M 2015

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

NEURONDYN: LIVE NEUROTRANSMITTER VESICLE MOVEMENT DYNAMICS IN LIVING NEURONS

HÉLDER DIOGO TEDIM MATOS MOREIRA
DISSERTAÇÃO DE MESTRADO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA BIOMÉDICA

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MEB - MESTRADO EM ENGENHARIA BIOMÉDICA **2014/2015**

A Dissertação intitulada

"NeuronDyn: Live neurotransmitter vesicle movement dynamics in living neurons"

foi aprovada em provas realizadas em 16-07-2015

o júri

Machado da Silva

Presidente Professor Doutor José Alberto Peixoto Machado da Silva
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da U. Porto

Hélder Filipe Pinto de Oliveira

Professor Doutor Hélder Filipe Pinto de Oliveira
Professor Auxiliar Convidado do Departamento de Engenharia Informática da Faculdade de Engenharia da U. Porto

Professor Doutor João Paulo Trigueiros da Silva Cunha
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da U. Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Hélder Diogo Tedim Matos Moreira

Autor - Hélder Diogo Tedim Matos Moreira

Faculdade de Engenharia da Universidade do Porto

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

NeuronDyn: Live Neurotransmitter Vesicle Movement Dynamics in Living Neurons

Hélder Diogo Tedim Matos Moreira

Mestrado em Engenharia Biomédica

Supervisor: Prof. Dr. João Paulo Cunha

Co-Supervisor: Prof.ª Dr.ª Paula Sampaio

July, 2015

NeuronDyn: Live Neurotransmitter Vesicle Movement Dynamics in Living Neurons

Hélder Diogo Tedim Matos Moreira

Mestrado em Engenharia Biomédica

Faculdade de Engenharia da Universidade do Porto

July, 2015

Contents

List of Figures

List of Tables

Abbreviations and Symbols

AD	Alzheimer’s Disease
ALS	Amyotrophic Lateral Sclerosis
ANN	Artificial Neural Network
AP	Action Potential
CLSM	Confocal Laser Scanning Microscopy
CNS	Central nervous System
FEUP	Faculdade de Engenharia da Universidade do Porto
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
GNN	Global Nearest Neighbour
GUI	Graphic User Interface
HD	Huntington’s Disease
HMM	Hidden Markov Models
IBMC	Instituto de Biologia Molecular e Celular
IFT	Intraflagellar Transport Dynein
KIF	Kinesin Superfamily Proteins
LoG	Laplacian of Gaussian
NT	Neurotransmitter
PD	Parkinson’s Disease
PNS	Peripheral Nervous System
SVM	Support Vector Machines
TN	True Negative
TP	True Positive

Chapter 1

Introduction

1.1 Context an Motivation

Communication between and inside cells is a vital function for human organism. In fact, intracellular transport of organelles is a fundamental process and despite being intensively studied, it is still an open challenge for researchers.

Neurons are highly differentiated cells composed of a cell body, dendrites and axon, and are responsible for transporting information from and to the brain. The communication between two neurons is established by endogenous chemical particles called neurotransmitters, which travel along neuron's axon aggregated in vesicles [1]. It is known that an abnormal transport of these vesicles is correlated with neurodegenerative diseases such as spastic paraplegia, Charcot Marie Tooth, amyotrophic lateral sclerosis (ALS), Alzheimer's, Huntington's and Parkinson's [2,3].

With the rapid evolution of microscopy techniques and computer science, it is now possible to acquire dynamic images of moving cells, including neurons, and automatically process them in order to extract a set of features that may considerably facilitate researchers' work. This project is being developed in a cooperation between BRAINlab group from INESC-TEC and IBMC.

1.2 Main Objectives

The resolution of the problem identified above depends on an extensive and strenuous analysis of microscopy films containing innumerous vesicles travelling along neuron's axon that must be done by one or more researchers. The main problem is that the image is not always in the best conditions for bare eye analysis, for example due to the presence of noise, lack of contrast or not good resolution.

As this project is a follow up to a master's thesis [4] developed at Faculdade de Engenharia da Universidade do Porto (FEUP), the main objective consists on improving the algorithm and GUI making it possible for researchers to better characterize vesicles' dynamics through the analysis of some important measures that will be described further in this report.

Some of the main improvements to the existent algorithm passes through a more robust classification, using an SVM based approach, and tracking algorithm, by optimizing the global nearest neighbor method. Also it is desirable to extract more features from the vesicles although its size makes this task extremely difficult once the majority of the features that can be used to characterize vesicles are related with its morphology. Another issue regards the difficulties that Matlab presents in processing videos with high resolution and duration. This problem could possibly be fixed by changing the programming language.

1.3 Contributions

The main improvements made to the algorithm were:

- New candidate detection method using adaptive threshold value.
- New method for segmentation of both candidates and objects selected in training step. Active contours approach selected instead of region growing which allows a better performance for datasets with different characteristics.
- New SVM based classification instead of ANN, used to improve computational time.
- Vesicle path can now be seen along time and not only on a static frame.

Two publications resulted from the present thesis: one local publication in 1st Doctoral Congress in Engineering which took place at FEUP, and an international publication in the 37th Annual International IEEE EMBS Conference, which will take place in Milan, Italy.

Hélder T. Moreira, Ivo M. Silva, Mónica Sousa, Paula Sampaio, João Paulo Silva Cunha. *Neurotransmitter Vesicle Movement Dynamics in Living Neurons*. In 1st Doctoral Congress in Engineering FEUP. Porto, Portugal, 2015. (see Appendix B)

Hélder T. Moreira, Ivo M. Silva, Mónica Sousa, Paula Sampaio, João Paulo Silva Cunha. *Neurotransmitter Vesicle Movement Dynamics in Living Neurons*. In 37th Annual International IEEE EMBS Conference. Milan, Italy, 2015. - Accepted for Publication (see Appendix A).

1.4 Work Structure

This thesis is divided in six main chapters, each one with an introductory note specifying the issue addressed in that chapter.

After this introductory chapter, it is presented a brief background review about the Nervous System, providing basic concepts of neuroscience as well as neurons properties for a better understanding of the following contents.

Chapter 3, Computational Analysis, is divided in several subtopics: the first on is focused on image acquisition methods, predominantly confocal microscopy, which was the technique used to obtain the images that are on the basis of this project; the second one addresses image processing and analysis, going from image restoration processes to feature extraction and classification; the third and last one, summarizes the most recent and commonly used tracking algorithms. Chapter 4, State of the Art, is directly related to the main problem addressed by this work, consisting on a general overview of several algorithms used in neurotransmitter vesicle tracking, as well as a detailed description of the most relevant ones. In Chapter 5, it is described the last version of the algorithm and explained in detail each step of the process. Finally, in Chapter 6 are drawn some conclusions about the whole work and highlighted the main improvements made.

Chapter 2

Nervous System

This chapter provides essential background about the nervous system and the interaction between nerve cells. Starting from general concepts of Neuroscience to detailed biological phenomena regarding neuronal transmission of information, it will offer the reader a better perception of the problem. This review is mainly supported by research papers, published in the last decade.

2.1 Neuroscience

Neuroscience can be defined as an interdisciplinary science that studies the nervous system from the neurons interactions to a complete neural network, focusing in different levels, such as molecular, cellular, structural, functional, computational and medical [5].

The first signs of human interest in Neuroscience are dated from as early as 7000 years ago, when humans started boring holes in each other's skulls with the aim not to kill but to cure. However, only many years later after the invention of the microscope in the nineteenth century, the nervous system was proved to being constituted by more than one type of cell [6]. In vertebrates, the nervous system is divided into central nervous system (CNS, composed of brain and spinal cord) and the peripheral nervous system (PNS) [7].

Today, to reduce the complexity of the problem, neuroscientists break neuroscience into smaller pieces, in which the size of the unit in study is called level of analysis. In ascending order of complexity, these levels are molecular, cellular, systems, behavioral, and cognitive [6].

All tissues and organs in the human body are constituted of cells. The specialized functions of each cell and how they interact determine the functions of organs. The brain is considered the most sophisticated and complex organ in nature. This way, in order to study its function, we must begin by learning how basic brain cells work individually and then see how they are assembled to work together [7].

2.2 Neurons

The human brain is composed of around one hundred billion neurons and one hundred trillion synapses [5]. All neurological processes are dependent on complex cell–cell interactions among single neurons as well as groups of related neurons. Neurons can be categorized according to their size, shape, neurochemical characteristics, location, and connectivity, which are important determinants of that particular functional role of the neuron in the brain. More importantly, neurons form circuits, and these circuits constitute the structural basis for brain function [8].

2.2.1 Anatomy

Neurons are generated from a special type of stem cells in a process called neurogenesis, which largely ceases during adulthood in humans. Generally, three functional classes of neurons make up the nervous system: afferent neurons, efferent neurons and interneurons. Despite some variations, neurons are generally composed by a cell body or soma, a dendritic arbor and an axon, being connected between them by synapses (Figures 2.1 and 2.2) [9].

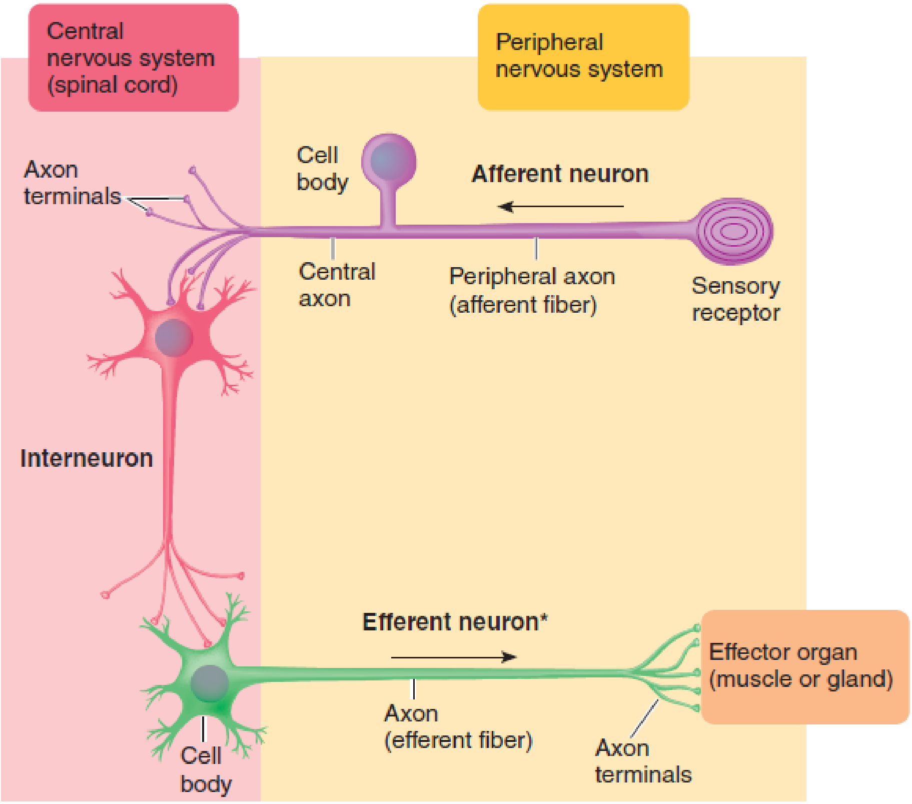


Figure 2.1: Structure and location of the three functional classes of neurons [9].

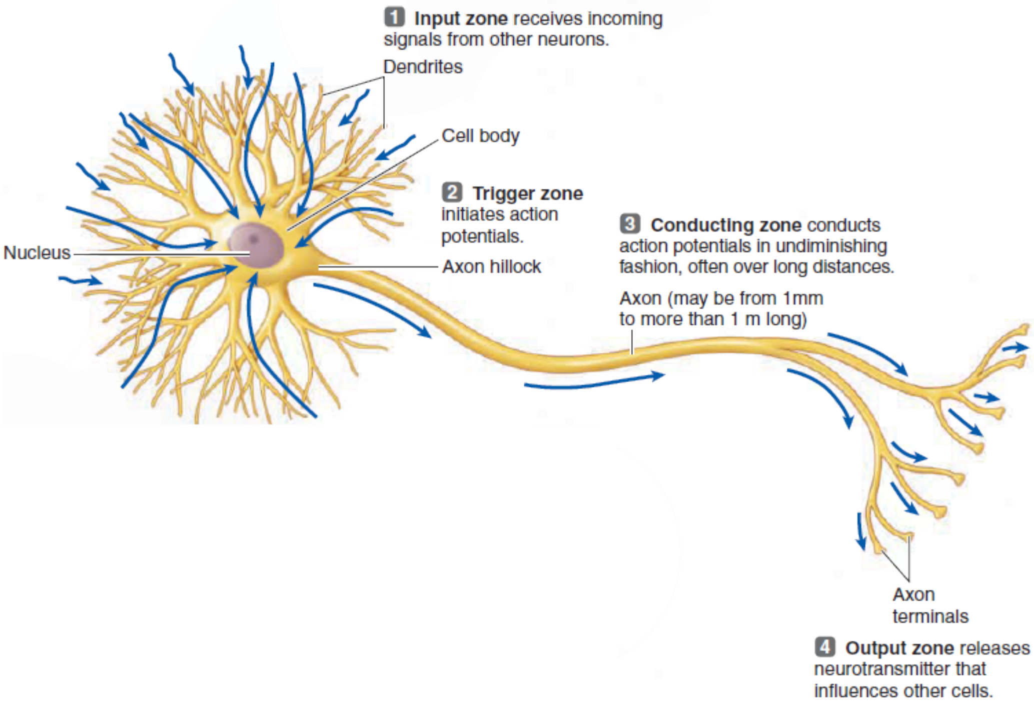

1 Input zone receives incoming signals from other neurons.
Dendrites
Cell body
2 Trigger zone initiates action potentials.
Nucleus
Axon hillock
3 Conducting zone conducts action potentials in undiminishing fashion, often over long distances.
Axon (may be from 1mm to more than 1 m long)
Axon terminals
4 Output zone releases neurotransmitter that influences other cells.

Figure 2.2: Anatomy of most common type of neuron [9].

Cell body contains the nucleus and organelles, and has numerous extensions (Dendrites) which project like antennae to increase the surface area available for receiving signals from other neurons. In most neurons, the plasma membrane of the dendrites and cell body contains protein receptors that bind chemical messengers from other neurons [10]. This way, the dendrites and cell body are the neuron's input zone, because these components receive and integrate incoming signals. This is the region where graded potentials are produced in response to triggering events, in this case, incoming chemical messengers [9–11].

The axon, or nerve fiber, is an elongated tubular extension that conducts electrical and chemical signals away from the cell body (anterograde transport) [12], but also into the cell body (retrograde transport) [13]. Axons vary in length from less than a millimeter to longer than a meter in neurons that communicate with distant parts of the nervous system or with peripheral organs [9].

2.2.2 Synapses

A neuron may terminate in one of three structures: a muscle, a gland, or another neuron. When a neuron terminates on a muscle or a gland, the neuron is said to innervate, or supply, the structure. By the other hand, if a neuron terminates on another neuron, the junction between them is called synapse [5].

There are two types of synapses: electrical and chemical, depending on the how the communications is established. In an electrical synapse, the information is transmitted through charge carrying ions which flow directly between the two neurons in both directions. Despite being extremely quick, these type of connections are relatively rare in human nervous system, appearing in the CNS, where they synchronize electrical activity in groups of neurons interconnected by gap junctions, and in specialized locations, such as the pulp of a tooth and the retina of the eye [9, 14].

2.2.3 Neurotransmitters

Until the beginning of the twentieth century, the nervous system was considered to be one continuous network (a syncytium), where every cell was in direct physical contact with the others. However, the pioneering studies of Ramon y Cajal revealed neurons as independent structures, which was confirmed many years later with the invention of electron microscopy. The main problem then was how nerve cells communicated with each other [8, 15].

Within the CNS, neurons communicate with other nerve cells but also with glands and muscles. In simpler animals, communication is mediated by hormones and growth factors that diffuse relatively long distances from secretory cells to target tissues. In humans, chemical messengers are secreted from specialized parts of neurons, called nerve terminals or nerve endings, to act on receptors in the membrane of neighboring target cells. These chemical messengers are called neurotransmitters [16].

