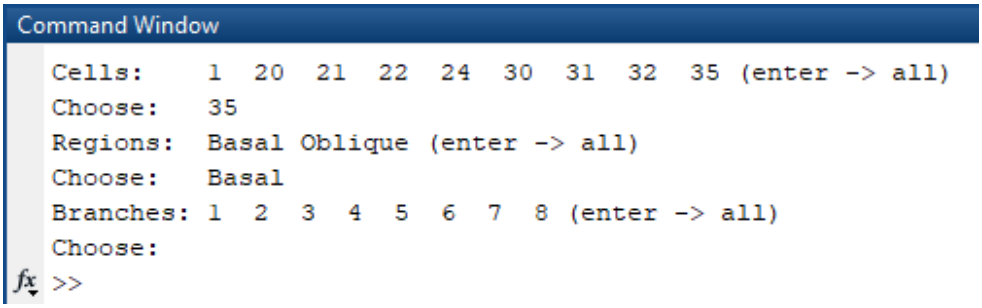


Figure 3.53: Interactive selection of the synaptic data based on cell, region and branch.

3.2.3.2 Graphical Display

Function:

```
Display_Branches(dataStruct)
```

The second function represents the data graphically so it can be easily read. Each synapse is represented as a circle, with a radius proportional to their intensity, positioned along a branch according to their distance value. The followed model is shown in Figure 3.54.

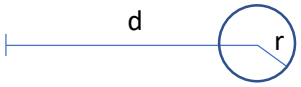


Figure 3.54: Chosen graphical model of a synapse.

Each synapse is plotted iteratively as a circle with a radius mapped linearly from its intensity value. To add information to the plot, the synaptic density is also calculated with a moving gaussian window across each branch. The standard deviation for the windows for both synapses was

chosen to be 1.2 μm. This value was chosen empirically with the precaution of being higher than 0.2 μm, the approximated light diffraction limit for the experimental setting in which synapses were imaged. To convert to window units, this value was multiplied by the resolution, chosen to be of 10 units/μm. For each type of synapse, each branch was converted to a binary vector, with ones being the location of the synapses. The vectors were convolved with the windows to create the synaptic density vectors, which could then be plotted on top of the branches. Optionally, the density can also be calculated considering the intensity of each synapse. In this case, instead of a binary vector, each branch is represented by a vector with intensity values in the position of the synapses, then to be convolved with the gaussian window.

The rest of the functions were designed to analyze the loaded data. Based on the model of Figure 3.54, a single synapse can be viewed as a distance or intensity. Furthermore, a set of synapses can be viewed as a set of distances, intensities, or simple by their number. The following functions explore these quantities, either with metrics targeting individual synapses, such as an intensity, or groups, such as a density. All functions distinguish inhibitory from excitatory synapses, so their results can be observed individually and compared. The algorithms are organized as follows.

3.2.3.3 Distance and intensity counts per cell

Functions:

```
Inspect_spineHistogram_Distances(dataStruct_cell)
Inspect_inhibHistogram_Distances(dataStruct_cell)
Inspect_spineHistogram_normDistances(dataStruct_cell)
Inspect_inhibHistogram_normDistances(dataStruct_cell)
Inspect_spineHistogram_Sizes(dataStruct_cell)
Inspect_inhibHistogram_Sizes(dataStruct_cell)
Inspect_spineHistogram_normSizes(dataStruct_cell)
Inspect_inhibHistogram_normSizes(dataStruct_cell)
```

The first set of functions characterize the data by counting how frequent fixed ranges of distances and intensities are in each selected cell. They select all distance and intensity values from inhibitory and excitatory synapses in a single cell and plot them using MATLAB's command `histogram` with automatic bin width. These functions have also normalization variations, in which distance and intensity values are divided by branch length, to make the analysis independent of their absolute value. These functions are inside for-loops which iterate over the cells selected in `Load_Data()`, allowing them to plot the results for every cell in a single run.

3.2.3.4 Distance *vs* intensity per cell

Functions:

```
Inspect_normDistance_Intensity(dataStruct_cell, color_now)
```

```
Inspect_Distance_normIntensity(dataStruct_cell, color_now)
```

After characterizing the distance and intensity values of synapses, these quantities were then correlated in a single plot. Each synapse was plotted with its distance and intensity (or their normalized variants) as its coordinates, to test if there was any sort of proportionality between the two.

3.2.3.5 Total excitatory distances, intensities and number *vs* total inhibitory distances, intensities and number

Functions: `Inspect_totalExciInhi_rawData(dataStruct_cell, color_now)`

Synapses were then characterized collectively by branch, by calculating the sum of distances, intensities and their number for each branch, plotted against the same quantities but for the opposite type of synapse.

3.2.3.6 Total intensity *vs* branch length

Functions:

```
Correlate_TotalIntensity_BranchSize(dataStruct)
```

Expanding on the last method, total intensity for each type of synapse was also graphed against branch length, to verify the hypothesis of longer branches having more and stronger synapses.

3.2.3.7 Inter-distances and inter-intensity counts

Functions:

```
Analyse_inhib_interDistances(dataStruct)
Analyse_spine_interDistances(dataStruct)
Analyse_spineDistances_around_Spines(dataStruct)
```

Using the distance and intensity data of each synapse, the distance differences between consecutive synapses and the intensity differences between consecutive spines were calculated. The calculated values were pooled with the values from all other chosen cells and plotted using the `histogram` command. For the inter-distance plot, the bin width was empirically chosen to be 0.3 μm for excitatory synapses and 4 μm for the inhibitory counterparts (since the data was scarcer); the same was done for every branch, with a bin width of 1 μm for excitatory and 6 μm for inhibitory synapses. For the inter-intensities, the bars were normalized to approximate a probability density function, and the bin widths were set to automatic. As a consecutive spine can have an intensity higher or lower than the previous, both negative and positive differences were used in the histogram plot. The inter-distance data was also represented as a gamma probability density function, using the commands “gamfit” and “gampdf”. To serve as a comparison reference, an

exponential probability density function with equal mean was generated, using "exppdf". Both functions were plotted on top of the inter-distance histogram to facilitate their visual interpretation. Synaptic density was the next quantity to be analyzed, as described below.

3.2.3.8 Density autocorrelation

Functions:

```
Analyse_inhib_interDistances_autoCorr Analyse_spine_interDistances_autoCorr...
```

To evaluate the periodicity of the synapses, a density autocorrelation was performed. For each type of synapse, each branch was represented as a binary vector with ones as its synapses. Instead of feeding these vectors directly into an autocorrelation command, they were transformed into density vectors, to smooth the data and to obtain broader but more robust results. Similarly to what was done for the density displays, each vector was convolved with a gaussian window, with standard deviation of 0.5 μm for excitatory synapses and 1.5 μm for inhibitory synapses, both multiplied by a resolution of 10 units/μm. Optionally, the density vectors could also consider the intensity of each synapse, using these values instead of the ones in the otherwise binary vectors. The density vectors were then fed into the "xcorr" command and their results plot separately for excitatory and inhibitory synapses.

3.2.3.9 Excitatory density around inhibitory synapses

Functions:

```
Analyse_spineDensity_around_inhibSynapse(dataStruct)
```

To evaluate the influence of inhibitory synapses on excitatory stimuli, a neighborhood of approximately 40 μm around each inhibitory synapse was analyzed. The set of densities spanning these neighborhoods were extracted from the excitatory density vectors (built as previously mentioned) and stored for each inhibitory synapse. Then, the mean and standard deviation were taken for each distance in the neighborhood, and plotted in the same graph.

3.2.3.10 Mean inhibitory intensity *vs* excitatory density

Functions:

```
Inspect_SizeA_branchDensityB(dataStruct_cell)
```

The mean intensity and standard deviation of the inhibitory synapses were calculated for every branch in the selected cells. The densities of the excitatory synapses were calculated for the same branches, and both values were plotted as coordinates in a scatter plot, with different colors corresponding to different cells.

3.2.3.11 Inhibitory intensity *vs* distance to nearest neighbor

Functions:

```
Correlate_inhIntensity_distanceNN(dataStruct)
```

Finally, the intensity of inhibitory synapses was correlated to the distance between them and the nearest inhibitory synapse. For each one, the minimum distance between it and the others was calculated and plotted against their intensity.

All figures generated by the toolbox functions are presented in the correspondent Results and Discussion section of this report.

Chapter 4

Results and Discussion

4.1 Geometrical 3D Algorithm

The proposed geometrical method was validated using the ground truth from a manually segmented image with dendritic spines. The manual segmentation was performed using ImageJ's IntSeg 3D plugin, an opensource tool for 3D image annotation created by Prof. Martin Heisenberg and his research team from the University of Würzburg, in Germany.

The performance of the method was evaluated objectively in terms of spine counting and segmentation. Although dendrite segmentation is not the aim of this work, the calculated masks are also discussed in this section, to contextualize subsequent results. Finally, the measurement values are presented, and interpreted considering the morphology of the correspondent segmented spines.

Following preprocessing, the dendrites were segmented from the 3D image data to create a region where spines could be detected and classified based on their morphology and context within the dendrite. From the second derivative and intensity content of the image, two complementary masks were created, one exhibiting spine necks although susceptible to noise (Hessian mask), and the other with a higher signal to noise ratio but incomplete necks (tubular mask). The comparison of each is shown by their contours in Figure 4.1, overlaid on top of the median filtered preprocessed image.

Although the Hessian mask classifies background voxels incorrectly in some regions of the image, it is not evident if it overestimates the foreground in true dendritic domains. In the bottom spine of Figure 4.1-b), for instance, the blue contour seems to more accurately enclose the intensities of the spine and neck, while the tubular mask suggests a non-uniform shape which does not seem to correlate with the intensity profile of the region. It was due to the ambiguity of the dendrite borders that the proposed shrinking step was made optional, allowing the spine regions of the Hessian mask to be considered as the final segmentation.

After dendrite segmentation, the automatic detection was performed. For the same image as of Figure 4.1, the detected spine coordinates are shown in Figure 4.2, overlaid with the tubular mask to facilitate the visualization.

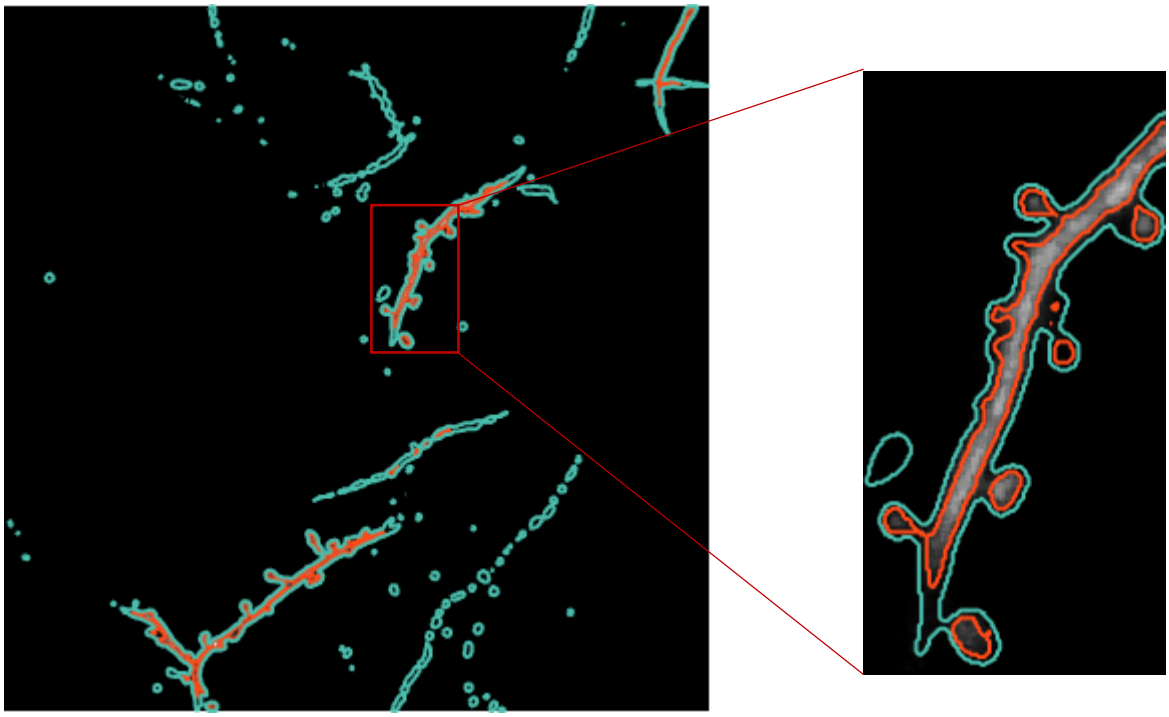


Figure 4.1: Contour comparison between the Hessian (blue) and tubular mask (red), overlaid on a slice of the median filtered stack.

The number of spines counted was 87. During the detection module, the three skeletons from the Hessian mask with most endpoints were selected to enhance the probability of detecting spines. However, neither of the last two had any spine points in the foreground of the tubular image, so they were discarded by the correction phase. From the 87 points, 46 were found belonging to the ground truth manually calculated, giving a precision of approximately 53%. As the ground truth includes 103 spines, this leads to a recall of 45% and an F1 score of 49%. This value is considered low and can be justified by two main reasons. First, slight variations between the Hessian mask and the ground truth could cause some points to be placed in the neighborhood of the correspondent spine, falling outside the ground truth image. In fact, if the ground truth was to be eroded by a spherical mask with a diameter of 3 voxels, the precision would increase to 69%, and to 72% if the diameter was set to 5 voxels. In these cases, the detection could still be useful as a semi-automatic tool to locate spines, since it suggests small regions where the probability of encountering a spine is moderately high. The second reason is due to the influence of noise on the Hessian mask. Since this mask considers the maximum second derivative response of the image convolved with Gaussian filters of different standard deviation, the presence of unfocused structures or noise can be captured by the calculation of neighborhood voxels, increasing the thickness of the foreground. Figure 4.3 shows that an unfocused dendrite edge can cause an artificially large Hessian mask. In this case, the thinning method creates a skeleton which does not represent the real structure, leading to inaccurate detection of spine coordinates.