Neurotransmitters (NT) are chemicals like glutamate, acetylcholine and dopamine. They travel in the interior of organelles called vesicles and are released, in response to a depolarizing action potential, by exocytosis of synaptic vesicles with the neuronal plasma membrane in the synaptic cleft, acting at the receptors of the postsynaptic neuron and triggering an action potential (AP). This process of neurotransmission is terminated by a number of processes including diffusion and metabolic degradation of the neurotransmitter, and desensitization of the receptors [17, 18].

2.3 Neuronal Intracellular Transport

The distinctive morphology of neurons, highly polarized cells with extended dendrites and axons, makes these cells extremely dependent on active intracellular transport. This way, transport of organelles, vesicles, proteins and RNA to every region of the neuron requires molecular motors that operate along the cellular cytoskeleton [1, 13].

Cargo movement occurs along axonal and dendritic thin tubes (diameter around 25nm) called microtubules [19]. The molecular motor involved in the transport of a certain type of cargo is directly related to the direction of the movement, for instance, kinesin motors are responsible for anterograde transport towards the cell periphery, while dynein motors are responsible for retrograde transport back to the cell body [9, 20] (Figure 2.3).

Whereas individual motors move unidirectionally along micro tubules, the directional transport of intracellular cargo is generally achieved through back-and-forth movements with an overall net directionality towards the desired destination. It has been hypothesized that the main advantages of this apparently inefficient behavior is to avoid eventual obstacles such as organelles or microtubule-associated proteins, or to provide a mechanism where the desired cargo destination is

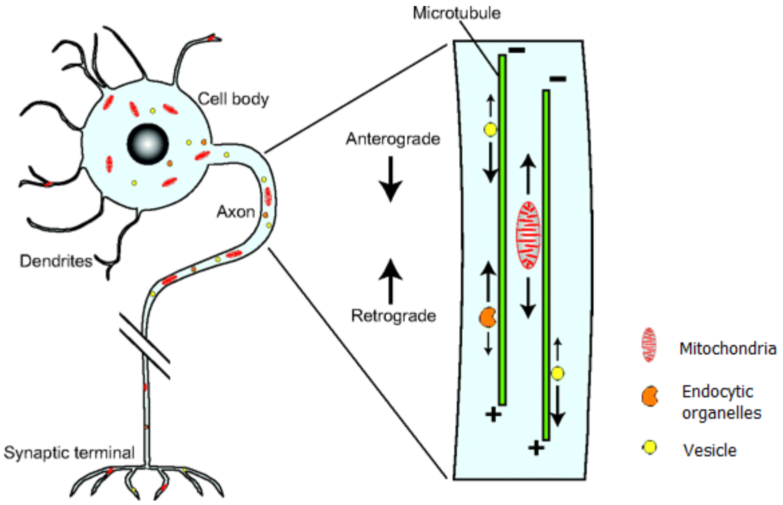


Figure 2.3: Anterograde and retrograde transport of organelles on the microtubules of the axon [21].

given by a sequence of instructions rather than by an 'all-or-nothing' decision determined by the initial directionality [22, 23].

We can separate intraneuronal transport in two types: Anterograde and Retrograde transport

Anterograde transport. Kinesin superfamily proteins (KIFs) are responsible for anterograde transport or plus-end directed [24]. The organelles moving in this direction are typically smaller but more numerous than those moving in the retrograde direction. Generally, there are two types of anterograde transport in the axon: fast transport of membranous organelles and slow transport of cytosolic proteins and cytoskeletal protein [1]. In terms of the fast transport, various cargo vesicles are conveyed by distinct KIFs. Cargos transported down the axon include, between others, synaptic vesicle precursors, active zone vesicles, and mitochondria (essential for energy supply) [19, 21, 25, 26].

Fast transport is bidirectional: many proteins that are distributed by fast anterograde transport are also returned in the retrograde direction. By opposition, proteins transported at slow rates are degraded when they reach their destination and are not detected in the retrograde component. [1, 13].

Retrograde transport. Retrograde transport or minus-end directed transport is mediated by dynein/dynactin complex which is responsible for cargo movement towards neuron's cell body. Despite existing numerous proteins in dynein family, only two of them (intraflagellar transport dynein (IFT) and cytoplasmatic dynein) are responsible for this kind of transport. A summary of neuronal transport is presented on Table 2.1 .

Table 2.1: Neurodegenerative diseases with axonal transport defects. Adapted from [2].

Mutated gene in patients	**Common protein name**	**Affected axonal transport transport-related process**
	Alzheimer's Disease and other dementias	
APP	Amyloid precursor protein	Unkown effect but shown to undergo axonal transport; Retrograde tranport of NGF;
PSEN1	Presenilin 1	Microtubule stabilization through GSK3 activation; Cargo binding to motor proteins;
	Parkinson's Disease and Perry syndrome	
SNCA	synuclein	Unknown effect but shown to undergo axonal transport;
PARK2	E3 ubiquitin-protein ligase parkin	Mitochondrial function
PINK1	Serine/threonine-protein kinase	Mitochondrial function
PARK7	Protein DJ1	Mitochondrial function
DCTN1	Dynactin subunit 1	Dynein complex function
	Huntington's Disease	

HIT	Huntingtin	Microtubule acetylation;
		Cargo binding to motor proteins;
		Kinesin binding to microtubules;
		Retrograde transport of BDNF (trophic support);

	Amyotrophic lateral sclerosis	
SOD1	Superoxide dismutase 1	Neurofilament phosphorylation and binding to motor proteins; Mitochondria binding to kinesin;
ALS2	Alsin	Endosomal trafficking;
VAPB	Vesicle-associated membrane protein-associated protein B/C	Endoplasmic reticulum to Golgi transfer;

2.4 Neurodegenerative Diseases and Intraneuronal Transport

Recent studies suggest that defects in axonal transport such as mutations in the molecular motors are potentially related with the degeneration of nervous cells [13]. In fact, it is proved that, for example, a disruption of anterograde transport (kinesin motor) is sufficient to provoke neurodegeneration [24]. Nevertheless only a few neurodegenerative diseases have been unswervingly related with kinesin motor malfunction, probably because of its functional redundancy (different elements from the kinesin family can transport the same cargo) [27].

By opposition, some studies link anomalies in retrograde transport directly to neurodegenerative disease [13]. In fact, defects in dynein-dynactin motor complex have already been linked to motor-neuron loss and muscle denervation in mouse models [28].

Alzheimer's disease (AD) is the most common form of dementia and generally affects elder people, causing global cognitive decline including a progressive loss of memory, orientation and reasoning [29]. AD is characterized by synaptic and neuronal loss, pathological accumulation of amyloid-beta peptide in senile plaques and formation of tangles of the microtubule-associated protein tau13 that inhibit axonal transport. Reduced axonal transport represents an early step of AD pathogenesis and may be used as a diagnosis method, once it is noticeable before the common AD symptoms [2,30–32].

Amyotrophic lateral sclerosis (ALS) is the most common adult-onset motor neuron disease and is characterized by the degeneration of cortical, bulbar and spinal motor neurons. This degeneration induces progressive muscle atrophy, paralysis and spasticity that eventually leads to respiratory failure [2]. Several studies have proved that swellings occur in the initial region of motor axons in patients with ALS and that these swellings contain vesicles, lysosomes, mitochondria, between others. These anomalies suggest axonal transport defects [33].

Huntington's disease (HD) is one of the most common Polyglutamine diseases. This disease is an inherited adult-onset neurodegenerative disorder caused by the expansion of a CAG tract in particular genes, leading to the loss of selected neuronal populations and the formation of aggregates that sequester essential cellular proteins. Huntington's disease is characterized by muscle incoordination, cognitive decline and dementia. It occurs when the number of CAG repeats in the coding region of the gene huntingtin (HTT) is above 36 [2,34,35].

Parkinson's disease (PD) is characterized by the degeneration of dopaminergic neurons in the substantia nigra, which causes rigidity, shaking and gait disturbance. The typical characteristic lesions of the disease are the Lewy body inclusions, which are composed of hyperphosphorylated synuclein. Moreover, axonal transport analysis in cultured neurons reveals that mutations mimicking permanent phosphorylation of αsynuclein similarly slow down the axonal movement of this protein [2,36].

Due to the wide variety of defects that can occur in intraneuronal movement of vesicles, an accurate detection and characterization is essential to distinguish between the numerous neurodegenerative diseases. In a near future, vesicle movement characterization can even work as a prediagnostic tool for this kind of disorder.

Chapter 3

Computational Analysis

This chapter approaches several aspects of the computational analysis of bioimages. It stars with a description of most commonly used bioimaging techniques and then it explores different phases of image processing methods, ranging from image restoration to classification and tracking algorithms. Finally it is described a similar tracking tool available on the market which was used for comparison with NeuronDyn.

3.1 Neuroimaging Techniques

Neuroimaging is a branch of neuroscience responsible for acquiring images of the nervous system. This discipline can be divided in: structural imaging, which deals with the diagnosis of injuries like tumors, and functional imaging, used to indirectly measure brain functions [37]. The automation of the acquisition and interpretation of biological data in microscopy began in the 1950s. This first step was focused on digitalizing optical information of the cell cultures to images to obtain basic measurements like cell size and count. During the last decades, the processing and extraction of information from images has become indispensable in the experimental research and neuroscientists have benefited from the emergence of many powerful techniques that cover both spatial and temporal scales [38, 39]. In Figure 3.1 there is a comparison in terms of the spatial and temporal resolution of some neuroimaging techniques [39].

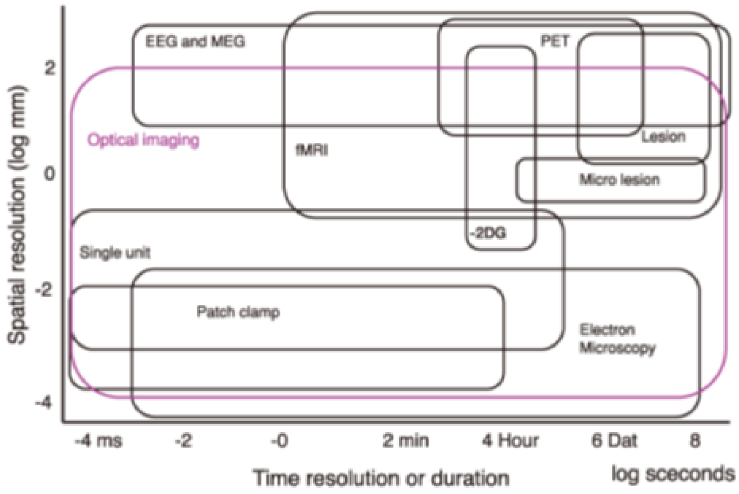
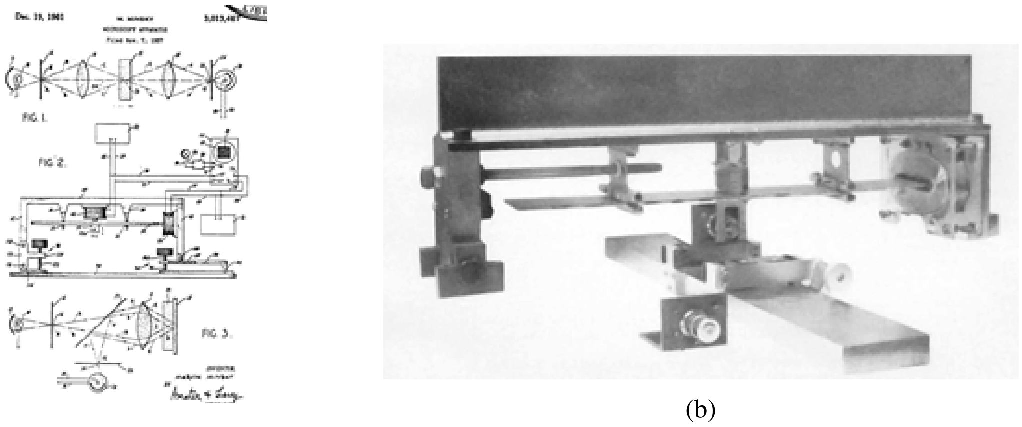


Figure 3.1: Temporal and spatial resolution of some Neuroimaging techniques with particular emphasis on optical imaging [39].

3.1.1 Spinning-disk Confocal Microscopy

The basic concept of confocal microscopy was originally developed by Marvin Minsky in the mid-1950s (patented in 1957) when he was a postdoctoral student at Harvard University. Minsky wanted to image neural networks in unstained preparations of brain tissue and was driven by the desire to image biological events as they occur in living systems (see Figure 3.2). Minsky's invention remained largely unnoticed, due most probably to the lack of intense light sources necessary for imaging and the computer horsepower required to handle large amounts of data [40–42].



(a)

(b)

Figure 3.2: a)First page of the patent registered in 1957; b)First prototype of a confocal microscope developed by Marvin Minsky [40].

Confocal microscopy offers several advantages over conventional wide field optical microscopy, including the ability to control depth of field, elimination or reduction of background information and the capability to collect serial optical sections from thick specimens. The basic key to the confocal approach is the use of spatial filtering techniques to eliminate out-of-focus light or glare in specimens whose thickness exceeds the immediate plane of focus. There has been a remarkable increase in the popularity of confocal microscopy in past years, due in part to the relative ease with which high-quality images can be obtained from specimens prepared for conventional fluorescence microscopy, and the growing number of applications in cell biology. The best advantage of these tools is that the movement of organelles, particles and even proteins can be visualized in living cells. In fact, confocal technology is proving to be one of the most important advances ever achieved in optical microscopy [40, 43].

Given the need for a rapid confocal scanning system, scientists developed what is called Confocal Laser Scanning Microscopy (CLSM). This technique was created to eliminate the out-of-focus haze of fluorescent objects. Other methods such as computational deconvolution are also used to sharpen images, but CLSM has quickly spread among life science laboratories because of its convenience and ease of use. However, there is a big serious problem if one wants to apply CLSM to vesicular transport. The original and many current models of confocal microscopy adopt a mechanical way of scanning. As shown in Figure 3.3, a typical confocal microscope focuses the laser beam after passing it through a pinhole into a very small light point. This point has to be scanned over the specimen by the mechanical movement of mirrors. This is called point-scanning or galvano-mirror method. The fluorescence emitted from the sample is now collected by a photomultiplier tube and its change in time is then reorganized into a 2-D image by a computer. Unfortunately, the single-beam laser confocal microscope is limited in image acquisition speed due to the need for extremely precise control of galvanometer mirrors that are used to raster scan the beam across the specimen, as well as the limited number of photons emitted by the specimen during the pixel settle time [43–45]. An example of this type of confocal microscope can be seen in Figure 3.3.

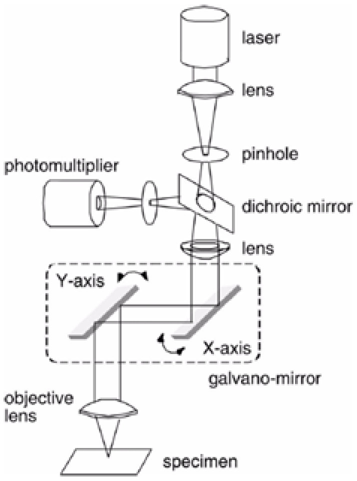
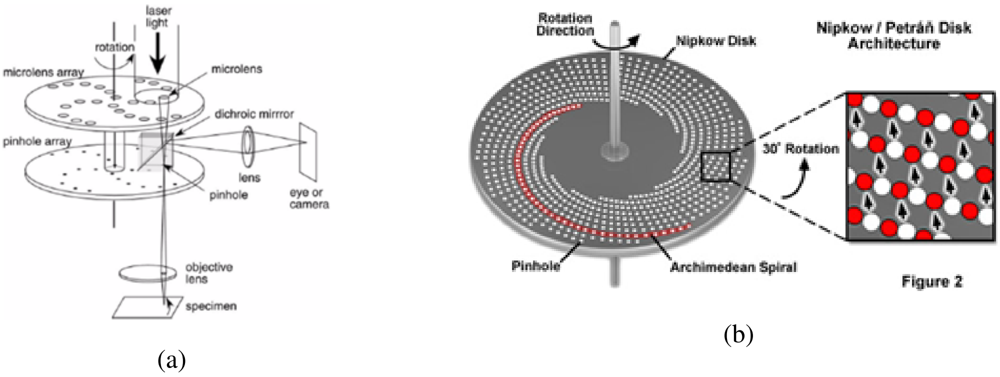


Figure 3.3: Conventional confocal laser scanning microscope [43].

A different type of scanning method, known as the Nipkow disk or spinning-disk method, raster-scans the specimen with many light points obtained through multiple pinholes. The Yokogawa Electric Corporation devised a unique Nipkow disk system with another coaxial spinning disk containing an array of microlenses, which efficiently guides the laser beams into pinholes (Figure 3.4).



(a)

(b)

Figure 3.4: a) Nipkow disk confocal laser scanning microscope with microlens. b) closer view of the Nipkow spinning disk [43].

Using this method, any of the speed limitations associated with point-scanning confocal microscopes can be overcome. Spinning disk confocal microscopes are emerging as a powerful tool for rapid spatial and temporal imaging of living cells. Although the technique was originally introduced over 40 years ago, recent improvements in microscope optical design and camera technology have significantly expanded the versatility and potential of this approach. Examples of images obtained through this imaging technique are presented on Figure 3.5.

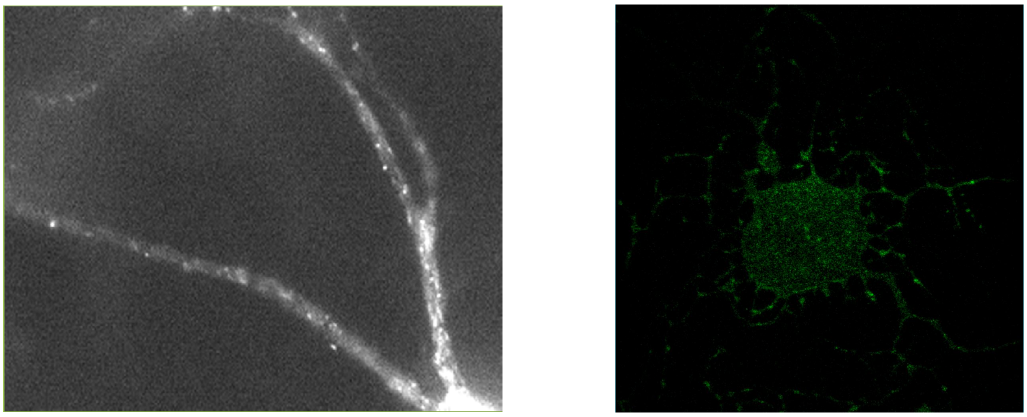


Figure 3.5: Examples of images obtained by spinning-disk confocal microscopy (withdrawn from the dataset used in our project).

3.2 Image Analysis in CAD Systems

These days, it is extremely important to extract, analyze and represent information from digital images obtained by neuroimaging techniques. After gathering the raw images, it is necessary to process them in order to enhance a determined characteristic of interest. These image processing tools are widely used in many fields like medicine, biological research and metallurgy [46]. In most of the cases the final objective of image processing is to detect and/or segment a given object (like cells, particles or organelles) or to study objects' movement over time.

3.2.1 Types of Noise

Noise is still very common in medical images, and is generally characterized by random intensity values that are distributed in many different ways. Bearing this in mind, several noise models were created, the most common ones are:

Gaussian Noise is an addictive type of noise, meaning that the corrupted pixel is the sum of the true pixel intensity value and a random Gaussian distributed noise value. It is difficult to remove because it generally affects almost every pixel in the image. Gaussian distribution is given by:

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{\frac{-(g-m)^2}{2\sigma^2}}$$

Where g represents the gray level, m is the mean or average of the function, and σ is the standard deviation of the noise.

Salt and pepper noise is an impulse type of noise, which means that the corrupted pixels are randomly set to either a maximum or minimum intensity value (generally 0 to "pepper" noise and 255 to "salt" noise). This kind of noise is normally related with defective pixel in camera sensors or errors in data transmission.

Speckle noise is a multiplicative noise that is common in many imaging systems. The source of this noise is attributed to random interference between the coherent returns. Speckle distribution is given by:

$$F(g) = \frac{g^{a-1}}{(a-1)!a^a} \times e^{\frac{-g}{a}}$$

Where a^α is the variance and g is the gray level.

Brownian noise is an example of fractal (1/f) noise which follows the mathematical model of fractional Brownian motion. Fractal Brownian motion is a non-stationary stochastic process that follows a normal distribution. Brownian noise is a special case of 1/f noise. It is obtained by integrating white noise [47].

3.2.2 Image Restoration

The quality of the images acquired through neuroimaging techniques is still relatively low and it is extremely difficult to extract proficuous information without any further processing. Artefacts such as noise, blur, optical aberrations, between others, must be attenuated in order to obtain better data from the images. Depending on the method used to acquire the images, we find different types of artefacts and consequently several different denoising processes have been developed [48].

After the type of noise is correctly identified, we can use the denoising method that best suits our needs. The main problem about this choice resides on the fact that most methods are specific for one problem and its performance may not be the ideal for our situation. Generally, denoising methods can be divided into two major groups: spatial domain filters and transform domain filters (illustrated in Figure 3.6).

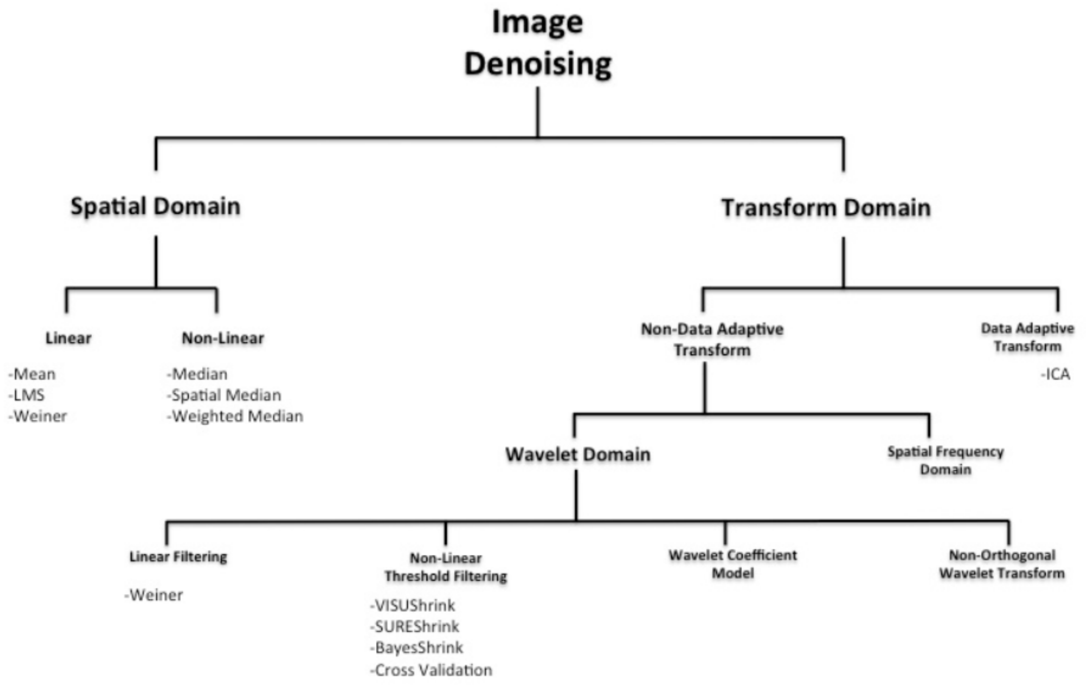


Figure 3.6: Classification of image denoising methods [4].

3.2.2.1 Spatial Domain Filters

These are the most traditional methods used in image restoration and they can be divided in linear and non-linear filters. In the first case, the output values are linear function of the pixels in the original image and are easier to analyze mathematically. The non-linear filters present more accurate results due to their ability to preserve edges.

Linear Filters. These filters are good in the presence of noise with known distribution model (e.g. Gaussian), they are able to remove noise to a reasonable extent and are easy and fast to implement but have the inconvenient of blurring edges. There are several filters in this category, the mean filter, for example, calculates the average value of a predefined area of the image and sets the region's central pixel intensity value to that average value. It smoothes the image by reducing the intensity variation between adjacent pixels being a good solution to Gaussian noise. The Wiener filter (also known as minimum mean-squared error – MMSE) convolves the image with a constant matrix to obtain a linear combination of neighborhood values. It removes the additive noise and inverts the blurring simultaneously, without compromising the visibility of edges [47–49].

Non-Linear Filters. With non-linear filters, the noise is removed without any attempts to explicitly identify it. Spatial filters employ a low pass filtering on groups of pixels with the assumption that the noise occupies the higher region of frequency spectrum. In recent years, a variety of nonlinear median type filters such as weighted median have been developed to remove noise but preserving edges, unlike most linear filters. Weighted median filtering combine the robustness and edge preserving capability of the classical median filter and great properties in sparsity representation [47, 48, 50].

3.2.2.2 Transform Domain Filters

Although the transform domain filtering methods can be subdivided in data adaptive and non-adaptive, the last are the most popular and most commonly used. Between non-adaptive filters the most used are:

Spatial Frequency Filters. Spatial-frequency filtering refers to the use of low pass filters using Fast Fourier Transform (FFT). In frequency smoothing methods the removal of the noise is achieved by designing a frequency domain filter and adapting a cut-off frequency when the noise components are decorrelated from the useful signal in the frequency domain. However, these methods are time consuming and depend on the cut-off frequency and the filter function behavior [48].

Wavelet Domain Filters. Many image denoising approaches perform denoising on wavelet domain. Wavelet decompositions have the desirable property of locality both in space and in frequency, which is not the case for other transforms, such as the Fourier transform. Wavelet-based denoising algorithms are based on the following steps. First, an image is transformed into a wavelet domain. Next, denoising is effected on the wavelet coefficients, and finally the denoised image is obtained by applying the inverse wavelet transform on the denoised wavelet coefficients. Linear filters such as Wiener filter yield optimal results in the wavelet domain when the signal corruption can be modeled as a Gaussian process and the accuracy criterion is the mean square error.

However, the most used domain in denoising using Wavelet Transform is the non-linear coefficient thresholding based methods. This methods exploit sparsity property of the wavelet transform and the fact that the Wavelet Transform maps white noise in the signal domain to white noise in the transform domain. Moreover, while signal energy becomes more concentrated into fewer coefficients in the transform domain, noise energy does not. It is this important principle that enables the separation of signal from noise [47, 48].

3.2.3 Image Enhancement

Following the image restoration, the aim is to perform the enhancement of the region of interest. In simple terms, image enhancement consists on improving the interpretability or perception of information in images for human viewers and providing an easier input for other automated image processing techniques. The principal goal of image enhancement is to modify attributes of an image to make it more suitable for a given task and a specific observer [51]. During this process, one or more attributes of the image are modified. The choice of attributes and the way they are modified are specific to a given task and image. Basically there are two main categories in which image enhancement can be divided (Figure 3.7).

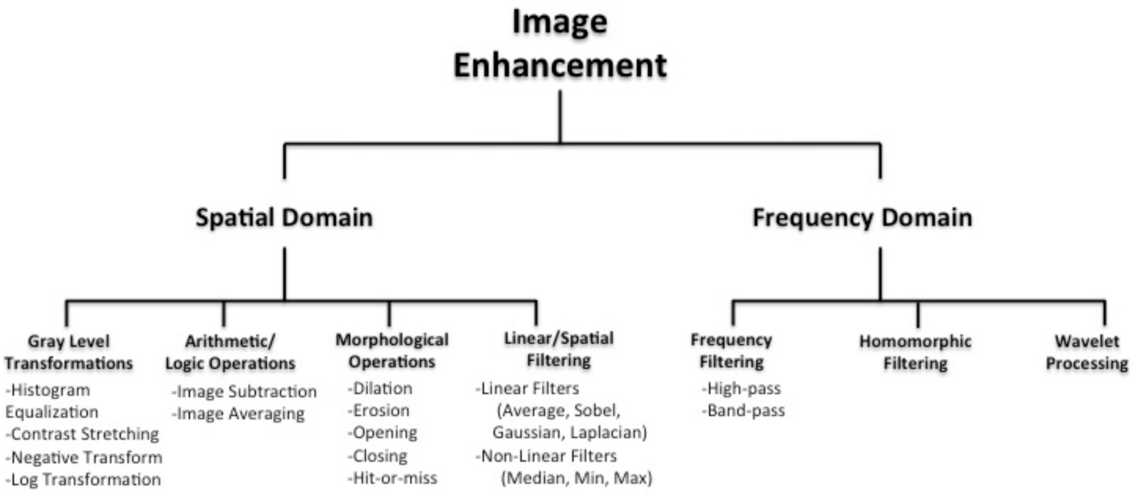


Figure 3.7: Image Enhancement Methods [4].

Spatial Domain Methods. Spatial domain techniques directly deal with the image pixels. The pixel values are manipulated to achieve desired enhancement. Between the most commonly used techniques are logarithmic transforms, power law transforms and histogram equalization. Spatial methods are particularly useful for directly altering the gray level values of individual pixels and hence the overall contrast of the entire image. However, they usually enhance the whole image in a uniform manner which in many cases produces undesirable results [51, 52].

Frequency Domain Methods. In frequency domain methods, the image is first transferred into frequency domain. It means that, the Fourier Transform of the image is computed first. All the enhancement operations are performed on the Fourier transform of the image and then the Inverse Fourier transform is performed to get the resultant image. These enhancement operations are performed in order to modify the image brightness, contrast or the distribution of the grey levels. As a consequence the pixel value (intensities) of the output image will be modified according to the transformation function applied on the input values [51–53].

3.2.4 Image Segmentation

The objective of this image segmentation methods is to separate the object we need to study from the rest of the image. Based on the response of the filters in the enhancement step, and according to exclusion criteria it's possible to map the object candidates. There are several techniques used for this purpose, depending on the type and quality of images, and on the final goal. These techniques can be divided into categories sorted by the method used to segment the image (Figure 3.8) [54].

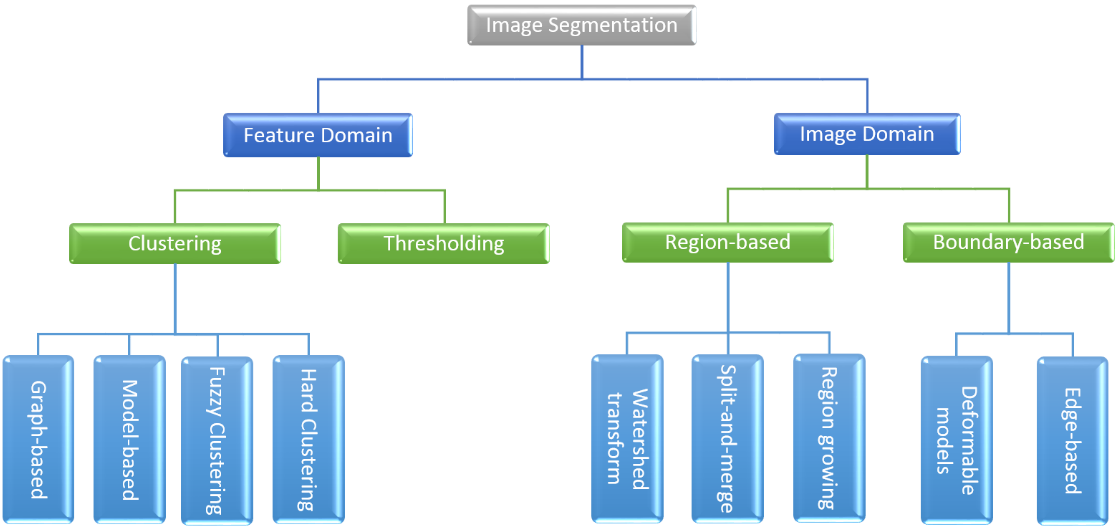


Figure 3.8: Principal Image Segmentation methods. Adapted from [54].

Thresholding. This method has a relatively simple principle, which is based on a clip-level (or a threshold value) to turn a gray-scale image into a binary image. This way, the most important factor while using this method resides on choosing an adequate threshold value that fits our needs and our image. Several popular methods include the maximum entropy method, Otsu's method (maximum variance), and k-means clustering [55].

Clustering. The most common clustering technique is the K-means algorithm. This method is an iterative technique that is used to partition an image into K clusters. Basically this algorithm calculates the center from each of the K clusters and measures the distance from each pixel in the image to every picked center. After comparing the obtained distances it decides that the pixel belongs to the cluster where the distance center-pixel is minimal. Then the pixel is incorporated in the cluster and a new center is calculated already counting with the recently classified pixel. All previous process is repeated until all pixels are classified.

This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K [56].