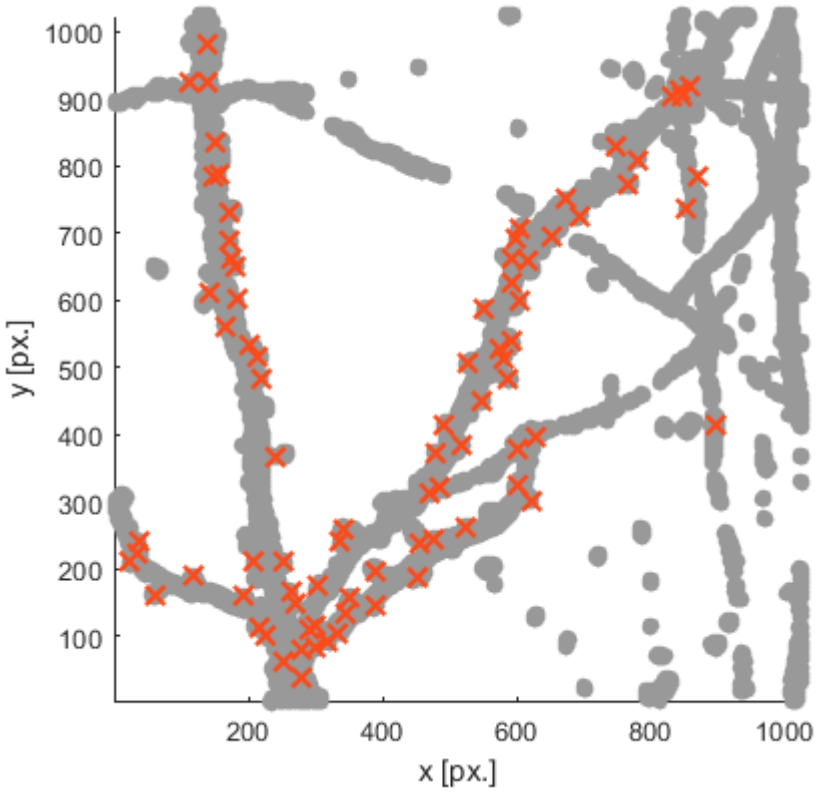
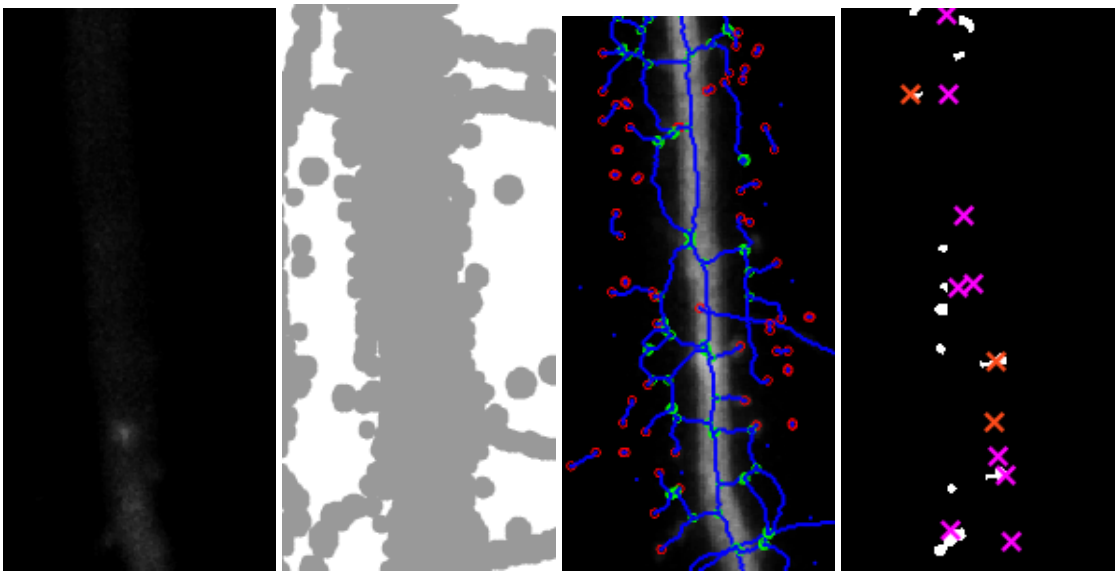


Figure 4.2: Spine coordinates (shown as red crosses) from the detection module.



(a) (b) (c) (d)

Figure 4.3: The effect of unfocused structures on the results of the detection module: a) Unfocused dendritic edge; b) Hessian mask in that region; c) Skeleton of the Hessian mask; d) Correctly (orange) and incorrectly (magenta) detected coordinates on a slice of the ground truth.

The reduction of noise by a stronger preprocessing step or a different thresholding technique reducing its effect (both without compromising the spine necks) would increase the quality of the results. If these challenges are surpassed, the method could use their results without further configuration.

It's worth mentioning that the noise reduction in the image collection phase may also be a possibility of improving the results. However, this task could be tightly connected to the equipment used for the acquisition, and expecting higher quality images than the ones provided (which show both noise and unfocused structures) might decrease the applicability of the program.

The final segmentation results are presented in Figure 4.4, numbered according to the order of segmentation.

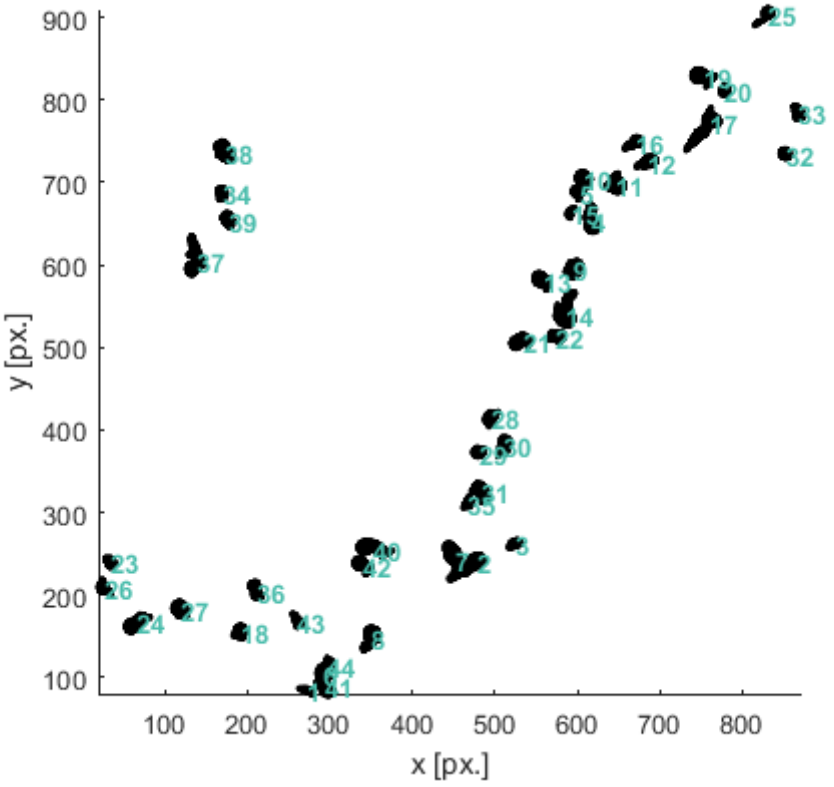


Figure 4.4: Final segmentation of the dendritic spines.

The spines are displayed as 3D volumes and thus can be interactively analyzed within MATLAB's environment. Their final number is 44, approximately half of the detection coordinates, roughly the number of detected coordinates found to belong to spines (46). The segmentation was evaluated based on the confusion matrix with respect to the ground truth, which is presented in Figure 4.5.

The highest number correspond to the true negatives, i.e., voxels that were determined correctly as background. The image is dominated by background voxels however, which would be

	Predicted non-spine	Predicted spine
Real non-spine	48117126	28141
Real spine	65247	23982

Figure 4.5: Confusion matrix of the final segmentation.

expected to be discarded as spine candidates, at least in their majority. The true positives have the lowest value of the remaining three, calculates relatively less spines correctly than incorrect spines or correct background voxels. The precision and recall reflect this performance, with respective values of 46% and 29%. This leads to the low value of the F1 accuracy measure of 34%. One of the reasons explaining such a performance level is the missed detection of spines in the detection module, and the detection of incorrect ones. Considering the correct coordinates, some remain in the shaft close to the spine, which can sabotage the tracing phase and reduce the number of candidate sites even further. The thickness of the Hessian mask not only affects the detection phase, but also the tracing, making an unclear path for the algorithm to travel through the spines and less drastic difference of the foreground between the spine and the shaft, reducing the efficacy of the stopping conditions.

One of the main faults is therefore attributed to the preprocessing and neurite segmentation stages. The attempt of performing a reliable segmentation in masks of incomplete qualities led to the constant influence of artefact's in every stage of the procedure, which surpassed the benefits both offered and undermined the results. As the spine segmentation module bases its algorithms in geometric properties still considered valid, it is projected that the improvement of the preliminary phases will improve the results significantly.

The quantification of the segmented spines is shown in Figure 4.6 and 4.7, divided in two due to their combined extent.

The volume and surface areas can be taken to describe the shape of spines. For instance, spine 18 seems to be spherical by visual inspection, and if the radius of an hypothetic sphere was calculated with their volume and and area, the results would be approximately the same (around 6 voxels), which indicates its close resemblence to this shape. The solidities close to one also indicate a high convexity, although are values above one outputed by MATLAB's routine are suspicious. In theory, the solidity is calculated by the ratio between the volume of a given object and a convex shape enclosing it. The values were not expected to be higher than one, which will require some investigation. The centroids were calculated to facilitate the distribution of volume amongst spines, and can therefore be useful in the classification between types of spines.

In spite of being undermined by the preliminary phases, this component still proposes an enhanced Rayburst tracing method, a fully original delimination phase and a set of objective measures, leading to the increasingly automated study of these structures.

ID	Volume	SurfaceArea	Solidity	CentroidProximity
1	321	318.41	2.5	4
2	1567	741.82	0.9	6.4
3	591	402.32	1	2.1
4	1901	1076.7	1.9	11.4
5	858	537.94	0.8	4.1
6	1762	826.2	1.1	7.5
7	2848	1685.2	0.6	20.8
8	1514	941.59	0.7	637.8
9	1958	856.67	0.9	6.6
10	1150	623.11	0.9	6.8
11	1830	942.09	0.7	7.6
12	1185	715.59	1.4	4.8
13	1200	748.29	0.7	7
14	3186	1411.9	1.6	9.7
15	347	284.26	0.9	1.4
16	827	558.53	1.2	3.1
17	1830	1238.1	0.4	10.5
18	787	504.54	0.8	5
19	1627	938.01	0.7	11
20	429	314.57	1	3.7

Figure 4.6: Measurement results for the first 20 segmented spines

21	1064	695.85	1.1	8.9
22	837	524.82	1.2	6
23	527	365.18	0.8	2.5
24	1159	797.66	0.8	11
25	620	475.29	0.6	2.3
26	908	565.82	0.8	2
27	1165	705.31	0.7	7.3
28	942	571.34	0.8	3.5
29	627	436.9	0.8	5.6
30	1552	766.46	1.1	4.3
31	1449	841.2	0.7	8.4
32	198	198.99	0.7	4.5
33	565	446.83	0.8	3
34	838	473.01	1.1	4.1
35	447	343.32	1.1	1.9
36	1452	763	1	3.6
37	2161	1300	2.4	6.1
38	1759	984.38	0.8	6.4
39	1076	586.09	1	4.2
40	1921	1194.7	1	17.4
41	1182	769.16	0.7	9.5
42	683	616.41	0.5	8.4
43	528	389.35	1.3	2.9
44	745	482.58	1.1	3

Figure 4.7: Measurement results for the last 24 segmented spines

4.2 Machine Learning 2D Algorithm

The performance of the Deep Learning based detection method was evaluated on small regions in the training, validation and testing phase, and finally in on a complete image in the prediction phase. The training/validation loss and accuracy are presented in Figure 4.8 for each epoch.