Region-based. Also known as region-growing, most part of these methods work in the following way: a set of seeds are defined as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighboring pixels to the regions. The difference between a pixel's intensity value and the region's mean is used as a measure of similarity. The pixel with the smallest difference measured is allocated to the respective region. This process continues until all pixels are allocated to a region. Seeded region growing requires seeds as additional input. The segmentation results are dependent on the choice of seeds. Noise in the image can cause the seeds to be poorly placed [57, 58].

Boundary-based. The basic idea in active contour models or snakes is to evolve a curve, subject to constraints from a given image, in order to detect objects in that image. For instance, starting with a curve around the object to be detected, the curve moves toward its interior normal and has to stop on the boundary of the object.

The snakes' model is popular in computer vision, and led to several developments in 2D and 3D. In two dimensions, the active shape model represents a discrete version of this approach, taking advantage of the point distribution model to restrict the shape range to an explicit domain learned from a training set.

This method is particularly efficient for noisy images, for example, when the interior of the object is too chaotic to apply a region growing method, as long as you can enhance and make visible the edges of the object or at least a part of them [59].

3.2.5 Feature Extraction

The main goal of image processing tools is to be able to detect and extract significant features from images. This way we can divide feature extraction method in two categories:

Low-level methods. We can define low-level features to be those basic features that can be extracted automatically from an image without any shape information. The first low-level feature is called edge detection and it aims to produce a line drawing. Edge detectors can also be divided into two groups: first-order detectors, which are equivalent to first-order differentiation, and second-order edge-detection operators that are equivalent to a higher level of differentiation. Between the most used first-order edge detectors are Prewitt, Sobel and Canny detectors while in second-order detectors we find the Laplacian and Marr-Hildreth operators [60]. A visual comparison between edge detection operators is shown in Figure 3.9. Other examples of low-level methods are corner, blob and ridge detectors.

High-level methods. By opposition, high-level feature extraction, concerns finding shapes and objects in computer images. To be able to recognize human faces automatically, for example, one approach is to extract the component features, like the eyes, the ears, and the nose, which are the major face features. To find them, we can use their shape: the eyes are approximately ellipsoidal; the mouth can appear as two lines, as do the eyebrows. Alternatively, we can view them as objects and use the low-level features to define collections of points which define the eyes, nose, and mouth, or even the whole face. This feature extraction process can be viewed as similar to the way we perceive the world. High-level detectors are also called shape detectors due to their concern about finding determined shapes in the images. Figure 3.10 provides an overview over shape descriptors.

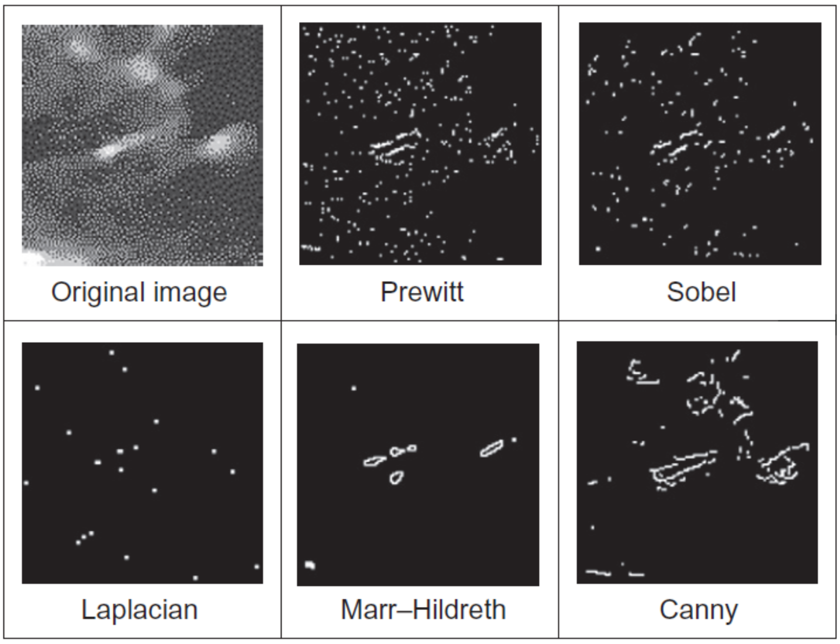


Figure 3.9: Comparison of edge-detection operators. Adapted from [60].

Any feature extraction method aims, obviously, to obtain certain features from a certain object. There are several types of features that can be extracted from images, namely:

Color Features. Color is one of the most important features of images. Color features are defined subject to a particular color space or model. A number of color spaces have been used in literature, such as RGB, LUV, HSV and HMMD. Once the color space is specified, color feature can be extracted from images or regions. A number of important color features have been proposed in the literatures, including color histogram, color moments, color coherence vector and color correlogram [62].

Textures Features. Texture is a very useful characterization for a wide range of image. It is generally believed that human visual systems use texture for recognition and interpretation. In general, color is usually a pixel property while texture can only be measured from a group of pixels. A large number of techniques have been proposed to extract texture features. Based on the domain from which the texture feature is extracted, they can be broadly classified into spatial texture feature extraction methods and spectral texture feature extraction methods. For the former approach, texture features are extracted by computing the pixel statistics or finding the local pixel structures in original image domain, whereas the latter transforms an image into frequency domain and then calculates feature from the transformed image. Both spatial and spectral features have advantage and disadvantages.

Shape Features. Shape is known as an important cue for human beings to identify and recognize the real-world objects, whose purpose is to encode simple geometrical forms such as straight lines in different directions. Shape feature extraction techniques can be broadly classified into two groups: contour based and region based methods. The first one calculates shape features only from

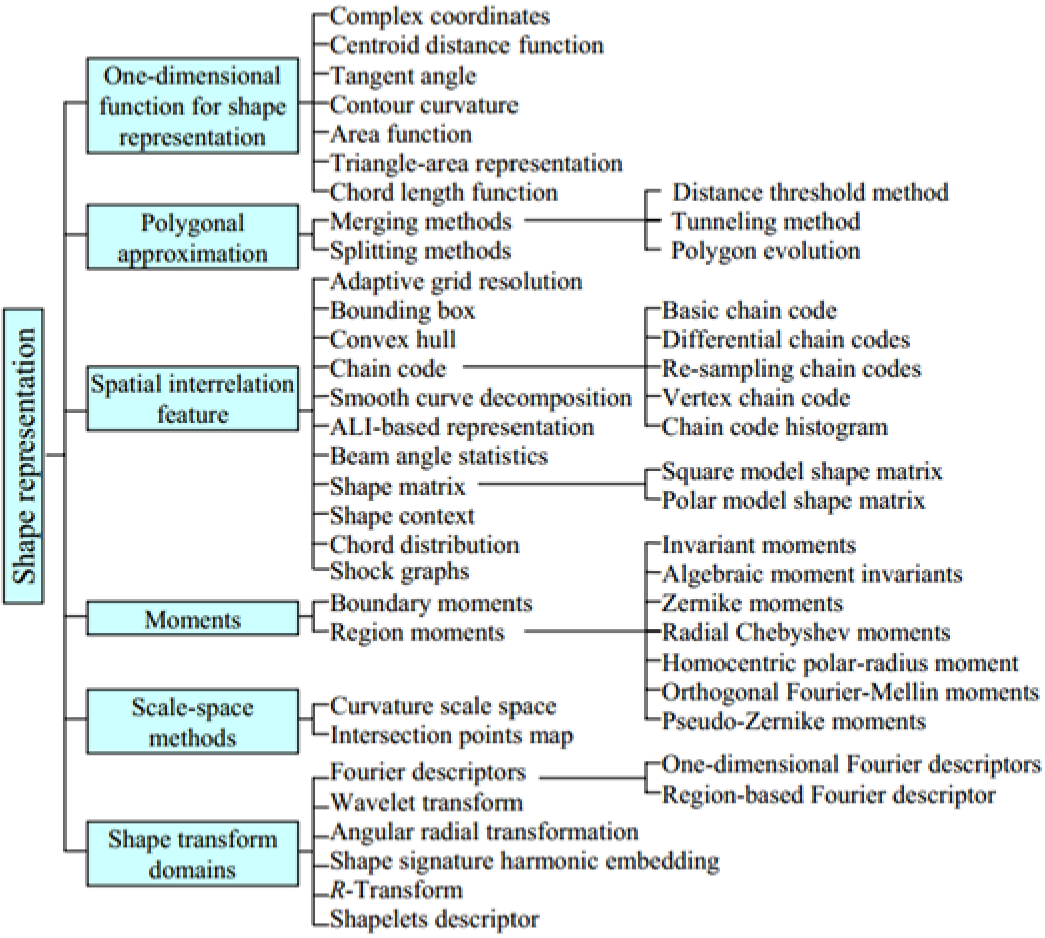


Figure 3.10: An overview of shape description techniques [61].

the boundary of the shape, while the latter method extracts features from the entire region [62]. Figure 3.11 presents a general overview of shape descriptors.

3.2.6 Machine Learning and Classification

Machine learning can be considered a subfield of computer science and statistics, which is employed in a range of computing tasks where designing and programming a rule-based algorithm is impractical. These methods have been widely used in the past decades in computer vision and image processing [64].

Generally, we can divide machine learning algorithms in two types: supervised and unsupervised methods. Most commonly used machine learning methods are presented in Table 3.1.

3.2.6.1 Supervised Learning

Supervised learning methods are based on inferring a function from a set of training examples. A supervised learning algorithm analyzes the training data and produces an inferred function, which

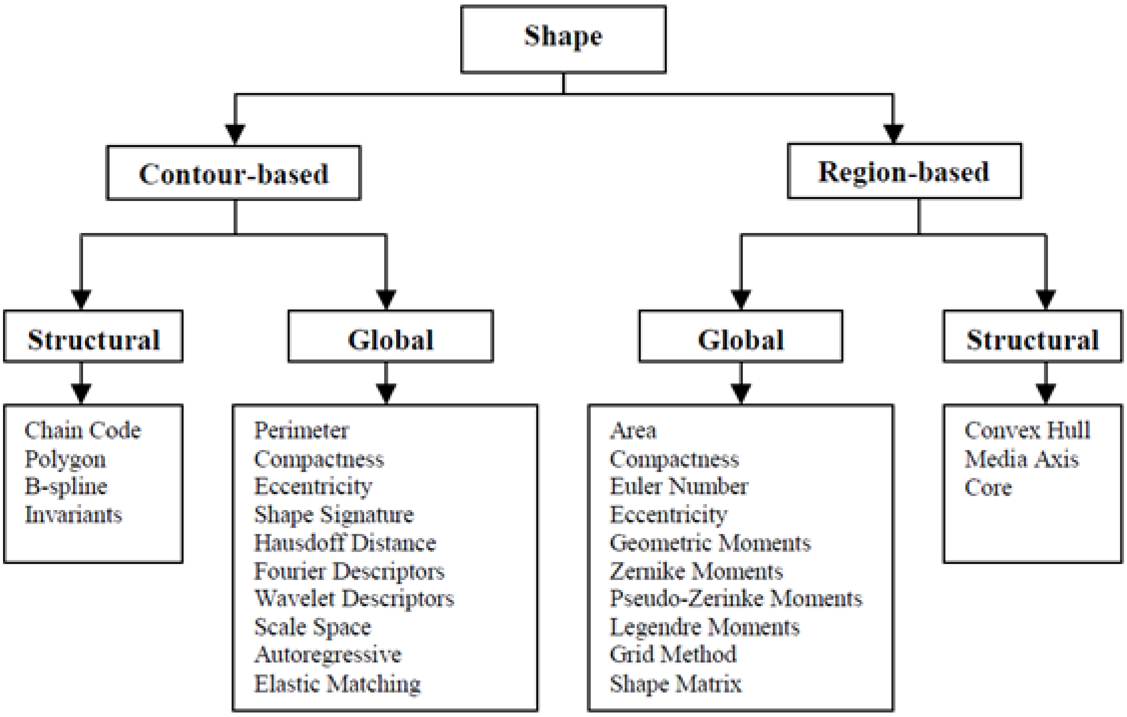


Figure 3.11: Overview of shape descriptors [63].

can be used for mapping new examples [64, 65]. Among supervised learning methods, the most widely used are:

Artificial Neural Networks (ANNs). ANNs are computational models inspired by biological neural networks and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown [66]. These artificial neurons work together in a distributed manner to learn from the input information, to coordinate internal processing, and to optimize its final output. The basic structure of a neuron can be theoretically modelled as shown in Figure 3.12, where X xi, i=1, 2, ..., n represent the inputs to the neuron and Y represents the output. Each input is multiplied by its weight wi, a bias b is associated with each neuron and their sum goes through a transfer function f.

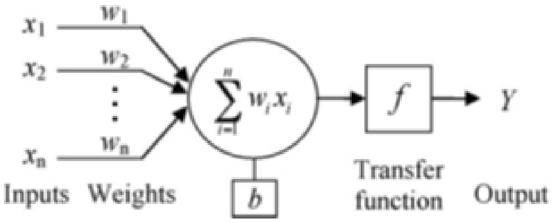


Figure 3.12: Basic functioning of artificial neuron networks [67].

Table 3.1: Machine Learning Methods

	Unsupervised	Supervised
Continuous	Clustering & Dimensionality Reduction: •Mean-shift •Fuzzy C-means •K-means	Regression: •Linear •Polynomial Decision Trees Random Forests
Categorical	Association Analysis • Apriori • FP - Growth Hidden Markov Models	Classification • KNN • Trees • ANN • Naive-Bayes • SVM

As a result, the relationship between input and output can be described by the following equation,

$$Y = f(\sum_{i=1}^{n} w_i x_i + b)$$

In recent years, ANN has been widely used for medical image segmentation and classification purposes and a large variety of ANN based algorithms have been developed to segment images with high accuracy rates.

Naive-Bayes classifiers. Bayesian classifiers have been used in many areas and fields due to their capability of “learning”, fundamental characteristic in many neuroscience studies. A certain collection of data (training stage) can be passed to the classifier in order to provide discriminatory information to differentiate the objects from the rest of the image. The more accurate and precise the information is, the better the classifying result. The Bayesian classifier is based on Bayes’ formula, expressed in the following equations:

$$P(W_i \mid x) = \frac{P(W_i)p(x \mid W_i)}{p(x)}$$

$$p(x) = \sum_{k=1}^{c} P(W_i)p(x \mid W_i)$$

where P(Wi|x) represents the probability of Wi occurring given x. P(Wi) is the probability alone of W occurring, whereas P (x|Wi) is the likelihood and p(x) is the evidence. c is the number of classes.

Summarizing, the Bayesian Classifier determines the probability of the evidence x belong to each of the c classes depending of his proper occurrence. Then, the decision is simple; it’s considered object if it belongs to a c class that presents the high probability, knowing that the exit is x, or it is not considered object otherwise [68]. The Bayesian classifier for normally distributed classes with equal covariance matrices is a linear classifier. Both Euclidean and minimum Mahalanobis distances are used in these situations to obtain the decision line between two classes. If the classes are normally distributed but without equal covariance matrices a quadratic classifier is applied [69].

Support Vector Machines (SVM). In machine learning, SVM are learning algorithms that analyze data and recognize patterns and are used for classification and regression analysis. Given a set of labelled training examples, an SVM training algorithm constructs a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [70, 71]. Two general attributes define the SVM algorithm: C, a hyper-parameter which controls the trade-off between margin maximization and error minimization; and kernel, a function that maps training data into high-dimensional features spaces. The kernel function is used to train SVMs classifiers. The type of kernel function used is a key factor on the performance of SVM classification algorithm. The types which are more commonly used are the linear (Linear SVM) and the gaussian (Radial Basis Function, RBF) - RBF SVM [72]. In this phase of the project we are using a linear kernel SVM classifier.

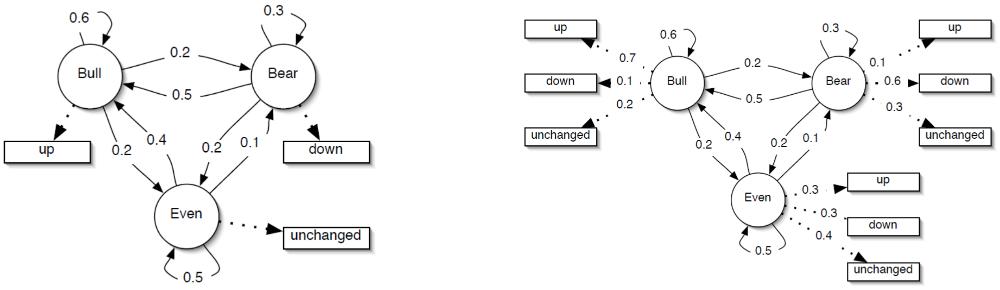
3.2.6.2 Unsupervised Learning

By opposition, unsupervised learning consists on trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. The most common unsupervised learning methods are clustering methods which consist on representing characteristics in the feature space to find natural grouping clusters [65–67]. The most common unsupervised learning algorithms are:

K-means. This algorithm starts with a set of information, and a choice of the number of clusters (k, number of different regions to segment). These clusters have a centroid (intensity mean e.g.) that can be selected randomly or not. Then, every pixel will be labelled as belonging to the cluster that has the minor distance to the mean. After all pixels are classified, the algorithm will estimate the new position of the centroids. This will lead to a new comparison of all pixels with these new means and new labelling, changing the clusters. This process occurs until the means of each cluster stays unchanged between two consecutive iterations. After this, the pixels of the image assume the value of the mean of its cluster, producing a histogram with only k intensities. Although it's a precise segmentation method it has high computational cost, and produces poor results if the images have low quality [56, 73].

Hidden Markov Models (HMMs). The model presented in Figure 3.13a describes a simple model for a stock market index. The model has three states, Bull, Bear and Even, and three index observations up, down, unchanged. The model is a finite state automaton, with probabilistic transitions between states. Given a sequence of observations, example: up-down-down we can easily verify that the state sequence that produced those observations was: Bull-Bear-Bear, and the probability of the sequence is simply the product of the transitions, in this case 0.2 0.3 0.3.

Figure 3.13b shows an example of how the previous model can be extended into a HMM. The new model now allows all observation symbols to be emitted from each state with a finite probability. This change makes the model much more expressive and able to better represent our intuition, in this case, that a bull market would have both good days and bad days, but there would be more good ones. The key difference is that now if we have the observation sequence up-down-down then we cannot say exactly what state sequence produced these observations and thus the state sequence is 'hidden'. We can however calculate the probability that the model produced the sequence, as well as which state sequence was most likely to have produced the observations.



(a)

(b)

Figure 3.13: a) Markov process representation; b) Hidden Markov model representation [74].

3.3 Tracking/Motion Analysis in CAD Systems

Full understanding of any animate entity requires studying not only its spatial (anatomic) but also its temporal (dynamic) properties [46]. It is therefore no surprise that research in medicine and biology has come to rely increasingly on timelapse imaging and longitudinal examinations. In both the health sciences and the life sciences, the technologically deficient times when researchers had to draw conclusions based on static two-dimensional (2D) images are long gone, and it is now commonplace to image and study subjects in three dimensions over time (denoted 3D+t or 4D). Actually, the number of publications which included the words particle tracking, cell tracking or similar as largely increased in the last years, as it is visible in Figure 3.14.

The general assumption made by most cell or particle tracking tools is that the cells are generally represented by bright regions against a darker background (the fluorescence microscopy scenario). If this is not the case, or if the images are too noisy, it is necessary to apply suitable filters to match this assumption. Most commercial tracking tools, as well as more general purpose open-source software packages, offer ample functionality for image preprocessing [75]. Some of the most relevant tracking algorithms on the market are summarized on Table 3.2.

Bearing in mind the main goal of this project it will be described in more detail FluoTracker system. It will also be described NeuronDynamics tool, which was the precursor of NeuronDyn.

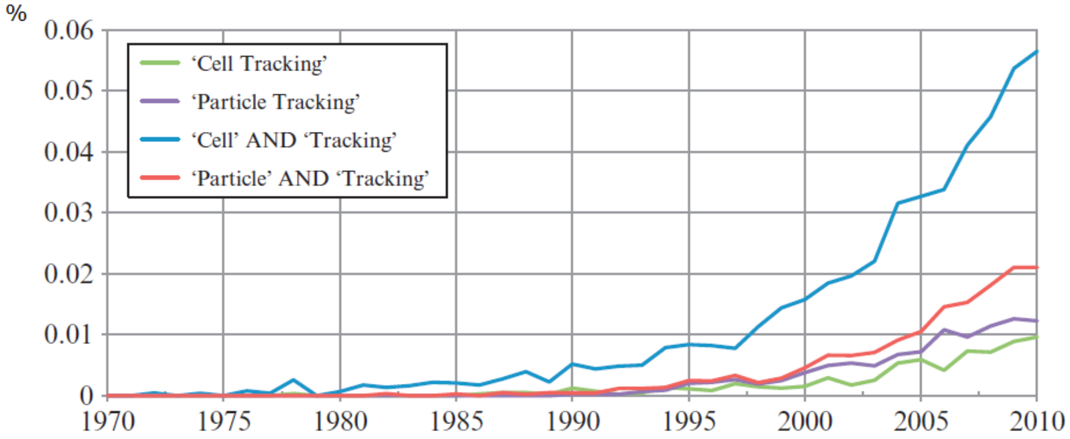


Figure 3.14: Percentage of publications in the PubMed database as a function of publication year for the indicated combinations of words in the title and/or abstract [75].

A final comparison will be established between these three algorithms in order to better understand how our algorithm can be improved to obtain even better results.

The main goal of object tracking is to locate a moving object in consecutive video frames. Normally, a video tracking system combines two key steps: object detection and object tracking. The first step, object detection, can be defined as the process of segmenting an object of interest from a sequence of video scenes. This process should keep track of its motion, orientation, occlusion, between others, in order to extract useful context information, which will be used on higher-level processes. In the second step, object tracking, given the locations of the particles in successive images, we need to link them along time. Object tracking can be a very difficult process due to several problems, namely: the object's shape and size may vary from frame to frame, partial and full object occlusion, presence of noise and blur in video, luminance and intensity changes and object's abrupt motion [76–78].

Designing an universal tracking algorithm is nearly impossible due to the enormous variety of images and objects to be tracked. Figure 3.15 illustrates object tracking process.

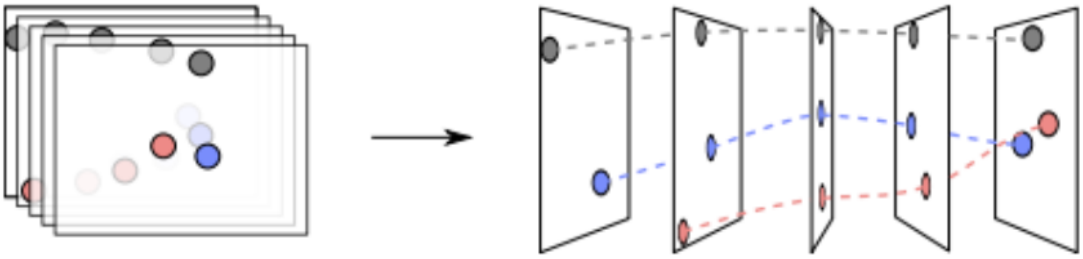


Figure 3.15: Schematic representation of multi-object tracking [76].

Table 3.2: List of some available particle tracking tools. Adapted from [75].

Name	Available	Platform	Dimension	Automation
CellProfiler	Free	Win/Lin/Mac	2D	Auto
ClusterTracker	Free	Matlab	2D	Auto
Fluotracker	Paid	Matlab	3D	Auto
Icy	Free	Java	3D	Auto
Image-Pro Plus	Paid	Win	3D	Auto
ImarisTrack	Paid	Win/Mac	3D	Auto
ManualTracking	Free	ImageJ	3D	Manual
MetaMorph	Paid	Win	3D	Auto
MTrack2	Free	ImageJ	2D	Auto
MTrackJ	Free	ImageJ	3D	Manual
MTT	Free	Matlab	2D	Auto
Octane	Free	ImageJ	2D	Auto
ParticleTracker	Free	ImageJ	3D	Auto
plusTipTracker	Free	Matlab	2D	Auto
PolyParticleTracker	Free	Matlab	2D	Auto
SpeckleTrackerJ	Free	ImageJ	2D	Semi
SpotTracker	Free	ImageJ	2D	Auto
TIKAL	Request	Win/Lin	3D	Auto
u-track	Free	Matlab	2D	Auto
Volocity	Paid	Win/Mac	3D	Auto

In an overall way, object tracking methods can be divided in three groups: point tracking, appearance (or kernel) tracking, and silhouette tracking.

Point tracking. The detected objects are represented by points, and the tracking of these points is based on the previous object states which can include object positions and motion. Point tracking is a difficult problem particularly in the existence of occlusions, false detections of object.

An example of object correspondence is shown in Figure 3.16a [77, 79].

Appearance tracking. The object appearance can be for example a rectangular template or an elliptical shape with an associated RGB color histogram. Objects are tracked by considering the coherence of their appearances in consecutive frames (Figure 3.16b) [77].

Silhouette tracking. The tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution (Figures 3.16c, 3.16d) [77].

Inside these three categories there are several tracking methods that are based on the same general objective but differ in the process to achieve it. The main tracking algorithms are specified in Figure 3.17. Some of the most used tracking algorithms are described then.

Kalman filter. The Kalman filter is an algorithm that uses a series of measurements observed

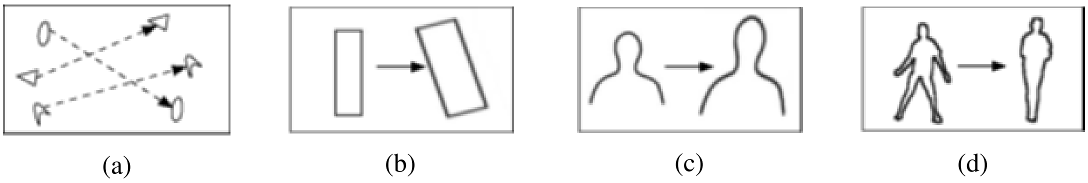


Figure 3.16: Illustration of different tracking approaches. (a) Multipoint correspondence, (b) Parametric transformation of a rectangular patch, (c, d). Two examples of silhouette matching [79].

over time, containing noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. More formally, the Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The filter is named for Rudolf E. Kálmán, one of the primary developers of this method [80].

The algorithm works in a two-step process. In the first step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. After gathering the results of the next measurement, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. Because of the algorithm's recursive nature, it can run in real time using only the present input measurements and the previously calculated state and its uncertainty matrix. One of this method's advantages is that Kalman filter does not require any assumption of error distribution [81].

NNA Method. A simple and commonly used approach to link objects is the nearest neighbour association. After an object is segmented in a certain frame, the distance between this object's position and similar objects' position in the following frame is measured. The par of objects with shortest distance between them is linked as the same object. [82]. Other extensions of this method are: the probabilistic nearest neighbor, distributed sequential nearest neighbor, suboptimal nearest neighbor, and global nearest neighbor (GNN).

The GNN tracking algorithm not only gives the nearest object from one frame to other, but it also relates the information of a defined of number of frames. This method can deal with gaps, which happen when one particle that is detected in one frame is not detected in the subsequent one, appearing in a further frame. This algorithm can efficiently identify and associate targets in complex state, such as targets with parallel movement, targets with intersecting movement, and targets with turning movement, but it only associates at most one target point, ignoring divisions [83, 84].

3.3.1 Vesicle Tracking

There are several problems regarding the tracking of neurotransmitter vesicles in images obtained from neuronal cultures, using optical imaging, namely: high noise level and image artefacts, non–uniform background, variant brightness of vesicles introduced by inherent heterogeneity of the staining, variability of vesicle shapes and sizes, and partial occlusion and clustering. These characteristics lead to difficulties in the detection of the vesicles. Inaccurate detection normally

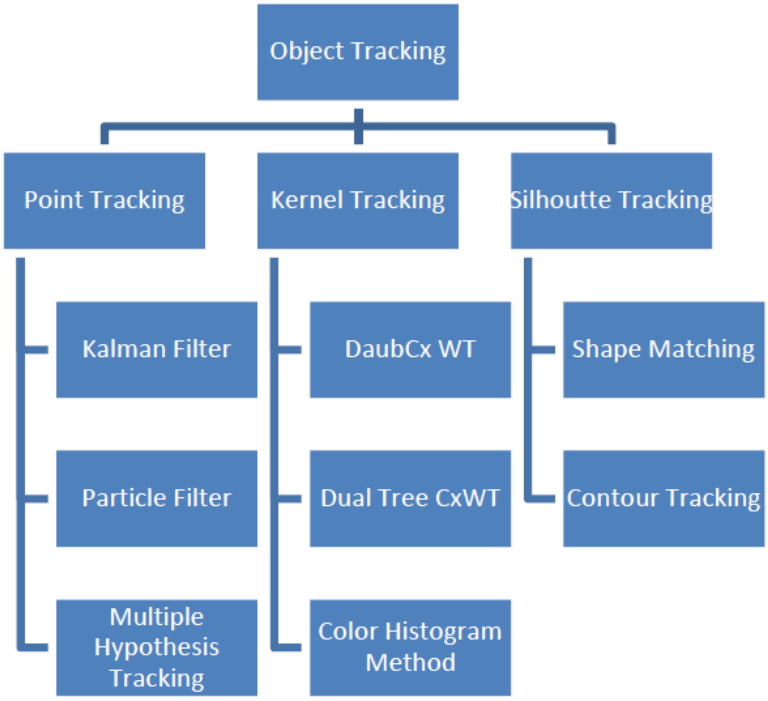


Figure 3.17: Object Tracking Methodologies [77].

leads to false positives (background noise classified as vesicle) or false negatives (missed detection of real vesicles), compromising the quantification [82, 85].

An essential requirement for tracking a vesicle is to determine its position in each frame of the image sequence. When the number of vesicles is fixed, there is high frame rate and the detection step was well performed, this task is reduced to link objects between the frames. In that case NNA offers a good and fast solution. As said before, NNA can be used to track vesicles, although it is not the best approach taking in consideration vesicle events like the merge/split, birth/death, clusters, lower quality of images with high background noise, changes of intensity of vesicles, and changes in object's area [82, 86]. Due to this nonlinear movement and events, non-linear approaches are the best solution to track these biological targets.

3.3.1.1 FluoTracker

FluoTracker is a recursive Bayesian estimation algorithm that exploits intrinsic information contained in an image sequence. The algorithm is sequential and uses information extracted from previous frames to predict the most likely object position. The objects are detected and tracked robustly despite complicating factors inherent to biological samples. In order to track a variable number of objects with different movement characteristics, it uses multi-hypothesis tracking to render the approach computationally feasible.

It addresses problems associated with biological samples (e.g. autofluorescence, expression levels), events (split/merge, birth/death and clutters) and imaging (e.g. excitation intensity, bleaching) using three elements in their algorithm design (Figure 3.18):

1. Automatically separate the background from objects of interest by taking the maximum intensity projected over the entire sequence using a range method. Additionally, for adjoining objects, automated local analysis was used to determine a refined threshold to classify the separate objects.

2. They introduced an appearance model, describing the pixel intensities of an object as a discrete histogram, to identify unique objects in each frame. Other information such as speed or shape was not reliable due to the large variation between frames.

3. After detection, it uses a method to link the objects throughout the image sequence while conserving their identity. A complicating factor typical for biological samples is the changing number of objects in the image, caused by splitting/merging and birth/death of objects.

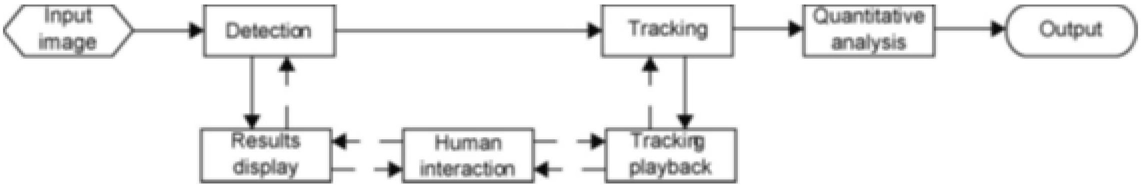


Figure 3.18: FluoTracker general workflow [82].

The program was coded in Matlab. Time-lapse stacks are converted into uncompressed AVI files using MetaMorph and loaded into the FluoTracker program using a graphical user interface. Automatic tracking of vesicles is done according to the flowchart shown in Figure 3.18. After the movie is converted to a Matlab format, the objects are detected and then tracked. After tracking, the objective magnification and time interval are entered and the location of the soma is indicated by a mouse click before quantitative analysis, and the results are exported to Excel. The program output consists of data on the number of frames each object was tracked, their direction and their average, minimal and maximal velocity over their respective track. For validation, is used the velocity and direction data from vesicles tracked from the first frame onwards for comparison with the manual tracking [82].