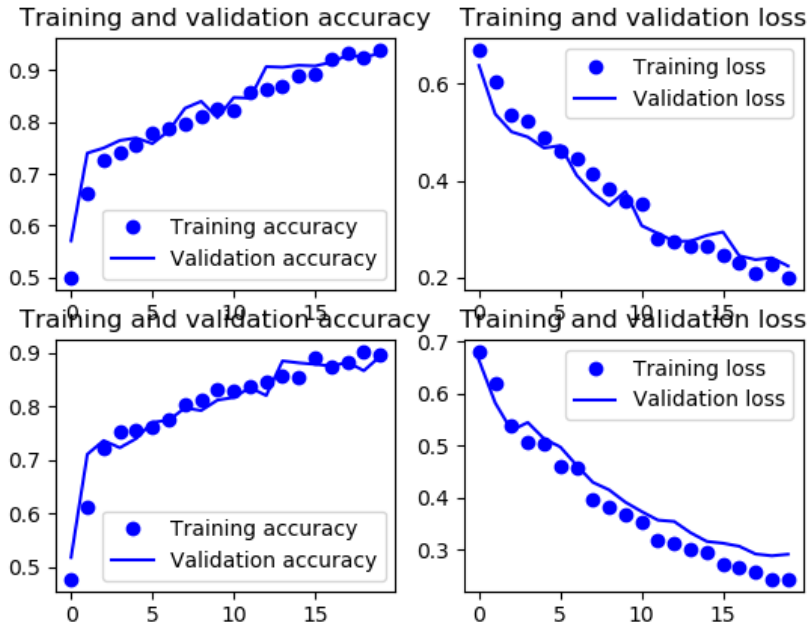


Figure 4.8: Performance of the neural network on both training and testing phases

As expected, during training the classification results improve over training iterations (epochs), expressed by either an increase of accuracy and a decrease in loss. The classification was compared to both the training and the validation set, since a 2-fold cross validation scheme was designed. Although it does not exclude the possibility of overfitting, these results show there is a consistent improvement of classification ability during this phase.

The algorithm was then tested against a small new test of windows, and outputed a classification accuracy of 87%. This was regarded as a positive result, although some networks with different architectures yielded higher accuracies. The reason why they were not selected was due to their poor performance on the final image. This network was considered to have a desirable balance between classification success and error, so that it didn't express a too high dependency on the training set.

Supported by the previous evaluations, the model was used in the prediction phase to originate a probability image, which by the post-processing phase, was transformed into the image of Figure 4.9.

The image was compared with the ground truth and the follow confusion matrix was calculated:

Figure 4.9: Final classification results by the 2D detection method

	Predicted non-spine	Predicted spine
Real non-spine	1039407	1008
Real spine	6805	1356

Figure 4.10: Confusion matrix of the final segmentation of the supervised method

The high number of true negatives is once again to the high ratio of background in relation to foreground. The method exhibits a precision value of 57%, a recall of 17% and a F1 measure of 26%. This algorithm, contrarily to the previous, does not aim the segmentation of spines, but solely its detection. Its underestimation of foreground pixels (lowering the values of recall and F1) is thus secondary, being the accurate marking of spines (independently of the size of the markers) preferable.

The number of detected spines by the algorithm takes the value of 90, a level closer to the manual selected 103 spines. Similarly to the geometrical algorithm, however, only 45 of those coordinates belong to manual selected spines. It is possible that, in some cases, the method was

able to detect the presence of a spine, but because it is based on a local window estimate, the center of the window turned to be slightly outside the border, generating a high spine probability value in a background pixel. In these cases, the suggestion of regions containing spines might be useful for semi-automatic detection methods, although their exact locations are not reliably inferred.

The size of the dataset might also had a central role in the accuracy of the method. Only 70 original images were collected, and replicated into 2100. The original collection hardly matches the amount of data extracted in similar classification problems, which can reach hundreds of thousands, and so a larger image dataset is recommended.

This component stands as a fast classification method, trained with a reduced number of images and a simple architecture.

4.3 Synapse Toolbox

The results of all functions introduced in the Methods section are presented below in the same order of introduction.

4.3.1 Graphical Display

Spines and inhibitory synapses were displayed graphically to enhance the visual interpretation of the unprocessed data. Figure 4.11 and 4.12 shows the distribution of each type in every branch of one arbitrarily chosen cell. The data is presented in two forms: one dividing each branch in two to separately plot each type of synapse (4.11), and another plotting them in the same line together with the density estimation for each type (4.12). The type of synapse is differentiated by color, their intensity by their circle diameter, and their position along the dendritic branches by their distance values, which are in respect to the beginning of the branch.

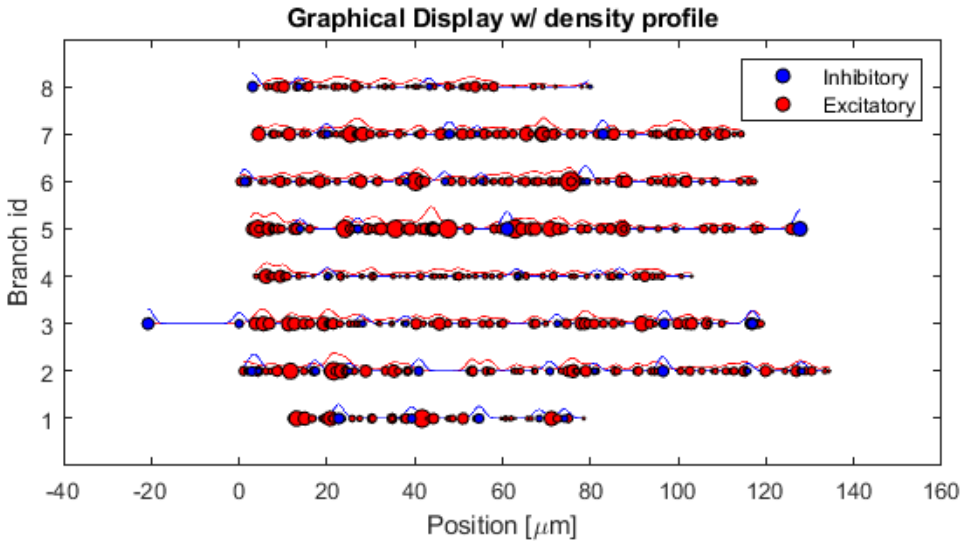


Figure 4.11: Graphical display of the synapses of Cell 21.

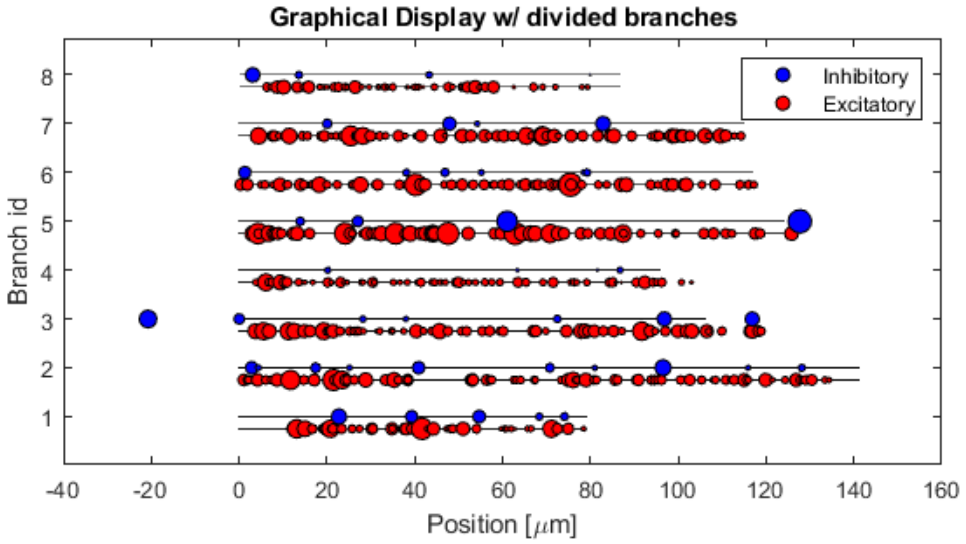


Figure 4.12: Graphical display of the synapses of Cell 21 with density profiles

As shown, spines are predominant in the measured dendrites of cell 21. This was to be expected, since studies suggest nearly 80% of synapses in the mammalian cortex are excitatory [52]. The ratio was found to be, however, more disparate, as this cell, for instance, had a percentage of 92% of spines in relation to the totality of synapses.

4.3.2 Distance and intensity counts per cell

To characterize the distance of synapses from the beginning of their branch and their intensity, these values were grouped in small intervals and plotted as histograms. Figure 4.13 shows the distance and intensity plots (normalized or not by branch length), for both types of synapse of three different cells.

As the bin width was set to automatic and there are more spines than inhibitory synapses, the number of their bins is also higher. The lower number of inhibitory data also hinders its interpretation, yet it was chosen not to merge the inhibitory data of different cells to increase its number out of precaution, since the data variability between cells was typically higher than the variability between branches of the same cell in several tests.

The histograms characterizing the position of synapses do not appear to show a unique behavior, suggesting there is no preferential distance for either spines and inhibitory synapses to form along the dendrite. This corroborates the conclusions drawn by [53] who, by using quantile analysis, found the locations of spines to be completely spatially random in their dataset, regardless of the maturity of the neurons. Contrarily, the peak value of the intensity histograms suggests a preferential synaptic intensity in every cell (although inhibitory synapses suffer from few synapses). The peak becomes higher if the intensities are normalized, further suggesting that the preferential intensity may depend on branch length.

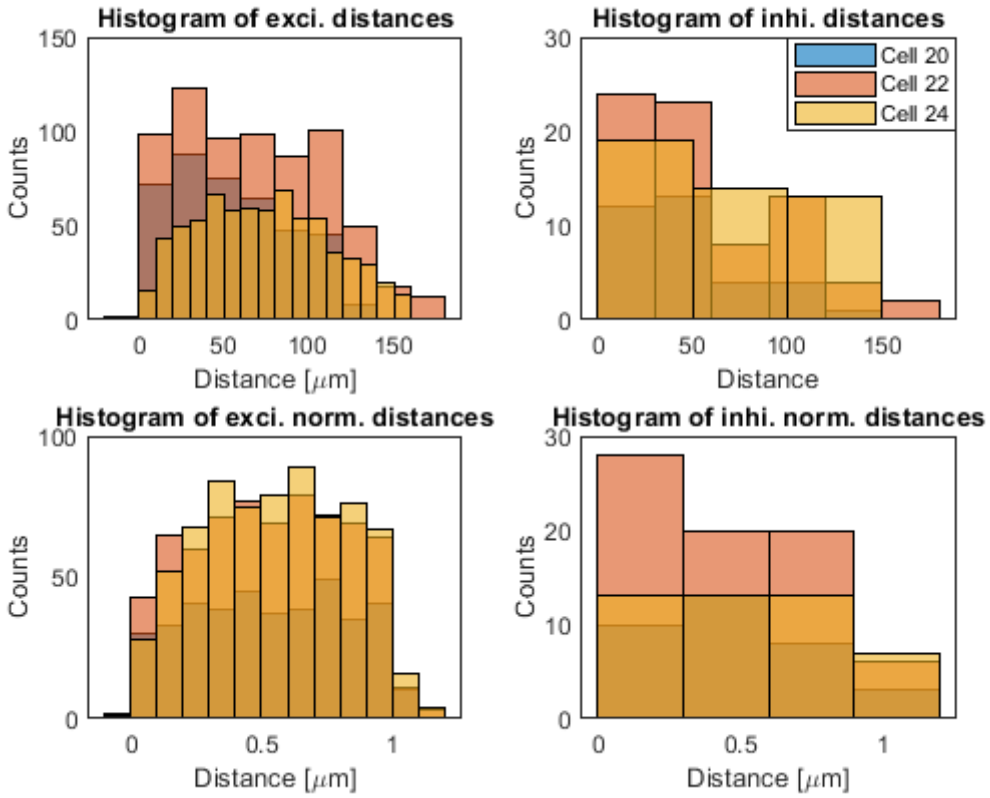


Figure 4.13: Distance histograms for the synapses of three distinct cells

A preferential value was also found by [54], but in spine actual size instead of fluorescence intensity. The volumes were measured by 3D reconstructions, and the distribution of spine volumes was also broad and asymmetrical, with a peak of 0.06 μm^3. Such similarity supports the validity of using fluorescence intensity to qualify spine volume.

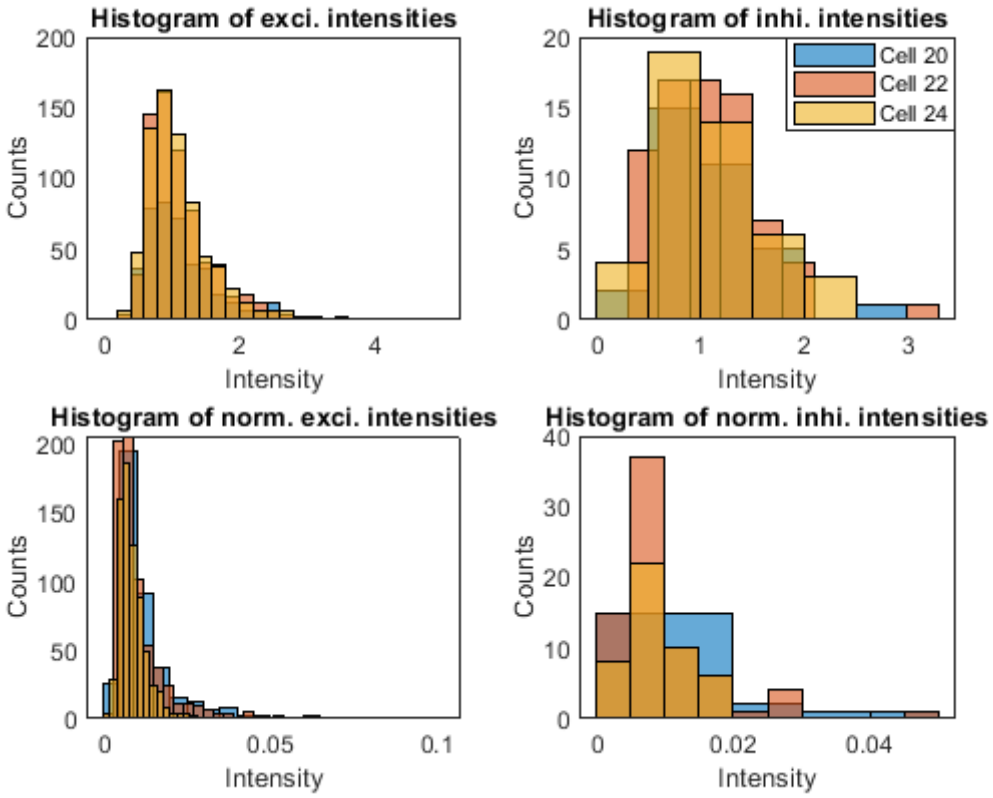


Figure 4.14: Intensity histograms for the synapses of three distinct cells

4.3.3 Distance *vs* intensity per cell

Distance and intensity were correlated to inspect the existence of regions with preferential intensities along branches, and a possible spatial proportionality across them. Figure 4.15 shows the correlations for both types of synapses, on three different cells.