3.3.2 Manual Tracking vs Automatic Tracking

In neuroscience, the quantification of the dynamic properties of cell constituents is mostly done manually, an intensive and tedious labour operation due to the vast amount of data, often exceeding thousands of cargo trajectories. This manual process is userbiased and requires experience to correctly identify particles like vesicles. Despite the considerable range of tracking algorithms, the most reliable results are still achieved through manual tracking. In contrast to human inspection, none of the automated tracking approaches to date controls the reliability of its output. Yet, cell and particle tracking has a particularly fatal error propagation mechanism. Consequently, even robust tracking algorithms with very low error rates can produce substantial amounts of false results, if their reliability is not assessed independently. These facts allow semi-automatic tracking tools to increase their potential, joining the benefits of the two types. They can build tools with a training step where the user detects some particles and then the algorithm learns and does the rest. These semi-automatic tracking tools present very good results and less computational errors.

3.3.3 Tracking Measures

The direct result of applying tracking tools is a sequence of coordinates indicating the position of each tracked object at each time point. From these coordinates it can be extracted some meaningful quantitative measures which are related to motility, diffusivity, velocity and morphology of the moving objects. In this analysis, the first step is to obtain the trajectories of the tracked objects from the measured coordinates. After this some measures related to motility can be obtained like the total distance travelled, the distance between start and end point, the maximum distance from the start or any other reference point, the orientation referring to a specific point and direction and path of movement. Other measures that can be easily obtained are instantaneous and mean velocity and morphological measures like area, centroid, major axis length and eccentricity [87].

3.4 Evaluation

Every CAD system needs an evaluation process to measure its performance and compare to a well-established ground truth, allowing to compete adequately in the market and to be used in clinical applications. The ground truth is normally established by at least 2 or 3 specialist in the field. Regarding the classification of the algorithm, four types of labels can be applied to the candidates (Figure 3.19):

- True Positive (TP): When a candidate is classified as positive and it really is;
- True Negative (TN): When a candidate is classified as negative and it really is;
- False Positive (FP): When a candidate, despite being negative, is classified as positive;
- False Negative (FN): When a candidate, despite being positive, is classified as negative [88].

Using these four quantities, there are many different parameters to evaluate the algorithm response: Accuracy, sensitivity, precision/recall, specificity, F-measure, ROC curve (receiver operating characteristic), among others.

Accuracy. The accuracy is a global performance measurement of the algorithm that shows the percentage of correct classifications.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

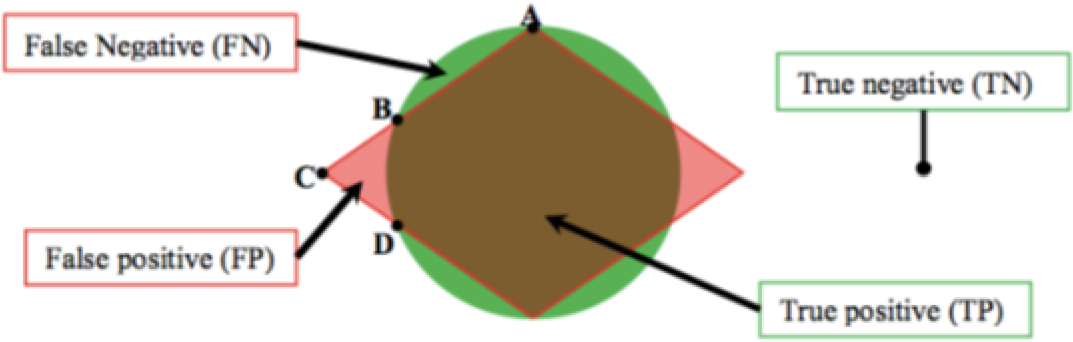


Figure 3.19: The round green object is miss-segmented as a diamond (red) [88].

Sensitivity. The sensitivity or recall indicates the true positive fraction, i.e., from all positives, which percentage the algorithm considers.

$$Sensitivity = Recall = \frac{TP}{TP+FN}$$

Specificity. The specificity expresses the percentage of times where the algorithm reports that there are no lesions, when in truth there aren't. 1-Specificity is the false positive fraction that, in other words, indicates how many lesions are indicated when they shouldn't.

$$Specificity = \frac{TN}{TN+FP}$$

Precision. The precision indicates which fraction of the detections is relevant, i.e., how many of the positive detected are really lesions.

$$Precision = \frac{TP}{TP+FP}$$

F-measure. The F-measure combines the Precision (P) and the Sensitivity/Recall (R) through a harmonic mean, showing the system's behaviour when it gives more importance to precision or to sensitivity, by comparing the resulting F-values to different values (that normally varies from 0.5 to 2). When = 0.5, precision counts double of the recall, and the inverse situation for = 2. With a F-value higher than = 0.5, the system is more capable of, from the positive choices, to have a higher true positive rate, but it's less capable of detecting all positives.

$$F_\beta = (1+\beta^2) \times \frac{P.R}{\beta^2 \times P + R}$$

ROC Curve. The type of measurements discussed above allows the construction of a performance curve (Receiver Operating Characteristic - ROC curve). This curve corresponds to the sensitivity vs 1-specificity (Figure 3.20). Every time the true positive rate rises, the same occurs to the false positive rate. The ideal operation point is the left superior corner, where only positives are detected. A bigger area under the curve corresponds to better performance of the algorithm [89].

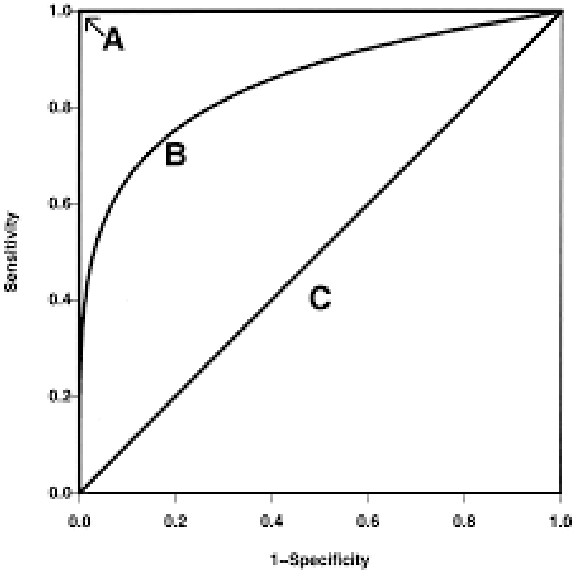


Figure 3.20: Example of several ROC curves. In this case, by best performance order curve A>B>C because there is a bigger area under the curve.

After reviewing the most recurring techniques used on image processing and analysis it is clear the importance of choosing the most adequate method to the data we are dealing with. Using some of the described evaluation parameters, we will evaluate our tool and compare it with a similar tool from the literature (FluoTracker).

Chapter 4

Previous Work

In this Chapter it will be presented some previous work developed in BRAINlab group including some algorithms regarding vesicle tracking problem. Firstly, we will analyze NeuronDynamics, a precursor of the first version of NeuronDyn [4]. The present work is the 2^{nd} version of NeuronDyn and it is based on the flaws of the first version, consisting on an improvement of the algorithm aiming to achieve more reliable results. This final version will be further described and the challenges to address in the future will be analyzed.

4.1 NeuronDynamics

As explained in point 3.3, this project is a follow up to a master's thesis [4] developed at FEUP. This way, NeuronDynamics algorithm consists on a precursor of the current algorithm which has been constantly improved in order to achieve better and more reliable results.

Image Acquisition. The video set used was provided by the Department of Functional Genomics, Vrije University, Amsterdam. It consisted in recordings of neurons conducting marked vesicles (NPY-EGFP) obtained by confocal microscopy. The dataset was composed of two videos, with a resolution of 399x201 $pixel^2$ and a video frame rate of one frame per second. The number of vesicles (true positives) per frame ranged between 25 and 29.

Algorithm Training. The first frame of the video was subjected to a visual enhancement operation (linear expansion of the histogram). Then, the user marks a predefined number of strong candidates for vesicles, a similar number of ambiguous candidates and the reference point for the movement (cellular body). Each user-marked point was used as a seed point to a region growing segmentation algorithm that allowed the extraction of four features for each vesicle: eccentricity, centroid coordinates, area and major axis length. The training method used a Bayesian approach with Mahalanobis distance, where both classes are obtained from the same image (with the same variance) to get a linear classifier.

Vesicle Segmentation and Classification. Each frame of the video, disregarding all objects next to the borders and with areas out of a user-predefined range, was enhanced by two sequential filters: 1) an average filter for smoothing the image and 2) a Laplacian of Gaussian (LoG) for high frequency emphasis. The result of the operation was subtracted to the original image using the criteria shown on the following equations:

$$I_p(i,j) = \begin{cases} I_{LOG}(i,j) & if \quad I_{LOG}(i,j) > T.max\{I_{LOG}(i,j)\} \\ 0 & otherwise \end{cases}$$

$$I_{out}(i,j) = I_p(i,j) - I_{in}(i,j)$$

Where T is a parameter to determine the threshold level, $I_{LOG}(i,j)$ is the output image from the enhancement step, $I_p(i,j)$ is the image after the threshold operation and $I_{in}(i,j)$ is the original image. For each segmented object the eccentricity and major axis length parameters were obtained, as well as the coordinates for its centroid. The objects were then classified using the fore-mentioned classifier.

Vesicle Tracking. The first step of the tracking stage was determining in which frame the number of vesicles was the lowest. This allows to keep a consistent number of vesicles in each frame and, at the same time, to avoid the influence of biological phenomena such as fusion, death, splitting and temporary disappearing in the z-plane. The tracking was then performed from that frame, to the first and to the last one. Each vesicle from each frame was compared to all the candidates of the next frame and the new position was determined using, on plane z, the Euclidean distance between two consecutive points $D_{v,k\rightarrow k+1}$, where the new coordinates are those who have the shortest distance to the previous position. This approach is derived from the NNA tracking method with a notorious improvement: no fixed velocity is required. The vesicle speed $S_{v,k\rightarrow k+1}$ was estimated (in pixels/second) by the following equation:

$$S_{v,k\rightarrow k+1} = \frac{D_{v,k\rightarrow k+1}}{VideoRate}$$

A vesicle is considered to be moving if it changes its position for at least a user-defined number of consecutive frames (typically three). If the new position is farther away to the cellular body than the last one, the vesicle is labelled as moving forward. If opposite, backwards, but can also be bidirectional or even stopped. The general preferred movement is determined as the most common vesicle movement in a global context, considering all the moving vesicles along all the frames. This result, as well as the global average velocity of each movement, average velocity and most common movement of each vesicle and vesicle velocity in each frame are outputs of the method and can be exported to a MS Excel file. In Figure 4.1 is a pipeline of the algorithm.

NeuronDynamics showed good performance considering the values of accuracy, sensitivity, specificity, precision and computational time (compared with FluoTracker and the ground-truth established manually), that are presented in Table 4.1. The low computational cost (182 ms/frame) is due to the less local information used. As most of the false positives are random noise or small cellular components with high fluorescence, discarding them increases the ratio of true positives and the precision of the detection [4].

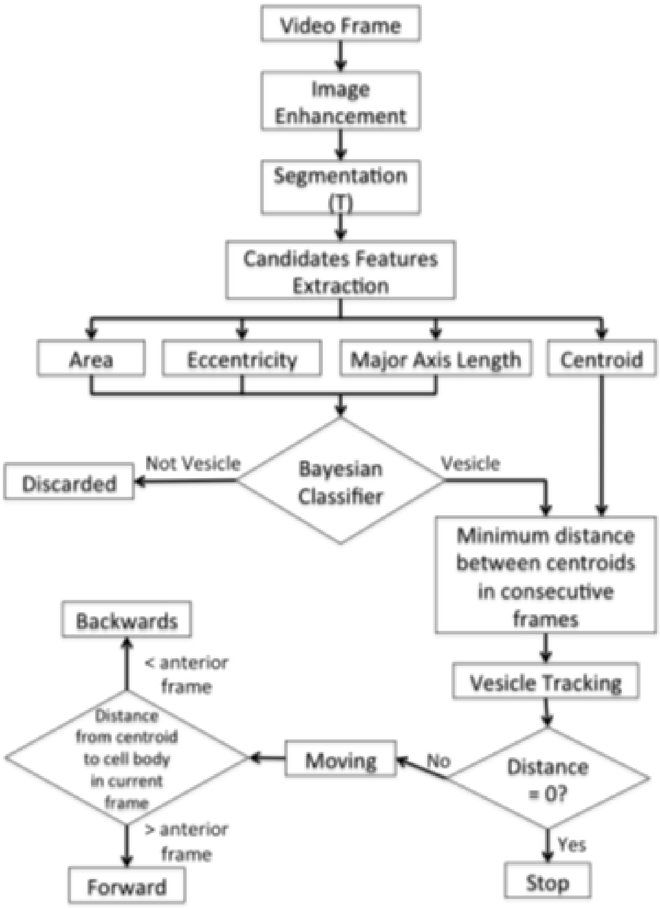


Figure 4.1: NeuronDynamics' pipeline [4].

Table 4.1: Performance comparison between Fluotracker and NeuronDynamics algorithms [4].

Algorithm	**Accuracy**	**Sensitivity/ Recall**	**Specificity**	**Precision**	**F1**	**F0.5**	**F2**	**Computational Time (s)**
NeuronDynamics	0.853	0.849	0.956	0.978	0.818	5	1.25	0.183
FluoTracker	0.414	0.434	0.403	0.285	0.344	5	1.25	125.2

4.2 NeuronDyn - First version

Improvements from NeuronDynamics.

The first version of NeuronDyn [4] was designed addressing the main problems of NeuronDynamics. This way the main improvements registered from NeuronDynamics to the first version of NeuronDyn included:

- In vesicle detection and segmentation. The parameter T used in vesicle detection is chosen automatically for each dataset unlike NeuronDynamics. This avoids user bias and saves time;
- Before the training stage, the user has the option to see the entire film and choose in which frame he wants to do the training stage. After this, the chosen frame is subject to a visual enhancement operation to increase the contrast between the vesicle and the background;
- Possibility to upload up to three videos simultaneously, using the same training stage to analyze all of the selected videos;
- The analysis can be performed on a single process of the neuron rather than in the entire image. This saves time when most of the objects of interest are located in a specific region of the image and it is pointless to spend time analyzing the rest of the image. Also reduces the influence of surrounding noise on the results;
- Bayesian approach for classification replaced by a more robust classifier based on artificial neural networks;
- Necessary adjustments on the GUI to include the above enumerated modifications.

Problems and open challenges.

Despite the good results obtained in this first version of NeuronDyn, there were still some flaws and issues that could be corrected/optimized. As the candidate identification step does not need to have high precision, a simpler method can be used improving algorithm's speed. The region growing methdod used for segmentation may become problematic if the stop condition is not independent from the dataset used. This can lead to infinite growing causing Matlab to freeze. Although the ANN classifier increased the results in terms of accuracy, specificity, sensitivity and precision, it is too expensive in terms of computational time and complex to implement. It is desirable to study other options that combine good results with low computational cost. Evaluation is done with small datasets and number of frames. It is crucial to enlarge the ground-truth available to ensure the reliability of the obtained results. The visualization of vesicles' path can only be done on a static image with their full path. A more interesting option would be to follow the evolution of their path along the video.

Chapter 5

NeuronDyn

5.1 Introduction

The goal of this project is to perform neurotransmitter vesicle segmentation and tracking in 2D confocal videos. The algorithm proposed uses an adaptive optimal threshold value for candidates identification and a support vector machines based approach for classification. The tracking module uses global nearest neighbor tracking method based in Hungarian algorithm [90] to associate the vesicles over sequences of confocal images. This research is a result of a cooperation between the Brain Lab group of INESC-TEC and Advanced Light Microscopy unit and Nerve Regeneration group from IBMC.The software was programmed using Matlab R2014a.

5.2 NeuronDyn Modules

In order to respond to the existent problems and to address the investigators' needs identified in Chapter 4 , a multistage approach was developed and is divided in several stages: Training, Candidate Detection and Classification, Vesicle Tracking and Quantification.

The NeuronDyn approach is schematized in the block diagram presented in Figure 5.1.

The algorithm starts by asking the user to upload the video and to select the desired type of analysis, either global (analysis of the entire frame) or by processes (choosing only a branch of the neuron). Then, the researcher can use an already trained classifier or train a new classifier. If the last case is chosen it must be selected a frame to perform the training stage where examples of vesicles and non-vesicles objects are marked. Based on this selection by the user, NeuronDyn performs the segmentation and extracts the candidates' characteristics. The classifier discards the candidates that are not vesicles and the algorithm performs the tracking of the remaining true vesicles to obtain some measures. Each block of NeuronDyn will now be presented with all its features. Their presentation will follow the exact path researchers will have to go through when using our tool.

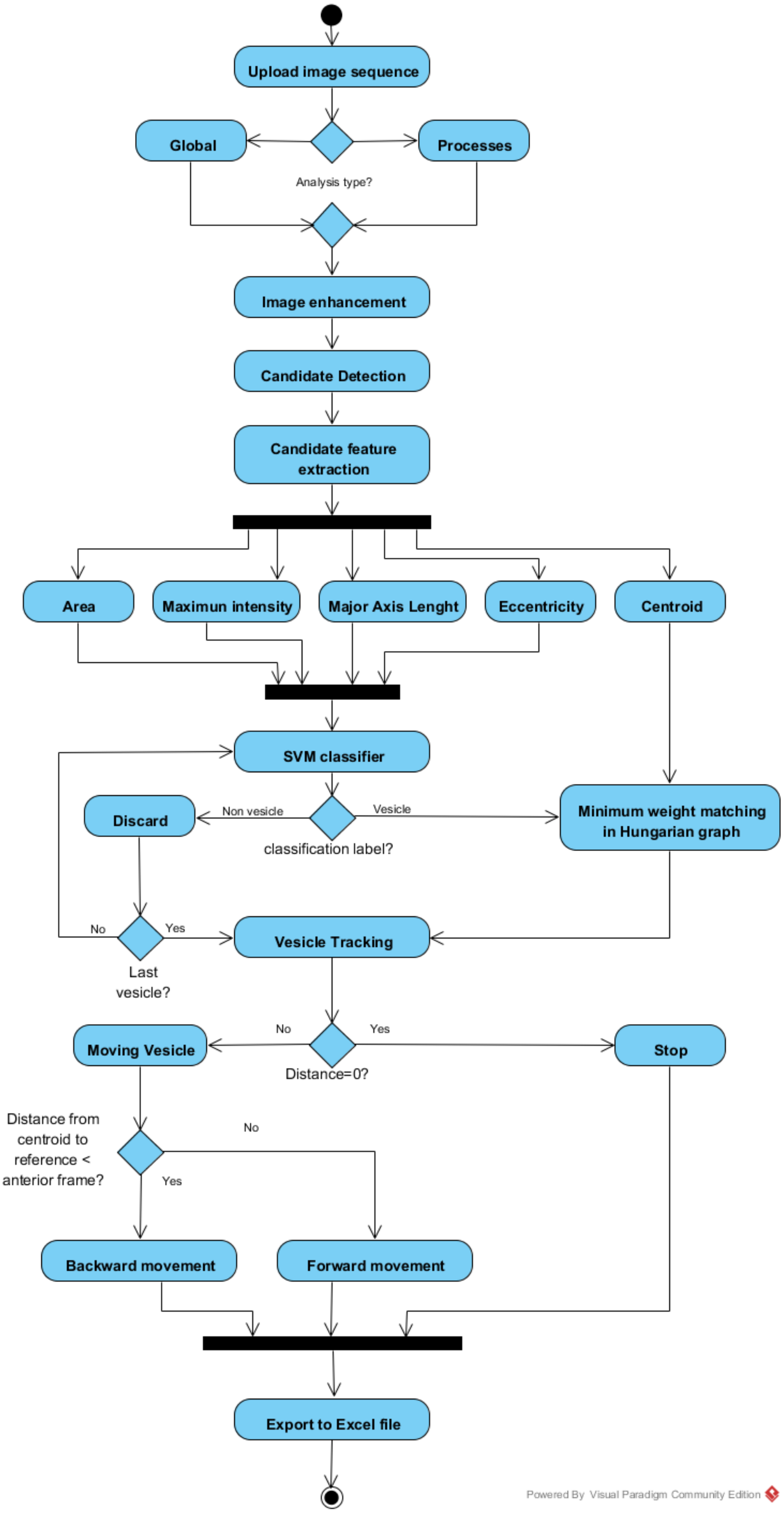


Figure 5.1: NeuronDyn's pipeline

5.2.1 Dataset Videos

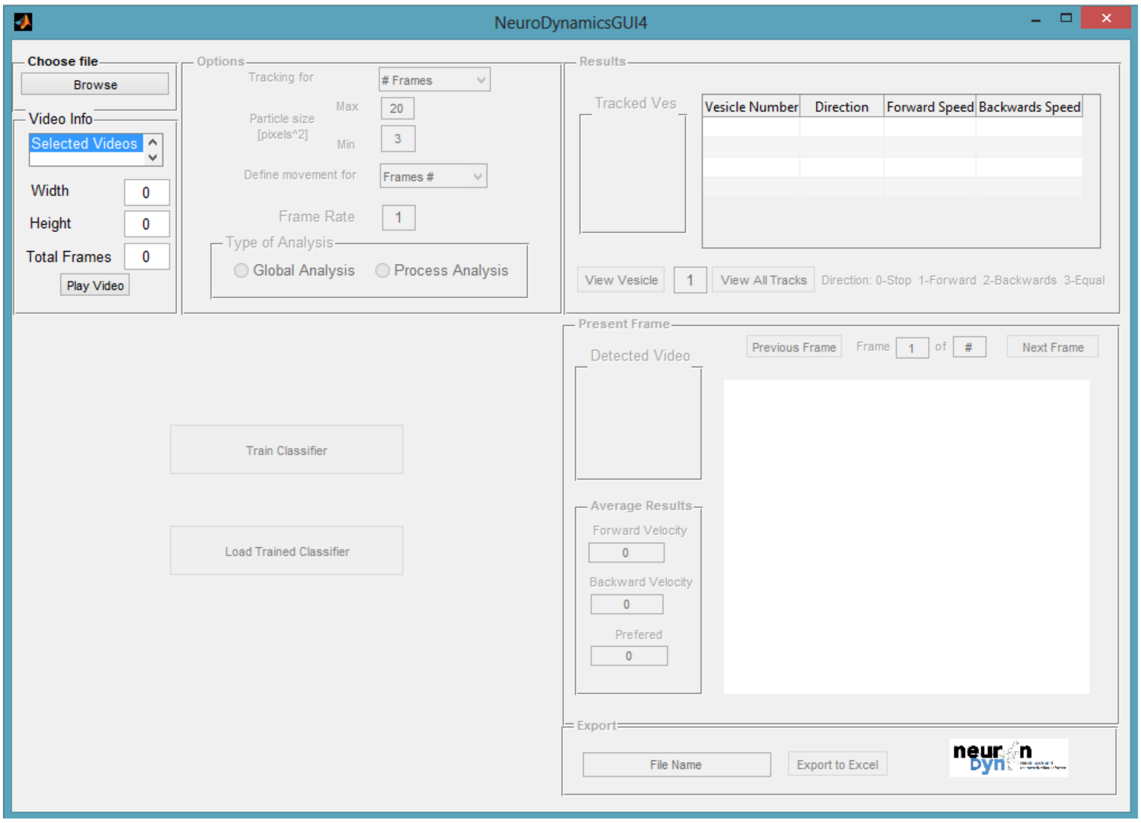
The dataset available for this project is composed of a total of 13 time-lapse confocal microscopy videos of living neurons. The first 10 videos were provided by IBMC while the last 3 were obtained from the Department of Functional Genomics, Vrije University, Amsterdam, with neurons conducting marked vesicles (NPY-EGFP). The number of true vesicles in each frame differs from one video to the other ranging from 15 to 25. Datasets 1, 2, 3, 4, 9 and 10 were manually labeled by experts and are our gold standard. The 13 videos' characteristics can be observed in Table 5.1.

Table 5.1: Dataset Characteristics

Dataset	**Resolution**	**Fram Rate (fps)**	**Number of Frames**
1 to 8	1024x1024	11	107
9	512x512	1	60
10	1024x1024	1	60
11	399x201	1	61
12	298x187	1	61
13	298x211	1	61

5.2.2 Input Videos to NeuronDyn

The first phase of the NeuronDyn algorithm is to upload one or more videos (up to three). This allows the user to run the algorithm and to do the training stage only one time, using the same classifier on all the selected videos, saving a considerable amount of time. To optimize this multivideo classification, all the videos should present similar characteristics. NeuronDyn's graphical user interface (GUI) is presented on Figure 5.2.



Width 0
Height 0
Total Frames 0

Tracking for # Frames
Particle size [pixels^2] Max 20 Min 3
Define movement for Frames #
Frame Rate 1

Vesicle Number	Direction	Forward Speed	Backwards Speed

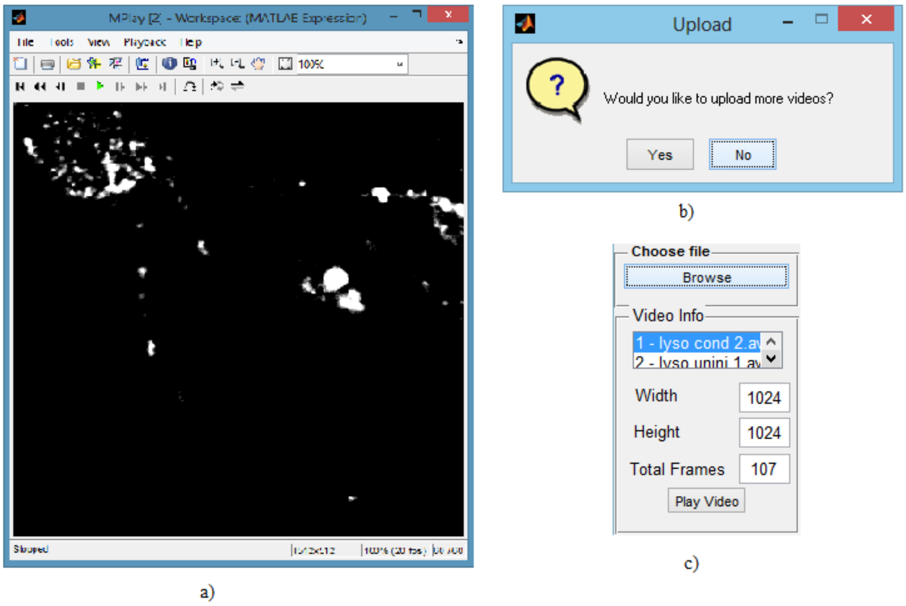
View Vesicle 1
Direction: 0-Stop 1-Forward 2-Backwards 3-Equal

Frame 1 of #

Forward Velocity 0
Backward Velocity 0
Prefered 0

Figure 5.2: NeuronDyn's Graphical User Interface

The initial block to upload the videos is constituted by a browse button and video information window. After the selection of the first video, a message window will appear, asking if the user wants to upload more videos (figure 5.3 b)). After the videos are selected and loaded, their name will appear on the selected videos list, where it is possible to see its width, height and number of frames. It is also possible to play each video in a separate window.



Would you like to upload more videos?

Width 1024

Height 1024

Total Frames 107

Figure 5.3: a) Video player in a new window; b) Message asking to upload more videos; c) Browsing button and video characteristics

This multi video approach simplifies researchers' work without compromising the results obtained once the objects usually have similar characteristics. The videos are saved in structures and can be reproduced multiple times before proceeding to the next step.

5.2.3 Parameters Selection

In this step, the user can select some parameters that will affect the tracking process, such as: the number of frames to track, the area of the particles to consider in the process (minimum and maximum value in $pixel^2$), the minimum number of consecutive frames without movement necessary for an object to be considered as non-moving vesicle, the frame rate, and the type of analysis: Global, for the entire image, or Process analysis, for a small portion of an axon (between two selected points).

Process Analysis

It is common that researchers only want to analyze a single neuron process. By selecting Process analysis option, the user can perform the following tasks only on a portion of the image, confining the area of interest to a smaller region that has higher density of objects to analyze. This time saving option is also a good way to reduce computational cost and to minimize the interference of noise on the area around the selected process.

The main steps of Process analysis are Path Detection and Process Selection:

Path Detection. To analyze the video by processes, first we need to know which region of the image has vesicle movement to create the path image of the neurons. This detection consists, firstly, on reducing the noise existent in each image so that its effects on the results are minimal.

We are currently using Wiener filter for this purpose, which allows the removal of background noise with good visual improvements in images with Gaussian distribution of noise. This adaptive wiener filter smooths the image in function of the local variance, i.e., higher variance producing lower smoothing and smaller variance, more smoothing. An adaptive filter obtains better results when compared with linear filtering, preserving the edges and other high frequency parts of the image. Also, a Gaussian filter was applied for visual enhancement. A Canny filter was applied to obtain the edges (the limits of the paths and objects) and then an absolute difference was made between consecutive frames. All the difference images resultant are overlapped and a global movement image is saved. The image is then submitted to a set of morphological operations to fill the holes and 10 pixels from each limit of the image are turned black to prevent the influence of noise. The objects are then labelled and the ones with area<100 $pixel^2$ (excluding objects which are certainly not vesicles) and with eccentricity<0.85 (vesicles present an approximately spherical shape) are eliminated. Figure 5.4 shows the pipeline of the Path Detection algorithm implemented for NeuronDyn.

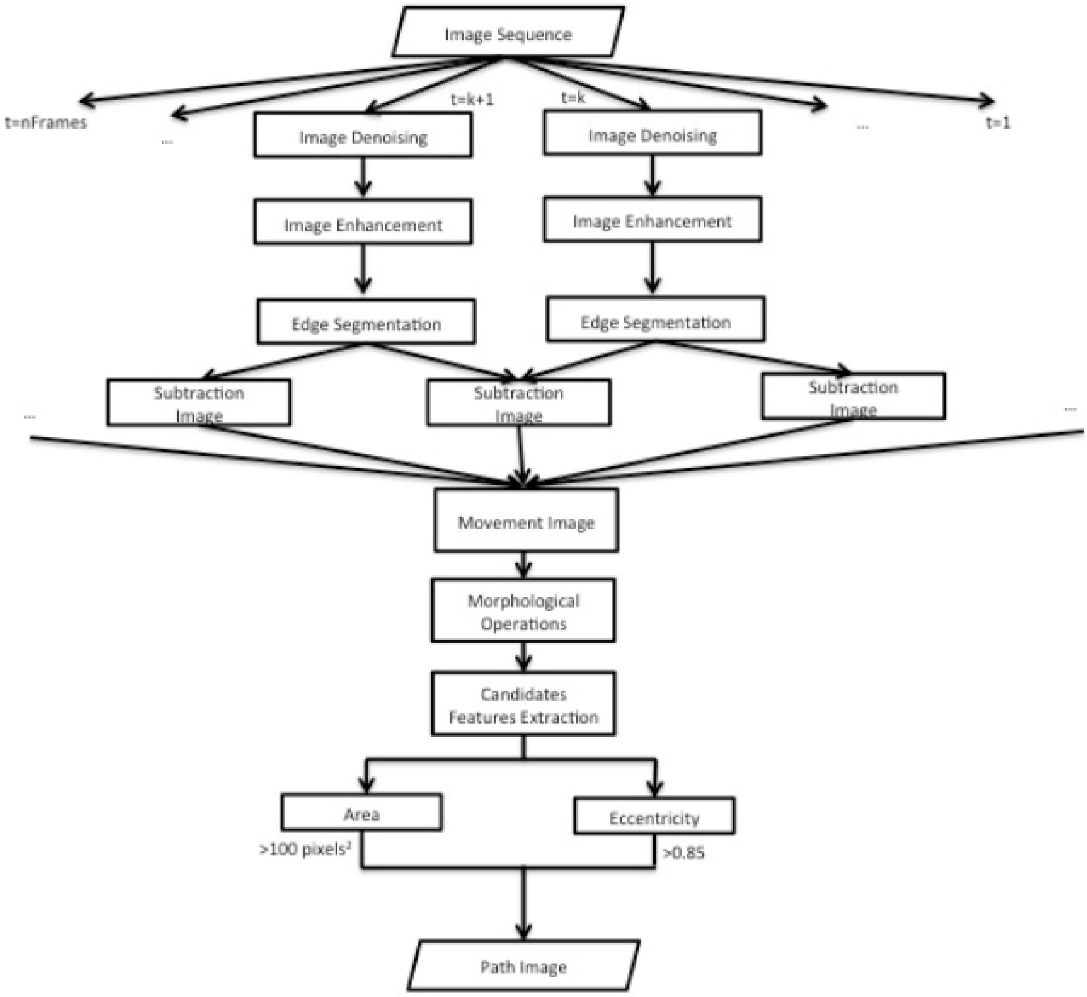


Figure 5.4: Path detection pipeline [4].