From the plots, it can be apparent that the data aligns horizontally near the intensity level 1. However, the coefficients of determination (R^2) for each linear regression were considered low, with 0.044 being their highest value, and corresponding to the excitatory synapses of cell 24. To complement this information, no significant relation between synaptic distance to the soma and intensity was also found by [54], who calculated a R^2 of 0.014 on data from layer 2/3 of pyramidal neurons. Thus it appears that distance along branch and dendrite is unrelated to the mechanisms which control the morphology of spines in pyramidal cells.

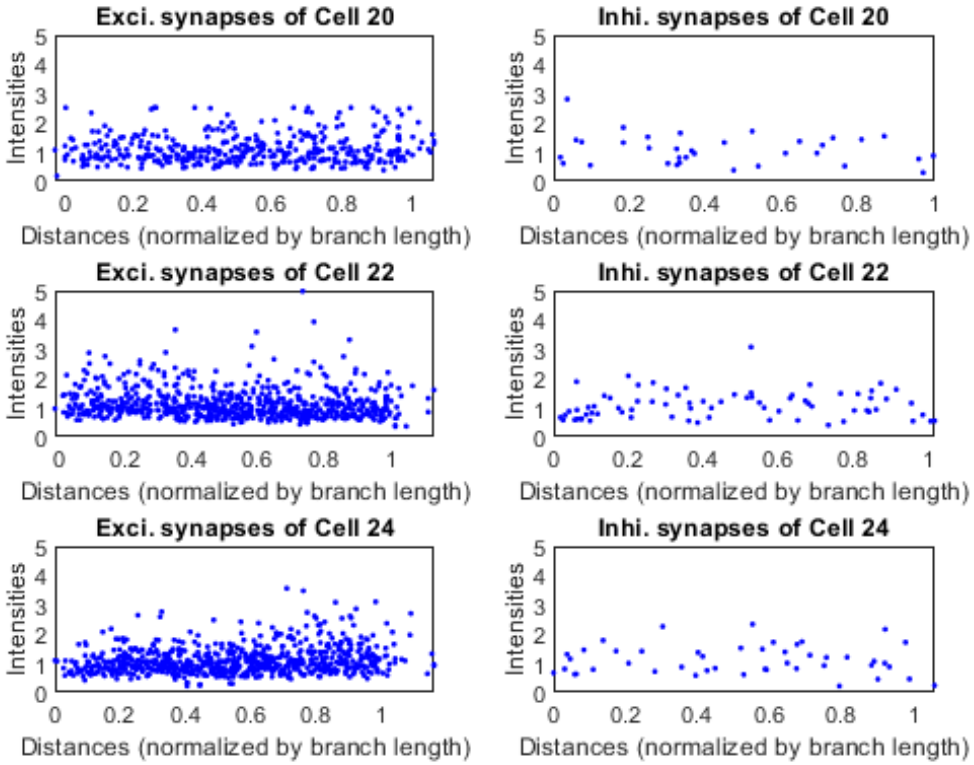


Figure 4.15: Correlation between distances and intensities, considered individually per cell

4.3.4 Total excitatory distances, intensities and number *vs* total inhibitory distances, intensities and number

The correlation between distances, number and intensities between synapses of different types in each branch, is displayed in Figure 4.16.

As it can be observed, for every branch (tagged with the letter B plus its number in the cell) the sum of distances from each type of synapses express a stronger linearity than their number. This suggests the position of synapses depend on their number, to compensate its variations and preserve a distance linearity. R^2 was found to be 0.75 for total distances. In regards to total intensity, the points exhibit a poor correlation with R^2 equal to 0.47.

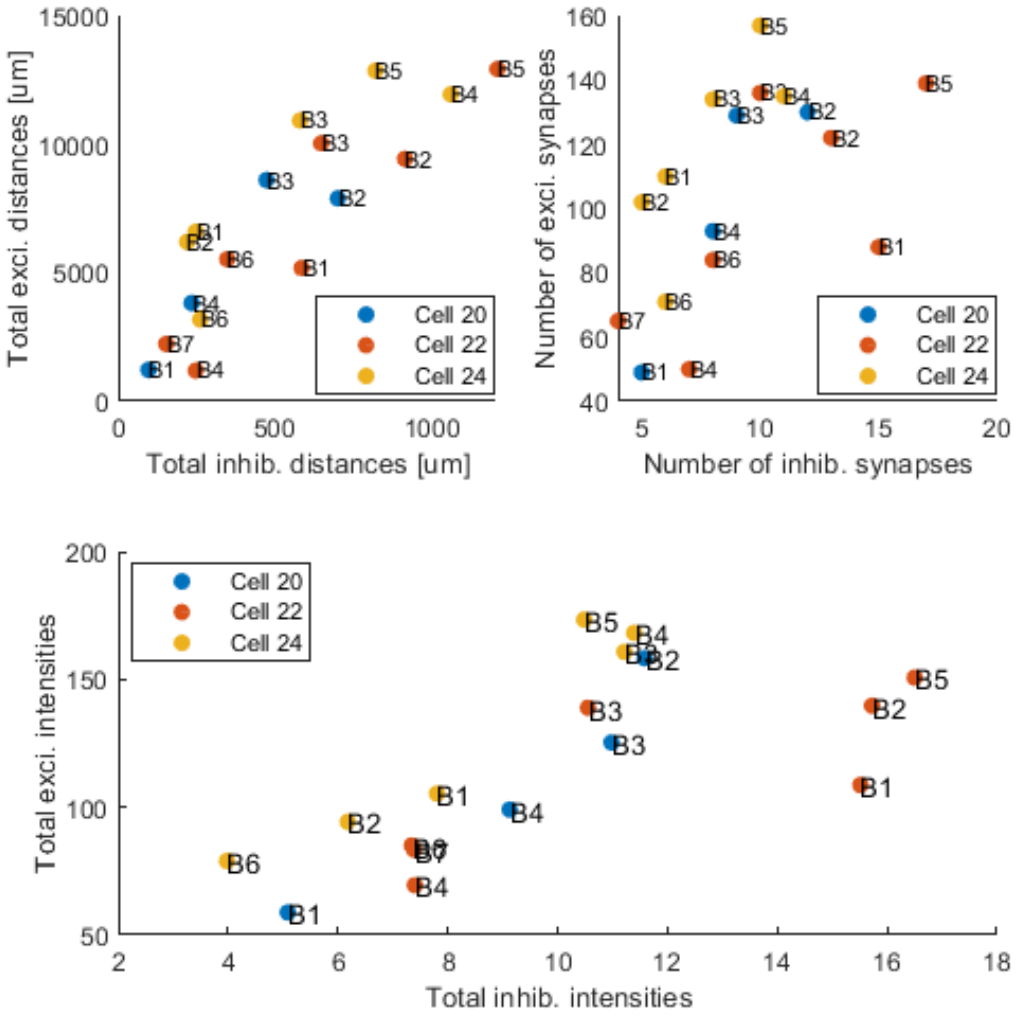


Figure 4.16: Correlation between the number and intensity of inhibitory and excitatory stimuli

4.3.5 Total intensity *vs* branch length

Figure 4.17 correlates the sum of intensities (for each type of synapse, in each branch of each cell) with branch length.

The sum of intensities exhibits a small dependency (R^2 equal to 0.76) with branch length, as shown in the left graph of Figure 4.17. The same is not verified, however, for the inhibitory stimuli, whose strength seems independent from the length of their branches, as shown in the right graph (R^2=0.50).

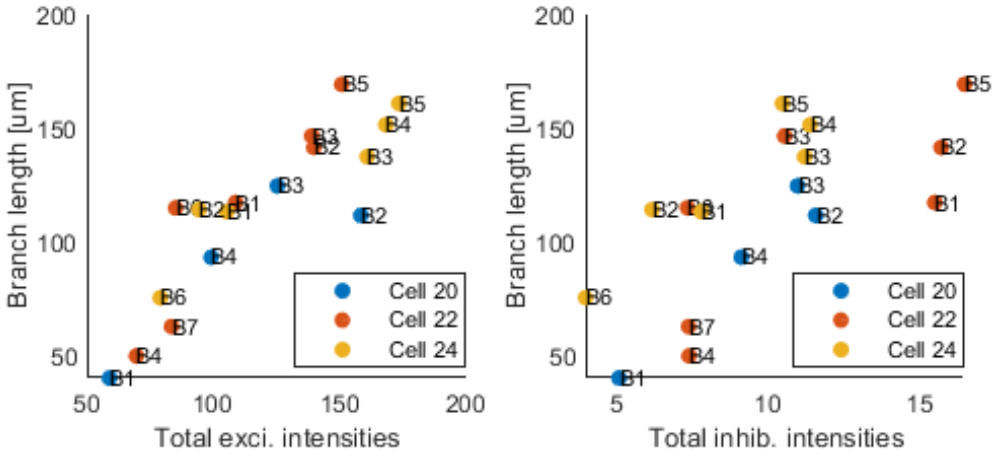


Figure 4.17: Excitatory and inhibitory intensities correlated with branch length

4.3.6 Inter-distances and inter-intensity counts

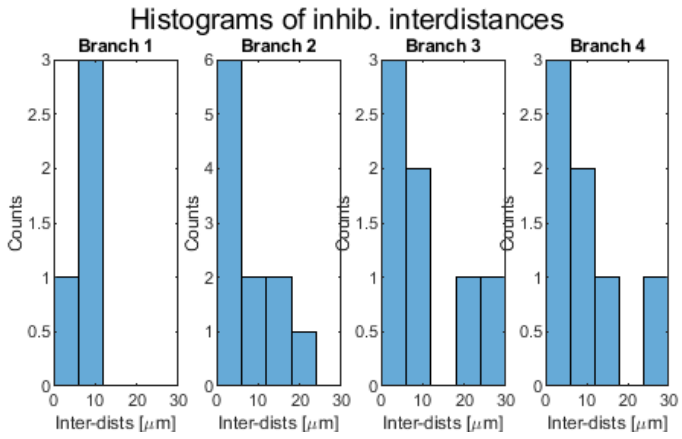
To analyze the spatial relationship between synapses and their neighbors of the same type, the distance between every consecutive pair was calculated. The histograms of these inhibitory and excitatory inter-distances are shown in Figures 4.18, considering the branches of one cell, and 4.19, considering three different cells.

The data from the previous three cells was merged to make these plots, in favor of the quantity of the inputs, since otherwise they would be too scarce for a histogram representation spanning their range of distances, as it is evident from the lack of data of Figure 4.18, specially for shaft synapses.

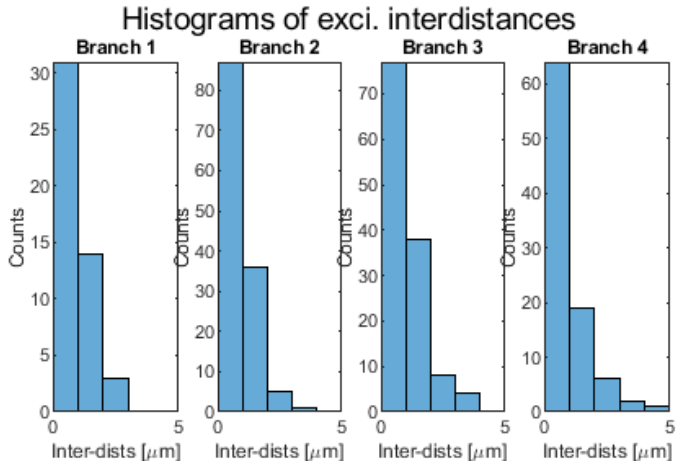
From visual inspection of Figure 4.19, it may be noticed that some spine inter-distances seem to cluster, although in the inhibitory type they are even more ambiguous. In the excitatory plot of the three cells, a decrease in the initial part of the of the histogram is noticeable, which indicates spines may repel each other under a certain distance. This is emphasized by the experimental exponential distribution function (blue), which behaves similarly to the exponential function based on random data (red), except for the beginning, where the experimental function forms a valley, indicating a lack of small distances between synapses.

Figure 4.20 shows the inter-intensity bar plot for the sorted synapses of the same three cells, as well as for a random permutation of their order.

There seems to be a bias towards low differences in intensity. However, the same occurs to the randomly permutated data, which by this method of analysis, makes the tendency insignificant.



(a)



(b)

Figure 4.18: Histograms of the inter-distances of the a) inhibitory and b) excitatory synapses of Cell 21, per branch.

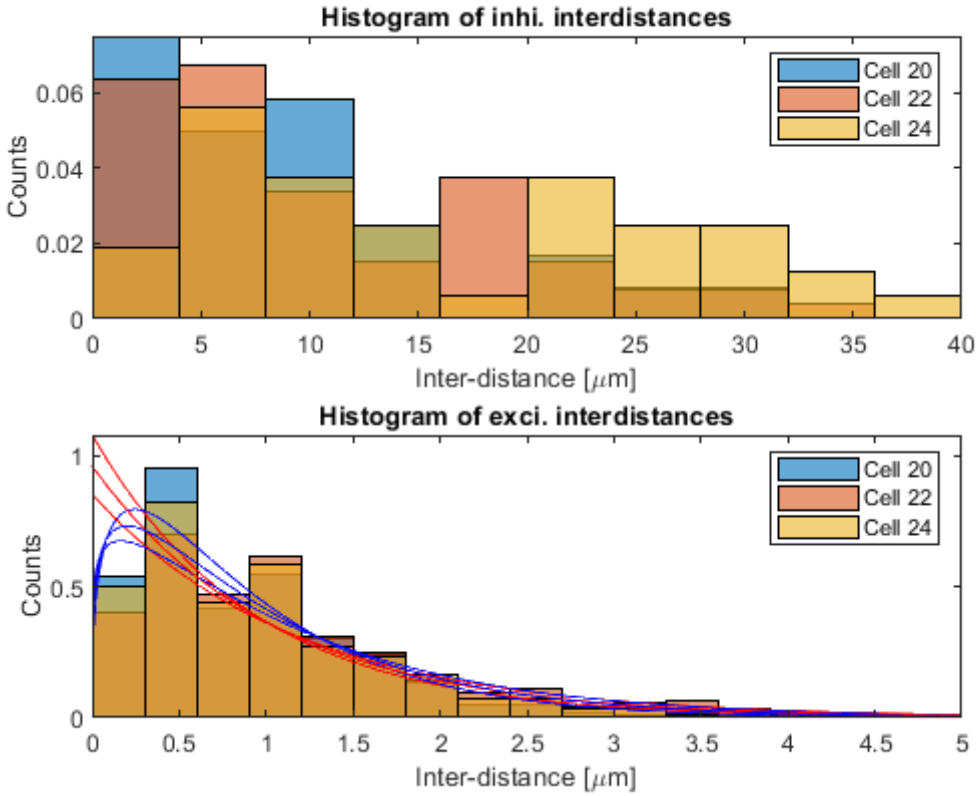


Figure 4.19: Histograms of the inter-distances of the inhibitory and excitatory synapses of Cells 20, 22 and 24.

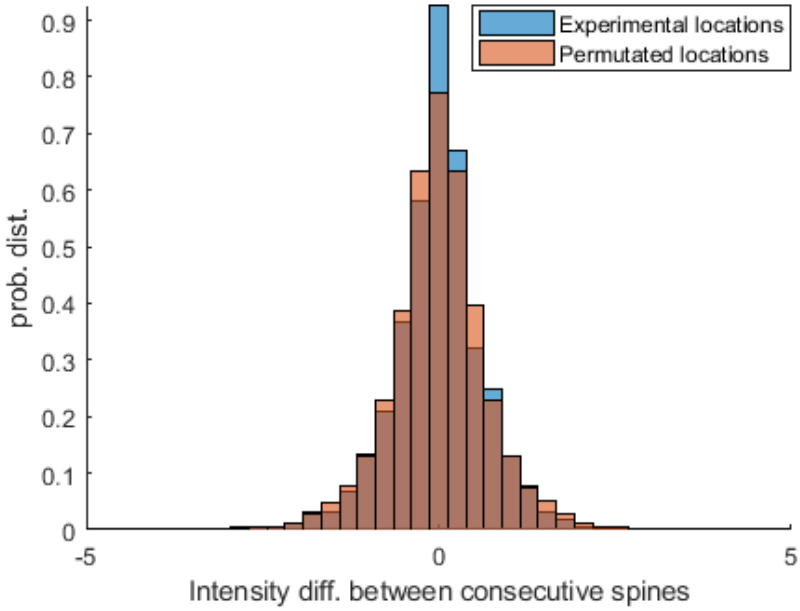


Figure 4.20: Distribution of the intensity differences between consecutive spines (inter-intensities)

4.3.7 Density autocorrelation

The autocorrelation analysis was done to inspect if there is any periodicity amongst the positions of synapses. Figures 4.21 and 4.22 show the results for inhibitory and excitatory synapses, respectively.

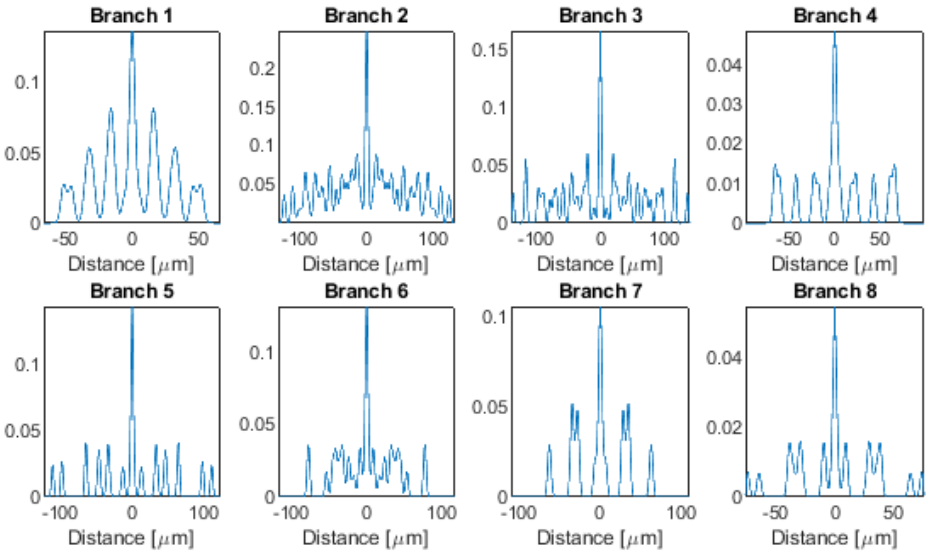


Figure 4.21: Periodicity analysis of inhibitory synapses by means of autocorrelation methods

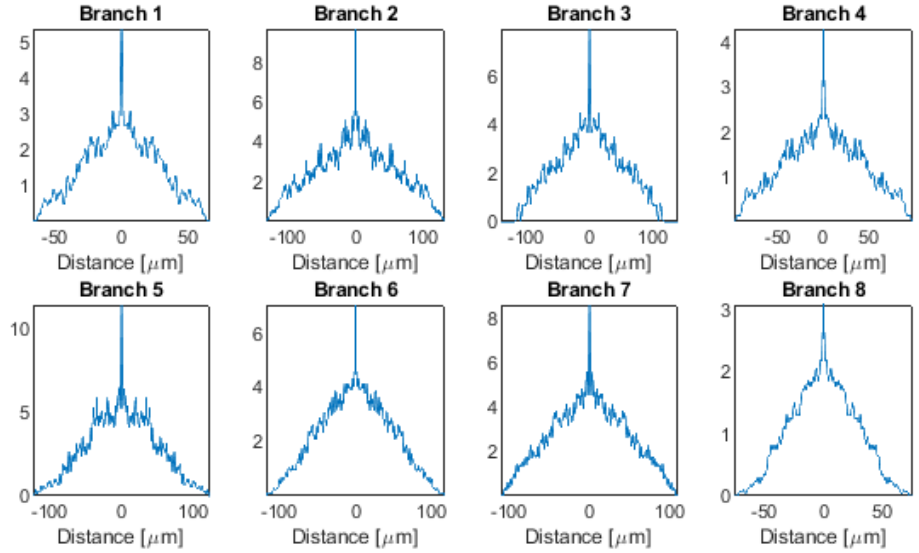


Figure 4.22: Periodicity analysis of excitatory synapses by means of autocorrelation methods

The central peaks result from the correlation of the signal with itself, shifted by 0 μm, and hence it should not be viewed as a sign of periodicity. In the excitatory graphs, no other outstanding

peaks are visible, which suggests a non-periodic disposition of spines across the dendrite. On the contrary, in the inhibitory synapse's plots, there are noticeable peaks evenly spaced, at least for branches 1 and 4. This could indicate a periodicity of inhibitory synapses, however the scarce inhibitory data for each branch could be biasing the autocorrelation results, and hence, branches with more inhibitory synapses should be considered to support this observation.

4.3.8 Excitatory density around inhibitory synapses

Inhibitory synapses can be positioned in the neighborhood of spines to block their effect. Depending on the side where they are formed, the effect can be stronger (the proximal side) or weaker (the distal side). To evaluate if there was a tendency in positioning spines and and inhibitory synapses in a specific order, the density of spines were evaluated in a surrounding length of 40 μm surrounding inhibitory puncta. The results are shown in Figure 4.23.

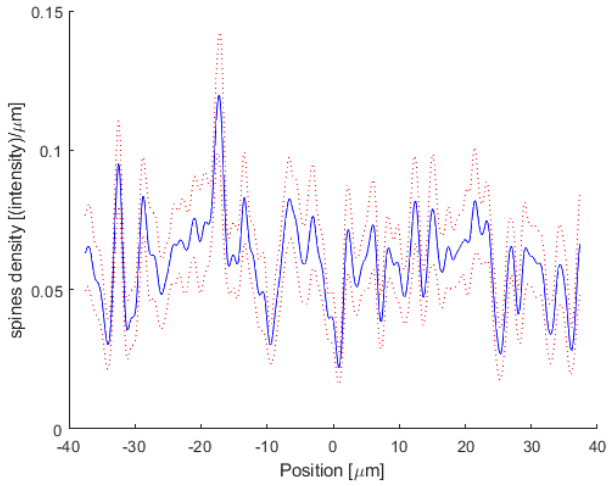


Figure 4.23: Spine density profile around inhibitory synapse for multiple branches of one cell

The graph is highly irregular and do not follow a distinctive behavior. No pattern was found to underlie spine formation in inhibitory sites, or vice-versa.

4.3.9 Mean inhibitory intensity *vs* excitatory density

Given that dendrites are more strongly populated with spines than inhibitory synapses, and that the number of inhibitory synapses does not appear to correlate with the one of their excitatory counterparts, it was hypothesized that the inhibitory/excitatory balance could be controlled differently for each type of synapse. For instance, cells could in-crease the excitatory stimuli in their dendrites by forming more spines, but increasing the inhibitory stimuli by increasing the intensity of inhibitory synapses. To test this hypothesis, the mean inhibitory intensity was plotted against spine density, as shown in Figure 4.24.

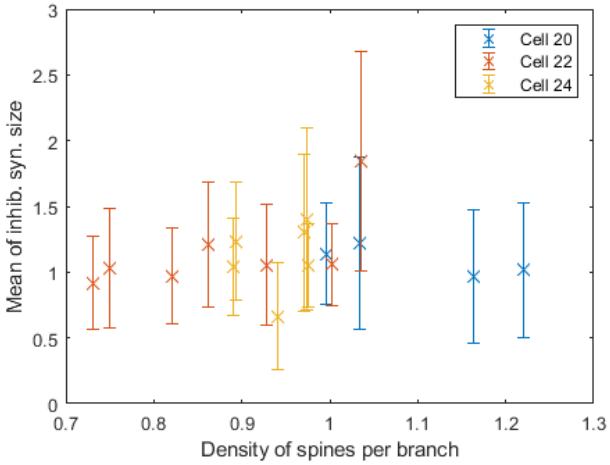


Figure 4.24: Variation of the mean inhibitory intensity against the density of spines per branch.

The variance of the mean intensities is too large to infer any concrete conclusion. If that was not the case and the centers remained in the same position, an horizontal linear correlation could be investigated, but even proven significant, it would oppose the previous hypothesis, as inhibitory branch intensity would appear to not depend on spine density.

4.3.10 Inhibitory intensity *vs* distance to nearest neighbor

To study how the position of inhibitory synapses affects each other intensities, the graph of Figure 4.25 was made, in which inhibitory intensity is plotted against distance to the nearest neighbor, for one cell.

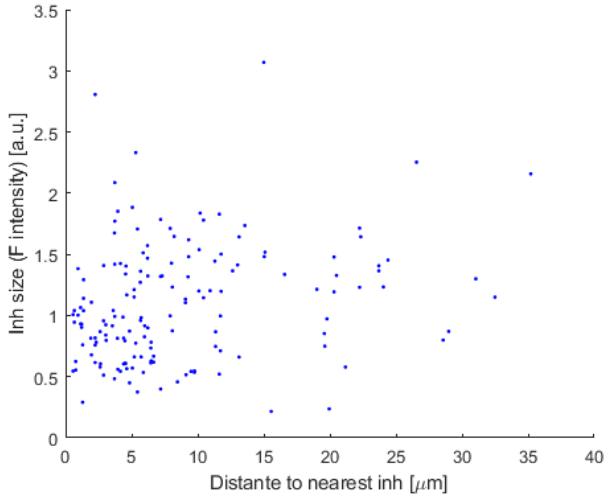


Figure 4.25: Influence of the position of the nearest inhibitory synapse on inhibitory intensity

Although points do not appear to follow a strictly random behavior, no strong evidence of proportionality was found between the two variables.

Chapter 5

Conclusions and Future Work

Dendritic spines have been postulated to underlie cognitive processes such as learning and memory, as well as several neurodegenerative diseases, including Alzheimer's disease, schizophrenia, intellectual disability, and autism spectrum disorders. Although their role is not completely understood, recent studies suggest that their morphology and distribution are correlated with their function. These characteristics can be captured by imaging cultures of neurons and digitally analyzing the produced images. In this Thesis, three advanced computational tools are proposed, capable of analyzing fluorescence microscopy image data to offer a detailed characterization of spines. These techniques are fully automatic and evaluate objectively the shape, dimension and locations of spines, as well as their relationship with shaft synapses.

In the course of this project, a broad set of scientific domains was explored and used towards the problematic of spines. In the geometrical method, novel 3D image processing techniques, offered by the latest version of MATLAB's repertoire, were studied and applied. These include the thinning algorithm of the detection module, as well as the volume characterization in the measuring module. Others were adapted from existing functions, such as the anisotropic filters in both preprocessing and dendrite segmentation modules. A third group was created, namely the tubular evaluator in the dendrite segmentation module, and the Rayburst tracing and plane delimitation algorithms in the spine segmentation module. Apart from these, graph theory and spatial geometry were studied for the detection and spine segmentation modules, respectively. In the machine learning method, modern artificial intelligence areas were delved into to produce a supervised method of identifying spines in 2D images. Moreover, the Python programming language was learned by the author and used to write the convolutional neural network and all associated routines. In the Synapse Toolbox, subjects from signal processing subjects, such as periodicity estimation based on autocorrelations, were applied to extract information from the provided data. With its broad scope, this Thesis granted a major learning opportunity for the student, who took it to contribute to the study of spines and expand his knowledge on different domains.

The first contribution is presented as the geometrical method, which is capable of characterizing the shape, dimension and locations of spines in 3D. In this method, it is shown that dendrites can be segmented based on their intensity and curvature, depicted by the second derivative content of the image. The segmentation follows a cylinder enhancement metric, proposed by [45], with anisotropic filters, to compensate for the resolution mismatch of the image stack across different axis. This metric, however, was proven to be insufficient for enhancement the dendrites without losing the necks of spines. Therefore, an intermediate result, originally designed to limit the volume in which tubular structures can be found, was used as the final segmentation result. Since it has a higher tolerance to irregular structures, this mask (Hessian mask) successfully preserves the necks, but is more susceptible to noise, being larger in noisy regions. The preservation of necks is a challenge in dendrite segmentation, and often implies intricate tactics to be addressed. Contrarily to spine segmentation, dendrite segmentation was not the focus of this Thesis, and therefore this simpler strategy was chosen to segment the dendrites, which undermined subsequent phases in certain regions but correctly segmented the structures in the ones cleared from noise.

The enhancement result was also kept to correct situations where the algorithm might treat the background as a spine, due to abnormalities in the mask. As the initial enhancement method only targets cylinders with a circular cross section, a set of new variations was created to consider multiple cross sections and adapt to the thickness of the neurites. Ultimately, an elliptical enhancement metric was chosen, which yielded a low noise and high intensity image, further converted into a second mask (tubular mask) by a simple threshold technique.

From the Hessian mask, spines were detected by first reducing the foreground to a skeleton, and then manipulating its branchpoints and endpoints until they defined the dendrite shaft and spines, respectively. The thinning phase was observed to be fast and computationally light, contrarily to what previous studies suggested for 3D images [41]. The routine is part of MATLAB's latest capabilities, and it was proven reliable for the thinning of dendrites.

Following thinning, the branchpoint and endpoint relationships in the skeleton were found by a novel algorithm as well. In this algorithm, the branchpoints and endpoints are removed, reducing the skeleton to isolated "bones". Then, each of these points is matched against a direct neighborhood of each bone tip. Points matched with the same bone are deemed connected. This method worked for every case and allowed the reliable representation of the skeleton into a graph, which is easier to manipulate.

Using the graph, the shaft can be found by determining its longest path. Hence, the tracing of the entire dendrite is avoided, which would be computationally expensive and subjected to many shaft irregularities. The only obstacle for its calculation is the presence of cycles, which would make the longest path infinite. These can be removed, however, by searching on each branchpoint for a path that both leaves and enters it, and reducing it to one of its points, transferring to it all the connections from the rest.

Spines can then be regarded as all the points directly connected to the shaft. This approach may exclude double spines from being detected, but ensures every spine is detected just once, in spite of how many points the thinning might produce in their place. After checking if these points

belong to the tubular mask, they were pushed away from their respective shaft branchpoints, until they were near the surface of the foreground. This step improved the success rate of the tracing phase, allowing more spines to be segmented in later stages.

From all the spines marked in the ground truth image, 45% were identified by the automatic detection method. Due to the disparity between the Hessian mask and the ground truth, several were detected outside the ground truth's foreground, and therefore were not counted as success cases. For instance, the dilation of the ground truth with a spherical element of 5 voxels of diameter would increase this number to 61%. The mask also made the detection unfeasible in regions populated with noise, as it became larger and agglomerated spines with the shaft. Furthermore, this result does not distinguish types of spines. In reality, stubby spines are more difficult to detect than their thin or mushroom counterparts, which is the reason that some studies consider the latter types or present a much lower accuracy to stubby [34]. This result encompasses all types, in a mask highly susceptible to noise and with no tolerance between it and the ground truth.

Spines can be extracted if their neck is found. Previous studies trace the dendritic shaft to find spine necks. In this project, spines are traced, from their detected points, to find their necks. An innovative tracing technique combining 3D, 2D and 1D Raybursts was developed to this end. By tracing spines instead of the shaft, this approach greatly reduces the tracing duration, avoids all irregularities that may be found across the large extent of the shaft, and does not compromise other spines if at any moment the tracing fails to continue. Moreover, no initial direction or center point is needed, since two opposite tracings are performed for each spine (forcing one to reach the neck) and each can adapt its path for any position inside the spine. A original neck identification method is also proposed, which considers the difference of angle and position of the last tracing iteration, as well as the iterations, mask and image limits. It can further be adjusted to best adapt to the properties of the image if needed. All spines that were detected in the considered region of interest were correctly traced by this method.

Following neck detection, a new method of separating shaft from spine was designed. It only requires an approximate neck point and the direction towards the spine, as it works exclusively by analyzing the cross section of the foreground. Based on its shape, it applies rotations and translations to an imaginary plane, initially centered around the neckpoint, in order to fit it precisely between the spine and the shaft. All spines traced in the region of interest were correctly delimited by this method. It can also save premature and late tracing endings, which may happen in cases where spines are too long for the tracing to reach the neck, or if the tracing fails to identify it and ends in a close region on the shaft.

Once limited, spines are extracted by a binary region growing method, beginning inside each spine and ending on their mask and neck limits. In spite of having a voxel limit to discard cases where the delimitation failed, all spines in the region of interest were correctly extracted.

To counteract the undesirable high thickness the dendrite mask might have, a grayscale region growing method is also available, which acts after the spine extraction to reduce their size based in the intensities of their interior. As some regions in the initial image were correctly segmented, this option was not used in the examples shown in this work.

The segmentation of spines expressed an F1 score of 34%. This module was, however, evaluated on the automatically detected spines described above, which had an already low accuracy level (F1 score equal to 49%). Thus, this value reflects, to a high extent, all the misdetections, which either did not provided points for some spines, or provided unsuitable points, such as ones inside the shaft and far from the spine, which could not be saved in the tracing nor delimitation phases. Moreover, even in the correctly determined spine points, if local noise is present, the mask can be enlarged to a degree where spines may not be distinguished from the shaft. Such cases are sure to sabotage the tracing phase, preventing more spines to be segmented. As the misdetections are mainly due to the enlarged mask as well, the usage of a reliable dendritic mask is sure to greatly improve the accuracy of both the detection and spine segmentation modules. As an alternative, in cases where no different mask can be provided, if a manual annotation of spines was to be performed, it would also reduce the misdetections. This would improve the accuracy of the segmentation, at the cost of turning the method semi-automatic.

The last phase of the geometrical method consisted in quantifying the segmentation results. Four properties were chosen: volume, surface area, solidarity and centroid proximity. Volume effectively characterizes the dimension of spines, and can be used together with area to infer their shape as well. For instance, both measures can be used to estimate the radius of an equivalent sphere, with the difference between radius expressing how similar the structure is to an actual sphere. Solidity takes another approach at estimating shape by measuring how much an equivalent convex shape would be filled by the spine. For instance, it can be used as a descriptor of how thin the neck is compared to the head, since a high disproportion would yield a high volume of empty space inside the equivalent shape. Centroid proximity tells where most of the volume concentrates in respect to the shaft, which can be used to estimate where the spine head is. All the results were verified visually, and are proposed as a set of new metrics capable of quantifying the morphology and dimension of spines.

The second contribution is presented as the machine learning method, which is capable of detecting the locations of spines in 2D. Instead of employing a thinning method, this component follows a deep learning algorithm for spine classification. Since there are few to none spine datasets, an interactive questionnaire is proposed, which selects windows out of the dendrite image and asks the user to classify them. The selection considers different probabilities depending on the intensity levels of the image, to account for the disproportionality between background and spine pixels. In the end, the longest group of windows is trimmed so that each class has the same number of samples. This method is easy to operate, reduces time in comparison to a completely random selection, and provides an objective way of selecting spine locations.

To account for a possibly small image, or for one having few spines, a replication method was designed, applying rotations and horizontal flips to the windows. These transformations are applied randomly, which ensures an unbiased replication process, and can be set to increase the dataset to any desirable number of samples.

The dataset is then used to train a machine learning model to classify spines and test its performance. A convolutional network with three convolution layers was built for this effect. It yielded an 87% accuracy on the test set, composed of windows, and it was then chosen to predict spines on complete images. The prediction works by first cropping the image in small windows, and using the network on each one of them. The probabilities outputted by the neck are then resized to match the original image. The last phase is a post-processing step which eliminates small and non-circular noise.

From all the spines marked in the ground truth image, 44% were identified (nearly the same as the 45% value from the geometrical detection). Similarly to the geometrical detection, several spines were detected in the periphery of the ground truth spines, excluding them from the calculation. No spine types were distinguished, and since stubby spines are more easily confused with the shaft, the low performance of the algorithm in this type of spines lowered the overall accuracy. The main reason justifying this result, however, is attributed to the dataset. Deep learning networks often require thousands of images to be reliable. Instead, the dataset presented was made with only 70 windows. Although their number was augmented, the original information was still drastically low, which may not have been sufficient to prepare the model for a diverse set of spines. Thus, it is deduced that a larger set of images would increase the performance greatly.

The final contribution is presented as the Synapse Toolbox, capable of correlating spine and shaft data gathered by fluorescence images. This component is prepared to accept data including the distance from synapses to the beginning of their branch, their intensity and their branch's length. The data is kept in the general-use .csv files, named with the cell, dendritic region and branch their synapses belong to. An interactive selection is proposed to load the data according to the desired cells, regions and branches, facilitating the data handling by the user. From this component, it was concluded that synapses, mainly inhibitory, express some periodicity, opposed to the idea of being distributed randomly across the dendrites. Synapse intensities, normalized by branch length, were found to be roughly constant between branches, especially for inhibitory synapses. Exceptions were encountered in spines, some larger branches exhibited a lower density than smaller branches. In many cells, a correlation between the number of inhibitory and excitatory synapses was also found, suggesting a close relationship between spines and other synaptic sites within the same cell.

Described above is the context of the work, the areas of study that were explored by the author, the outlining of each proposed method and the highlight of their results. The impact of these contributions into the research panorama around spines is mentioned in the next section, as well as their future directions suggested by the author.

5.1 Achievements

Despite the central role of spines in many neurological processes, and the broad set of imaging techniques able to depict them, most of their research is still performed in laboratories without the expertise for advanced image analysis. This leads to an ineffective and inefficient usage of the image's information, since not all data is accessed, and the part that is arises from a laborious, time consuming and subjective manual analysis. This work offers three free, open-source computational tools to aid in the detailed analysis of dendritic spines. Images, image stacks or plain spine measurements can be analyzed to characterize their physicality. All tools are automatic, and offer the detection, segmentation and measuring of spines with objective metrics. The programs are also organized into modules, making their selection easy, and facilitating further improvements. Other methodologies and variables can also be easily integrated if desired, such as a manual segmentation phase or dendrite segmentation mask. With these methods, several ideas were tested and validated successfully against the problems presented. New characteristics about the behavior of spines were also discovered, unraveling some aspects of their nature. By implementing new ideas with state of the art techniques, the programs provide a functional workbench that can be promptly used in the study of these structures.

5.2 Future Work

Regarding new directions and improvements, a set of suggestions are described below for each component.

In the geometrical method, as emphasized, the improvement of the dendrite segmentation phase would greatly benefit the detection and segmentation of spines. Efforts should be directed towards having an accurate representation of dendrites with spine necks, allowing for more spines to be identified, and each more accurately. As far as new directions, the interactive measurement of lengths in the extracted 3D spines would be an interesting addition, since it would allow users to make their own measurements, easily and without further coding. This could be made in a software such as Blender, by providing the spine masks calculated in MATLAB and programming the interactive tools through its Python API.

In the case of the machine learning approach, the expansion of the dataset would be the primal suggestion. New images could be used to generate more windows through the dataset creation method, or an external spine dataset could be used directly in the training and testing of the neural network. The network could also experiment new variations. For instance, transfer learning could be incorporated by using classification models trained on generic datasets. This could improve the identification of transversal image features without the need for a dendrite specific database. There are also quite some possibilities to adjust the networks layers and parameters to the spine classification problem, which could also be tested and empirically explored.

Regarding the synapse statistical toolbox, the function repertoire could be expanded based on new hypothesis regarding the behavior of spines. More data could also be gathered as it would turn calculation such as the periodicity estimate more reliable.

Although being simply operated, each component would benefit for the construction of a graphical user interface as well. Since the components are written in MATLAB and Python, this feature would eliminate the need for fundamental programming skills in either language, making it accessible to more members of the research community.

These measures would expand the capabilities of the proposed method, further contributing to the characterization of spines, as well as their relationship within the dendrites of neurons.

Appendix A

Biology of Dendritic Spines

Disorder	Protein involved	Role in dendritic spine morphogenesis	Reference
Alzheimer's disease	Kalirin	↑ spine size and density	[55]
	ApoE	↓ spine density	[56]
	Preselinin-1	↑ spine size and density	[57]
	Drebrin A	↑ spine size and density	[58]
	Calcineurin (PP2B)	↓ spine density	[59]
Schizophrenia	DISC1	↑ or ↓ spine size and density	[60]
	NRG1	↑ spine size and density	[61]
	ErbB4	↑ spine size and density	[62]
	Kalirin	↑ spine size and density	[55]
Fragile X Syndrome	FMRP	↑ spine density	[63]
Austism Spectrum Disorders	Neuroligin-3	↑ spine density	[64]
	Neuroligin-4	↑ spine density	[65]
	Neurexin1	↑ spine density	[66]
	Shank3	↑ spine density	[67]
	Shank2	↑ spine size	[68]
	Epac2	↓ spine size and stability	[69]

Figure A.1: Spine associated proteins whose genes are altered in a set of neurodegenerative disorders.

References

[1] Francisco López-Muñoz, Jesús Boya, and Cecilio Alamo. Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal. *Brain Research Bulletin*, 70(4-6):391–405, 10 2006. URL: http://linkinghub.elsevier.com/retrieve/pii/S0361923006002334, doi:10.1016/j.brainresbull.2006.07.010.

[2] Robert Moore. *Organization of the Human Circadian System*. 1992.

[3] Esther A. Nimchinsky, Bernardo L. Sabatini, and Karel Svoboda. Structure and Function of Dendritic Spines. *Annual Review of Physiology*, 64(1):313–353, 3 2002. URL: http://www.annualreviews.org/doi/10.1146/annurev.physiol.64.081501.160008, doi:10.1146/annurev.physiol.64.081501.160008.

[4] Shengxiang Zhang, Jiangbi Wang, and Lei Wang. Structural plasticity of dendritic spines. *Frontiers in Biology*, 5(1):48–58, 2 2010. URL: http://link.springer.com/10.1007/s11515-010-0011-z, doi:10.1007/s11515-010-0011-z.

[5] Myrrhe van Spronsen and Casper C. Hoogenraad. Synapse Pathology in Psychiatric and Neurologic Disease. *Current Neurology and Neuroscience Reports*, 10(3):207–214, 5 2010. URL: http://link.springer.com/10.1007/s11910-010-0104-8, doi:10.1007/s11910-010-0104-8.

[6] Kwok-On Lai and Nancy Y. Ip. Structural plasticity of dendritic spines: The underlying mechanisms and its dysregulation in brain disorders. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1832(12):2257–2263, 12 2013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925443913002779, doi:10.1016/j.bbadis.2013.08.012.

[7] D.L. Dickstein, C.M. Weaver, J.I. Luebke, and P.R. Hof. Dendritic spine changes associated with normal aging. *Neuroscience*, 251:21–32, 10 2013. URL: http://linkinghub.elsevier.com/retrieve/pii/S0306452212010093, doi:10.1016/j.neuroscience.2012.09.077.

[8] Heike Hering and Morgan Sheng. Dentritic spines : structure, dynamics and regulation. *Nature Reviews Neuroscience*, 2(12):880–888, 12 2001. URL: http://www.nature.com/articles/35104061, doi:10.1038/35104061.

[9] Saman Ebrahimi and Shigeo Okabe. Structural dynamics of dendritic spines: Molecular composition, geometry and functional regulation. *Biochimica et Biophysica Acta (BBA) - Biomembranes*, 1838(10):2391–2398, 10 2014. URL: https://linkinghub.elsevier.com/retrieve/pii/S0005273614002119, doi:10.1016/j.bbamem.2014.06.002.

[10] Carlo Sala and Menahem Segal. Dendritic Spines: The Locus of Structural and Functional Plasticity. *Physiological Reviews*, 94(1):141–188, 1 2014. URL: http://www.physiology.org/doi/10.1152/physrev.00012.2013, doi:10.1152/physrev.00012.2013.

[11] Miquel Bosch and Yasunori Hayashi. Structural plasticity of dendritic spines. *Current opinion in neurobiology*, 22(3):383–8, 6 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/21963169http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4281347, doi:10.1016/j.conb.2011.09.002.

[12] Myrrhe van Spronsen and Casper C. Hoogenraad. Synapse Pathology in Psychiatric and Neurologic Disease. *Current Neurology and Neuroscience Reports*, 10(3):207–214, 5 2010. URL: http://link.springer.com/10.1007/s11910-010-0104-8, doi:10.1007/s11910-010-0104-8.

[13] Kevin F. H. Lee, Cary Soares, and Jean-Claude Béïque. Examining Form and Function of Dendritic Spines. *Neural Plasticity*, 2012:1–9, 2012. URL: http://www.hindawi.com/journals/np/2012/704103/, doi:10.1155/2012/704103.

[14] Nathalie L Rochefort and Arthur Konnerth. Dendritic spines: from structure to in vivo function. *EMBO reports*, 13(8):699–708, 7 2012. URL: http://embor.embopress.org/cgi/doi/10.1038/embor.2012.102, doi:10.1038/embor.2012.102.

[15] Alfredo Rodriguez, Douglas B. Ehlenberger, Dara L. Dickstein, Patrick R. Hof, and Susan L. Wearne. Automated Three-Dimensional Detection and Shape Classification of Dendritic Spines from Fluorescence Microscopy Images. *PLoS ONE*, 3(4):e1997, 4 2008. URL: https://dx.plos.org/10.1371/journal.pone.0001997, doi:10.1371/journal.pone.0001997.

[16] John R. Hughes. Post-Tetanic Potentiation. *Physiological Reviews*, 38(1):91–113, 1 1958. URL: http://www.physiology.org/doi/10.1152/physrev.1958.38.1.91, doi:10.1152/physrev.1958.38.1.91.

[17] S. F. Cooke. Plasticity in the human central nervous system. *Brain*, 129(7):1659–1673, 7 2006. URL: https://academic.oup.com/brain/article-lookup/doi/10.1093/brain/awl082, doi:10.1093/brain/awl082.

[18] M. C. Ashby, S. R. Maier, A. Nishimune, and J. M. Henley. Lateral Diffusion Drives Constitutive Exchange of AMPA Receptors at Dendritic Spines and Is Regulated by Spine Morphology. *Journal of Neuroscience*, 26(26):7046–7055, 6 2006. URL: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.1235-06.2006, doi:10.1523/JNEUROSCI.1235-06.2006.

[19] David Holcman and Antoine Triller. Modeling Synaptic Dynamics Driven by Receptor Lateral Diffusion. *Biophysical Journal*, 91(7):2405–2415, 10 2006. URL: http://linkinghub.elsevier.com/retrieve/pii/S0006349506719567, doi:10.1529/biophysj.106.081935.

[20] Daniel Choquet. Fast AMPAR trafficking for a high-frequency synaptic transmission. *European Journal of Neuroscience*, 32(2):250–260, 7 2010. URL: http://doi.wiley.com/10.1111/j.1460-9568.2010.07350.x, doi:10.1111/j.1460-9568.2010.07350.x.

[21] R. Araya, J. Jiang, K. B. Eisenthal, and R. Yuste. The spine neck filters membrane potentials. *Proceedings of the National Academy of Sciences*, 103(47):17961–17966, 11 2006. URL: http://www.pnas.org/cgi/doi/10.1073/pnas.0608755103, doi:10.1073/pnas.0608755103.

[22] Jocelyn J Lippman Bell, Tamar Lordkipanidze, Natalie Cobb, and Anna Dunaevsky. Bergmann glial ensheathment of dendritic spines regulates synapse number without affecting spine motility. *Neuron glia biology*, 6(3):193–200, 8 2010. URL: http://www.ncbi.nlm.nih.gov/pubmed/21044397http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3244562, doi:10.1017/S1740925X10000165.

[23] Nikolai Medvedev, Victor Popov, Christian Henneberger, Igor Kraev, Dmitri A Rusakov, and Michael G Stewart. Glia selectively approach synapses on thin dendritic spines. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 369(1654):20140047, 10 2014. URL: http://www.ncbi.nlm.nih.gov/pubmed/25225105http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4173297, doi:10.1098/rstb.2014.0047.

[24] Mario M Dorostkar, · Chengyu Zou, Lidia Blazquez-Llorca, and Jochen Herms. Analyzing dendritic spine pathology in Alzheimer's disease: problems and opportunities. *Acta Neuropathol*, 3:1–19, 2015. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4469300/pdf/401_2015_Article_1449.pdf, doi:10.1007/s00401-015-1449-5.

[25] Glenn T. Konopaske, Nicholas Lange, Joseph T. Coyle, and Francine M. Benes. Prefrontal Cortical Dendritic Spine Pathology in Schizophrenia and Bipolar Disorder. *JAMA Psychiatry*, 71(12):1323, 12 2014. URL: http://archpsyc.jamanetwork.com/article.aspx?doi=10.1001/jamapsychiatry.2014.1582, doi:10.1001/jamapsychiatry.2014.1582.

[26] Yong Zhang, Kun Chen, Matthew Baron, Merilee A Teylan, Yong Kim, Zhihuan Song, Paul Greengard, and Stephen T C Wong. A Neurocomputational Method for Fully Automated 3D Dendritic Spine Detection and Segmentation of Medium-sized Spiny Neurons. 50(4):1472–1484, 2010. URL: http://www.bitplane.com/, doi:10.1016/j.neuroimage.2010.01.048.

[27] J.J. Mancuso, Y. Chen, X. Li, Z. Xue, and S.T.C. Wong. Methods of dendritic spine detection: From Golgi to high-resolution optical imaging. *Neuroscience*, 251:129–140, 10 2013. URL: http://linkinghub.elsevier.com/retrieve/pii/S0306452212003193, doi:10.1016/j.neuroscience.2012.04.010.

[28] Martin L. Feldman and C. Dowd. Loss of dendritic spines in aging cerebral cortex. *Anatomy and Embryology*, 148(3):279–301, 1975. URL: http://link.springer.com/10.1007/BF00319848, doi:10.1007/BF00319848.

[29] James E. Reilly, Hugo H. Hanson, Mónica Fernández-Monreal, Susan L. Wearne, Patrick R. Hof, and Greg R. Phillips. Characterization of MSB Synapses in Dissociated Hippocampal Culture with Simultaneous Pre- and Postsynaptic Live Microscopy. *PLoS ONE*, 6(10):e26478, 10 2011. URL: https://dx.plos.org/10.1371/journal.pone.0026478, doi:10.1371/journal.pone.0026478.

[30] Alessio Attardo, James E. Fitzgerald, and Mark J. Schnitzer. Impermanence of dendritic spines in live adult CA1 hippocampus. *Nature*, 523(7562):592–596, 7 2015. URL: http://www.nature.com/articles/nature14467, doi:10.1038/nature14467.

[31] Natalya Korogod, Carl CH Petersen, and Graham W Knott. Ultrastructural analysis of adult mouse neocortex comparing aldehyde perfusion with cryo fixation. *eLife*, 4, 8 2015. URL: https://elifesciences.org/articles/05793, doi:10.7554/eLife.05793.

[32] T Hosokawa, D A Rusakov, T V Bliss, and A Fine. Repeated confocal imaging of individual dendritic spines in the living hippocampal slice: evidence for changes in length and orientation associated with chemically induced LTP. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 15(8):5560–73, 8 1995. URL: http://www.ncbi.nlm.nih.gov/pubmed/7643201.

[33] Christina M. Weaver, Patrick R. Hof, Susan L. Wearne, and W. Brent Lindquist. Automated Algorithms for Multiscale Morphometry of Neuronal Dendrites. *Neural Computation*, 16(7):1353–1383, 7 2004. URL: http://www.mitpressjournals.org/doi/10.1162/089976604323057425, doi:10.1162/089976604323057425.

[34] Ingrid Y. Y. Koh, W. Brent Lindquist, Karen Zito, Esther A. Nimchinsky, and Karel Svoboda. An Image Analysis Algorithm for Dendritic Spines. *Neural Computation*, 14(6):1283–1310, 6 2002. URL: http://www.ncbi.nlm.nih.gov/pubmed/12020447http://www.mitpressjournals.org/doi/10.1162/089976602753712945, doi:10.1162/089976602753712945.

[35] Jie Cheng, Xiaobo Zhou, Eric Miller, Rochelle M. Witt, Jinmin Zhu, Bernardo L. Sabatini, and Steven T.C. Wong. A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *Journal of Neuroscience Methods*, 165(1):122–134, 9 2007. URL: http://www.ncbi.nlm.nih.gov/pubmed/17629570http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1989684http://linkinghub.elsevier.com/retrieve/pii/S0165027007002294, doi:10.1016/j.jneumeth.2007.05.020.

[36] Yong Zhang, Kun Chen, Matthew Baron, Merilee A. Teylan, Yong Kim, Zhihuan Song, Paul Greengard, and Stephen T.C. Wong. A neurocomputational method for fully automated 3D dendritic spine detection and segmentation of medium-sized spiny neurons. *NeuroImage*, 50(4):1472–1484, 5 2010. URL: http://www.ncbi.nlm.nih.gov/pubmed/20100579http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2839064http://linkinghub.elsevier.com/retrieve/pii/S1053811910000704, doi:10.1016/j.neuroimage.2010.01.048.

[37] Firdaus Janoos, Kishore Mosaliganti, Xiaoyin Xu, Raghu Machiraju, Kun Huang, and Stephen T.C. Wong. Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical Image Analysis*, 13(1):167–179, 2 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/18819835http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2663851http://linkinghub.elsevier.com/retrieve/pii/S1361841508000832, doi:10.1016/j.media.2008.06.019.

[38] J. SON, S. SONG, S. LEE, S. CHANG, and M. KIM. Morphological change tracking of dendritic spines based on structural features. *Journal of Microscopy*,

241(3):261–272, 3 2011. URL: http://www.ncbi.nlm.nih.gov/pubmed/21223260http://doi.wiley.com/10.1111/j.1365-2818.2010.03427.x, doi:10.1111/j.1365-2818.2010.03427.x.

[39] Tiancheng He, Zhong Xue, Yong Kim, and Stephen T. Wong. Three-dimensional dendritic spine detection based on minimal cross-sectional curvature. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1639–1642. IEEE, 5 2012. URL: http://ieeexplore.ieee.org/document/6235891/, doi:10.1109/ISBI.2012.6235891.

[40] J. F. Evers, S. Schmitt, M. Sibila, and C. Duch. Progress in Functional Neuroanatomy: Precise Automatic Geometric Reconstruction of Neuronal Morphology From Confocal Image Stacks. *Journal of Neurophysiology*, 93(4):2331–2342, 4 2005. URL: http://www.ncbi.nlm.nih.gov/pubmed/15537815http://www.physiology.org/doi/10.1152/jn.00761.2004, doi:10.1152/jn.00761.2004.

[41] Erik Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77A(7):693–704, 3 2010. URL: http://doi.wiley.com/10.1002/cyto.a.20895, doi:10.1002/cyto.a.20895.

[42] Alejandro F. Frangi, Wiro J. Niessen, Koen L. Vincken, and Max A. Viergever. Multiscale vessel enhancement filtering. pages 130–137. Springer, Berlin, Heidelberg, 10 1998. URL: http://link.springer.com/10.1007/BFb0056195, doi:10.1007/BFb0056195.

[43] Karl Krissian, Grégoire Malandain, Nicholas Ayache, Régis Vaillant, and Yves Trousset. Model-Based Detection of Tubular Structures in 3D Images. *Computer Vision and Image Understanding*, 80(2):130–171, 11 2000. URL: https://www.sciencedirect.com/science/article/pii/S107731420090866X, doi:10.1006/CVIU.2000.0866.

[44] 1.1.2 Assumptions of Differentiability. Technical report. URL: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MORSE/diffgeom.pdf.

[45] Tim Jerman, Franjo Pernuš, Boštjan Likar, and Žiga Špiclin. Beyond Frangi: an improved multiscale vesselness filter. volume 9413, page 94132A. International Society for Optics and Photonics, 3 2015. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2081147, doi:10.1117/12.2081147.

[46] Shih-Feng Yang and Ching-Hsue Cheng. Fast computation of Hessian-based enhancement filters for medical images. *Computer Methods and Programs in Biomedicine*, 116(3):215–225, 10 2014. URL: https://www.sciencedirect.com/science/article/pii/S0169260714001680, doi:10.1016/J.CMPB.2014.05.002.

[47] Qing Qing Li and Zhigang Zhigang Deng. A Surface-Based 3-D Dendritic Spine Detection Approach From Confocal Microscopy Images. *IEEE Transactions on Image Processing*, 21(3):1223–1230, 3 2012. URL: http://www.ncbi.nlm.nih.gov/pubmed/21896386http://ieeexplore.ieee.org/document/6008641/, doi:10.1109/TIP.2011.2166973.

[48] *Biotechniques*. [Eaton Pub. Co.], 1983. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=3585997.

[49] Ekaterina Myasnikova, Svetlana Surkova, Lena Panok, Maria Samsonova, and John Reinitz. Estimation of errors introduced by confocal imaging into the data on segmentation gene expression in Drosophila. *Bioinformatics*, 25(3):346–352, 2 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/19052059http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2639076https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btn620, doi:10.1093/bioinformatics/btn620.

[50] Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data*, 4:170193, 12 2017. URL: http://www.nature.com/articles/sdata2017193, doi:10.1038/sdata.2017.193.

[51] Katherine L. Villa, Kalen P. Berry, Jaichandar Subramanian, Jae Won Cha, Won Chan Oh, Hyung-Bae Kwon, Yoshiyuki Kubota, Peter T.C. So, and Elly Nedivi. Inhibitory Synapses Are Repeatedly Assembled and Removed at Persistent Sites In Vivo. *Neuron*, 89(4):756–769, 2 2016. URL: http://www.ncbi.nlm.nih.gov/pubmed/26853302http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4760889https://linkinghub.elsevier.com/retrieve/pii/S0896627316000118, doi:10.1016/j.neuron.2016.01.010.

[52] Jyoti Mishra, Jean-Marc Fellous, and Terrence J. Sejnowski. Selective attention through phase relationship of excitatory and inhibitory input synchrony in a model cortical neuron. *Neural Networks*, 19(9):1329–1346, 11 2006. URL: http://www.ncbi.nlm.nih.gov/pubmed/17027225http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1815390http://linkinghub.elsevier.com/retrieve/pii/S0893608006001808, doi:10.1016/j.neunet.2006.08.005.

[53] Aruna Jammalamadaka, Sourav Banerjee, Bangalore S Manjunath, and Kenneth S Kosik. Statistical analysis of dendritic spine distributions in rat hippocampal cultures. *BMC Bioinformatics*, 14(1):287, 10 2013. URL: http://www.ncbi.nlm.nih.gov/pubmed/24088199http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3871014http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-287, doi:10.1186/1471-2105-14-287.

[54] Jon I. Arellano, Ruth Benavides-Piccione, Javier Defelipe, and Rafael Yuste. Ultrastructure of dendritic spines: correlation between synaptic and spine morphologies. *Frontiers in Neuroscience*, 1(1):131–143, 11 2007. URL: http://www.ncbi.nlm.nih.gov/pubmed/18982124http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2518053http://journal.frontiersin.org/article/10.3389/neuro.01.1.1.010.2007/abstract, doi:10.3389/neuro.01.1.1.010.2007.

[55] Zhong Xie, Deepak P. Srivastava, Huzefa Photowala, Li Kai, Michael E. Cahill, Kevin M. Woolfrey, Cassandra Y. Shum, D. James Surmeier, and Peter Penzes. Kalirin-7 Controls Activity-Dependent Structural and Functional Plasticity of Dendritic Spines. *Neuron*, 56(4):640–656, 11 2007. URL: http://www.ncbi.nlm.nih.gov/pubmed/18031682http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2118058http://linkinghub.elsevier.com/retrieve/pii/S0896627307007611, doi:10.1016/j.neuron.2007.10.005.

[56] Y Ji, Y Gong, W Gan, T Beach, D M Holtzman, and T Wisniewski. Apolipoprotein E isoform-specific regulation of dendritic spine morphology in apolipoprotein E transgenic mice and Alzheimer's disease patients. *Neuroscience*, 122(2):305–15, 2003. URL: http://www.ncbi.nlm.nih.gov/pubmed/14614898.

[57] S. Knafo, L. Alonso-Nanclares, J. Gonzalez-Soriano, P. Merino-Serrais, I. Fernaud-Espinosa, I. Ferrer, and J. DeFelipe. Widespread Changes in Dendritic Spines in a Model of Alzheimer's Disease. *Cerebral Cortex*, 19(3):586–592, 3 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/18632740https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhn111, doi:10.1093/cercor/bhn111.

[58] S. Knafo, L. Alonso-Nanclares, J. Gonzalez-Soriano, P. Merino-Serrais, I. Fernaud-Espinosa, I. Ferrer, and J. DeFelipe. Widespread Changes in Dendritic Spines in a Model of Alzheimer's Disease. *Cerebral Cortex*, 19(3):586–592, 3 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/18632740https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/bhn111, doi:10.1093/cercor/bhn111.

[59] Marlen Knobloch and Isabelle M. Mansuy. Dendritic Spine Loss and Synaptic Alterations in Alzheimer's Disease. *Molecular Neurobiology*, 37(1):73–82, 2 2008. URL: http://www.ncbi.nlm.nih.gov/pubmed/18438727http://link.springer.com/10.1007/s12035-008-8018-z, doi:10.1007/s12035-008-8018-z.

[60] Akiko Hayashi-Takagi, Manabu Takaki, Nick Graziane, Saurav Seshadri, Hannah Murdoch, Allan J Dunlop, Yuichi Makino, Anupamaa J Seshadri, Koko Ishizuka, Deepak P Srivastava, Zhong Xie, Jay M Baraban, Miles D Houslay, Toshifumi Tomoda, Nicholas J Brandon, Atsushi Kamiya, Zhen Yan, Peter Penzes, and Akira Sawa. Disrupted-in-Schizophrenia 1 (DISC1) regulates spines of the glutamate synapse via Rac1. *Nature Neuroscience*, 13(3):327–332, 3 2010. URL: http://www.nature.com/articles/nn.2487, doi:10.1038/nn.2487.

[61] C. S. Barros, B. Calabrese, P. Chamero, A. J. Roberts, E. Korzus, K. Lloyd, L. Stowers, M. Mayford, S. Halpain, and U. Muller. Impaired maturation of dendritic spines without disorganization of cortical cell layers in mice lacking NRG1/ErbB signaling in the central nervous system. *Proceedings of the National Academy of Sciences*, 106(11):4507–4512, 3 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/19240213http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2657442http://www.pnas.org/cgi/doi/10.1073/pnas.0900355106, doi:10.1073/pnas.0900355106.

[62] Bo Li, Ran-Sook Woo, Lin Mei, and Roberto Malinow. The Neuregulin-1 Receptor ErbB4 Controls Glutamatergic Synapse Maturation and Plasticity. *Neuron*, 54(4):583–597, 5 2007. URL: http://www.ncbi.nlm.nih.gov/pubmed/17521571http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2031848http://linkinghub.elsevier.com/retrieve/pii/S0896627307002607, doi:10.1016/j.neuron.2007.03.028.

[63] Claudia Bagni and William T. Greenough. From mRNP trafficking to spine dysmorphogenesis: the roots of fragile X syndrome. *Nature Reviews Neuroscience*, 6(5):376–387, 5 2005. URL: http://www.ncbi.nlm.nih.gov/pubmed/15861180http://www.nature.com/articles/nrn1667, doi:10.1038/nrn1667.

[64] Ben Chih, Shehla Khan Afridi, Lorraine Clark, and Peter Scheiffele. Disorder-associated mutations lead to functional inactivation of neuroligins. *Human Molecular Genetics*, 13(14):1471–1477, 7 2004. URL: http://www.ncbi.nlm.nih.gov/pubmed/15150161https://academic.oup.com/hmg/article-lookup/doi/10.1093/hmg/ddh158, doi:10.1093/hmg/ddh158.

[65] Irina Dudanova, Katsuhiko Tabuchi, Astrid Rohlmann, Thomas C. Südhof, and Markus Missler. Deletion of α-neurexins does not cause a major impairment of axonal pathfinding or synapse formation. *The Journal of Comparative Neurology*, 502(2):261–274, 5 2007. URL: http://www.ncbi.nlm.nih.gov/pubmed/17347997http://doi.wiley.com/10.1002/cne.21305, doi:10.1002/cne.21305.

[66] G. Roussignol, Fabrice Ango, Stefano Romorini, Jian Cheng Tu, Carlo Sala, Paul F Worley, Joël Bockaert, and Laurent Fagni. Shank Expression Is Sufficient to Induce Functional Dendritic Spine Synapses in Aspiny Neurons. *Journal of Neuroscience*, 25(14):3560–3570, 4 2005. URL: http://www.ncbi.nlm.nih.gov/pubmed/15814786http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.4354-04.2005, doi:10.1523/JNEUROSCI.4354-04.2005.

[67] Pascal Steiner, Michael J. Higley, Weifeng Xu, Brian L. Czervionke, Robert C. Malenka, and Bernardo L. Sabatini. Destabilization of the Postsynaptic Density by PSD-95 Serine 73 Phosphorylation Inhibits Spine Growth and Synaptic Plasticity. *Neuron*, 60(5):788–802, 12 2008. URL: http://www.ncbi.nlm.nih.gov/pubmed/19081375http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2671083http://linkinghub.elsevier.com/retrieve/pii/S0896627308008878, doi:10.1016/j.neuron.2008.10.014.

[68] Kevin M Woolfrey, Deepak P Srivastava, Huzefa Photowala, Megumi Yamashita, Maria V Barbolina, Michael E Cahill, Zhong Xie, Kelly A Jones, Lawrence A Quilliam, Murali Prakriya, and Peter Penzes. Epac2 induces synapse remodeling and depression and its disease-associated forms alter spines. *Nature Neuroscience*, 12(10):1275–1284, 10 2009. URL: http://www.ncbi.nlm.nih.gov/pubmed/19734897http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2754861http://www.nature.com/articles/nn.2386, doi:10.1038/nn.2386.

[69] Giorgio A. Ascoli. Neuroinformatics Grand Challenges. *Neuroinformatics*, 6(1):1–3, 3 2008. URL: http://www.ncbi.nlm.nih.gov/pubmed/18253866http://link.springer.com/10.1007/s12021-008-9010-5, doi:10.1007/s12021-008-9010-5.