Process selection. To select the process to analyze, the user is asked to select a starting and end point on the original image. Figure 5.5 shows an example of that selection. Between the two selected points a region-growing function was used to find the path that connects the two points in the path image. If the points don't belong to the same axon an error message is sent, until the user selects two valid points.

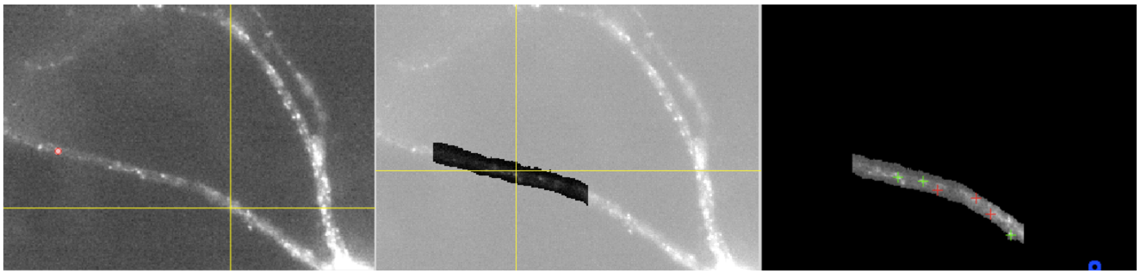


Figure 5.5: Process selection and vesicle training detection [4].

Global Analysis

In the global analysis, the entire image is subjected to a visual enhancement operation (linear expansion of the histogram). This option presents higher computational cost as the algorithm will consider candidates from the whole image and will label all of them according to their features. It may be the worst option if the goal is to obtain only a few representative examples of vesicle dynamics in that video.

5.2.4 Algorithm Training

In the training block, the user can choose to use an already trained classifier or instead to train the classifier at that moment using the selected video. In this last case, he can visualize each frame of the selected video and choose in which he wants to perform the training stage. Next, the user has to manually mark x examples of objects that are unequivocally vesicles and the same number of ambiguous objects. Each user-marked point is then used as a seed point to a region growing segmentation algorithm that allows the extraction of the centroid coordinates, object eccentricity, major axis length and maximum intensity value. The training method uses an SVM based classifier with linear kernel. The features used for classification were the major axis length and the maximum intensity values. Figure 5.6 shows GUI's training block and training frame chosen by the user, where the training input must be given.

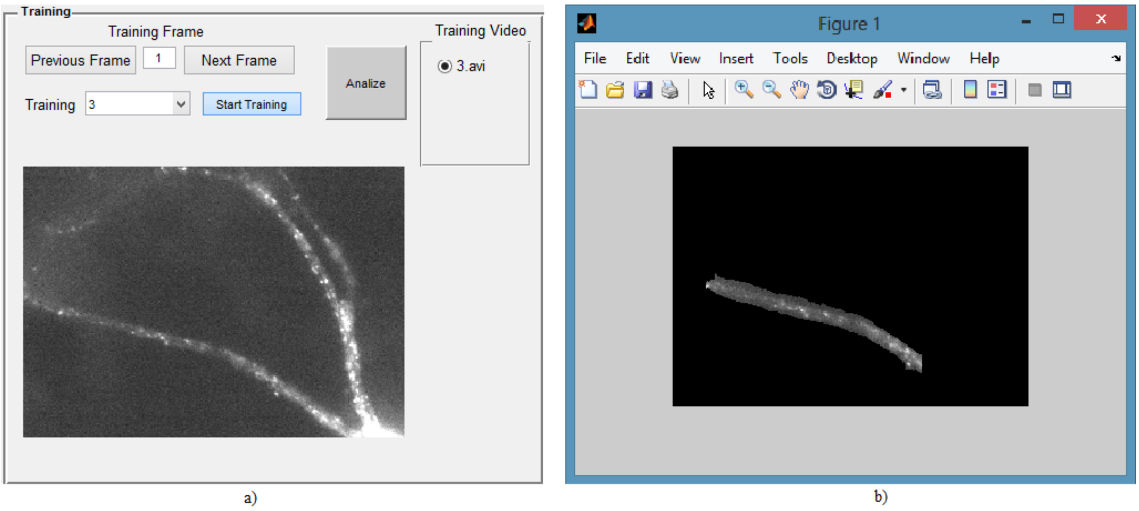


Figure 5.6: a) GUI's training block revealing some of the parameters given by the user; b) training frame chosen by the user, where the training examples are marked (in this case the user selected process analysis);

5.2.5 Candidates Identification

Despite being an essential step in this kind of problem, candidate identification does not need to be precise to the point of including only true vesicles, once the classification step will then exclude exceeding candidates. Usually, researchers prefer not to detect all the true vesicles rather than classify a non vesicle as a vesicle. This way, we opted for a relatively simple approach that could perform well with low computational cost.

This way, an adaptive algorithm was used to select an optimal segmentation threshold to separate the candidates to vesicles from the background. The segmentation threshold is selected through the following iterative procedure: Let T^i be the segmentation threshold at step i, obtained by calculating the mean intensity value between the two highest peaks on images histogram. To choose a new segmentation threshold, we apply T_i to the image to separate object and background pixels. Let μ_b and μ_n be the mean gray-level of the object pixels and background pixels after segmentation with T^i. Then the new threshold for step $i{+}1$ is

$$T^{i+1} = \frac{\mu_b + \mu_n}{2}$$

This iterative threshold update procedure is repeated until there is no change in the threshold, i.e., T^{i+1}=T^i . T^{i+1} is then selected as the optimal threshold value for that dataset [91].

The image is then submitted to a set of morphological operations to remove misdetections such as objects with an area much bigger than the average vesicle size or with high eccentricity values. This post-processing operation reduces possible artifacts which were considered candidates, saving time on the classification step.

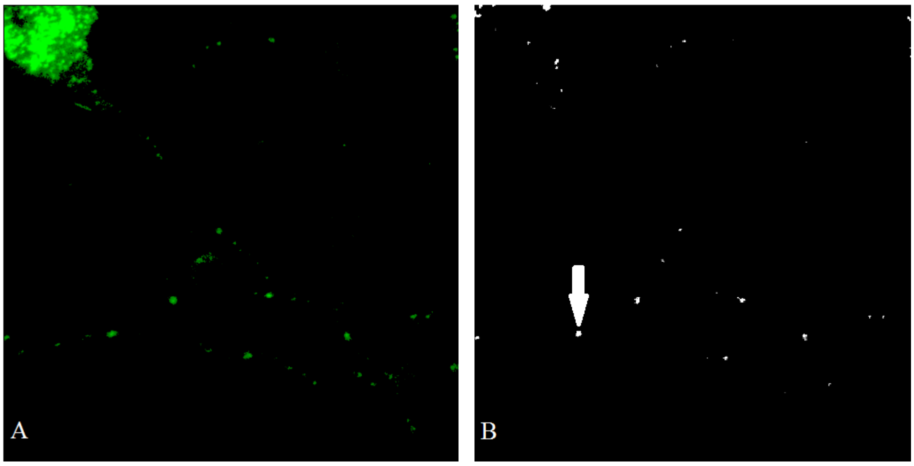


Figure 5.7: a) original image; b) candidates detected using optimal threshold value;

5.2.6 Classification

Before the tracking step, we need to find out which of the previously identified candidates are, indeed, vesicles and which of them are not. In order to do so, we are using a SVM based algorithm which, given a set of labelled training examples, constructs a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [70, 71]. Two general attributes define the SVM algorithm: C, a hyper-parameter which controls the trade-off between margin maximization and error minimization; and kernel, a function that maps training data into high-dimensional features spaces. The kernel function is used to train SVMs classifiers. The type of kernel function used is a key factor on the performance of SVM classification algorithm. The types which are more commonly used are the linear (Linear SVM) and the gaussian (Radial Basis Function, RBF) - RBF SVM [72]. In this phase of the project we are using a linear kernel SVM classifier.

5.2.7 Vesicle Tracking

Usually, researchers are not interested on quantifying the movement of every vesicle from every video. In fact they prefer to have only a few vesicles with profitable information, than to take the risk of having every vesicle present on the video but also some misclassified vesicles that can compromise the final results. The tracking method was chosen and designed taking this in consideration and also the low image quality and the differences of fluorescence that the same object can present in different frames. As only strong candidates are considered vesicles, one common problem can be the exclusion of some vesicles that show low intensity in a determined frame and, as a consequence, the tracing of two different paths instead of one. To overcome this issue, an optimal association for multi-target tracking based on the Hungarian method was performed. This hungarian method provides Global Nearest neighbour tracking algorithm the unique ability to deal with eventual gaps, once it consists on creating links amongst particle pars found to be the closest (euclidean distance). It is also ensured that the sum of the pair distances is minimized over all particles. Unlike the traditional nearest neighbour method, which simply links an object with its nearest neighbour from the next frame, this algorithm takes in consideration a user-defined number of frames choosing the most probable path. In Figure 5.8 it is possible to see and example of the paths detected.

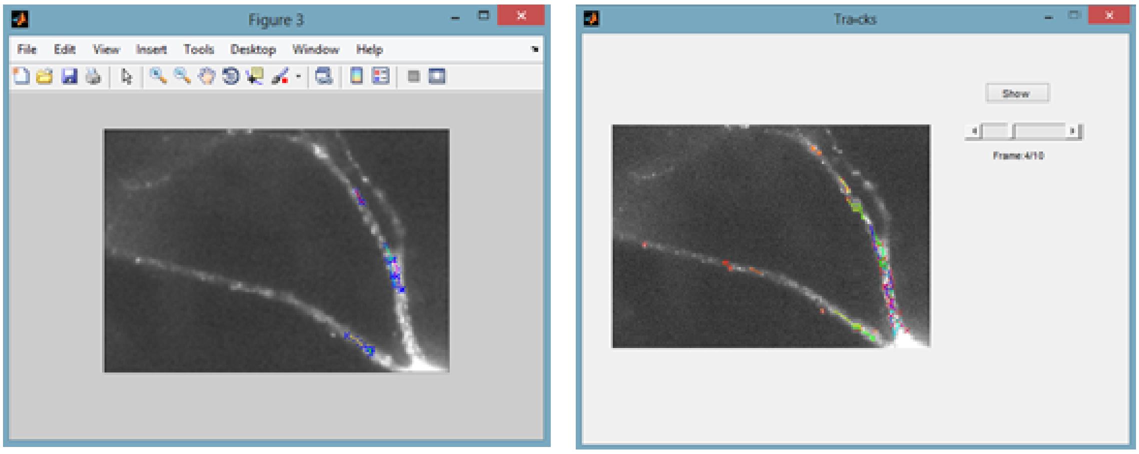


Figure 5.8: Static image of al detected paths (on the left) and possibility do see all paths frame by frame (on the right).

After obtaining the tracks, the vesicle speed is estimated (in pixels per second). The vesicle is then considered to be moving if it changes its position for at least a user-defined number of consecutive frames (three by default), if its position does not change than the vesicle is considered as stopped. If the new position is farther away to the cellular body (or the reference point, in the absence of cell body in the image) than the previous one, the vesicle is labelled as moving forward. Otherwise it is considered as moving backward and can even be considered as moving bidirectionally if the direction of the movement changes along the video. The general preferred movement is obtained as the most common vesicle movement in a global context, considering each moving vesicle along all the frames. These results, as well as the global average velocity of each movement and preferred movement of each vesicle are outputs of the algorithm. The output data is then exported to an Excel file and can be further analysed and compared with data from other videos. By default, the excel will be saved with the name of the program (NeuronDyn), and the date.

5.3 Results and Discussion

Denoising and Enhancement

To reduce the background noise and improve contrast a sequence of Weiner and Gaussian filters is used. In Figure 5.9 it is shown pixel intensities and the effect of each filter on them.

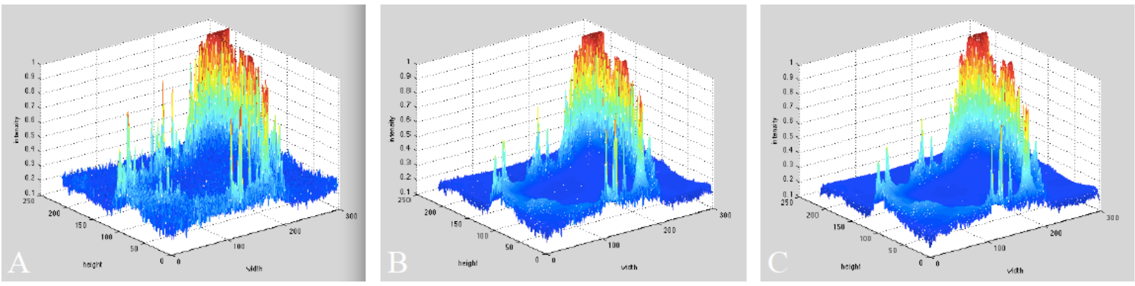
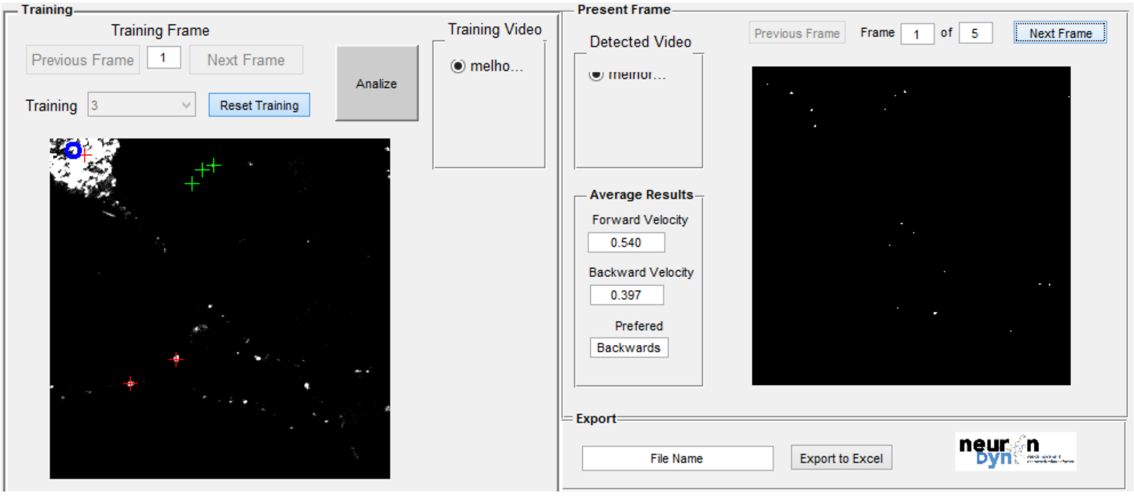


Figure 5.9: A) Acquired image with intensity profile from Dataset 3; B) Image produced after wiener filter; C) Image produced after Gaussian filter.

Detection and Segmentation

The core of the tracking procedure is candidate identification and classification. If this first steps are accurate, the following steps will have better results. However there are some aspects to consider that can difficult this task, including: the recordings are often noisy; the training stage (when performed) has to be very strict because it can compromise the following steps producing erroneous results.

Although we used a relatively simple method, the adaptive threshold approach performed well and was able to exclude a big part of the artefacts and noise. Figure 5.10 shows the detected vesicles in NeuronDyn based in the training choices.



Training Frame: 1

Training: 3

Training Video: ● melho...

Detected Video: ● melhor...

Frame 1 of 5

Forward Velocity: 0.540

Backward Velocity: 0.397

Prefered: Backwards

File Name

Figure 5.10: Detected vesicles (right image) based on the training choices (left image).

Classification

The SVM classifier performed well specially if we considered the low computational cost it presents. It is a very flexible method once you can optimize several parameters including the kernel used. An example of SVM calssifier using RBF kernel is shown in Figure 5.11

Tracking

The principle behind tracking in NeuronDyn is a Global Nearest Neighbor, where the association is performed by Hungarian method. This method produces better results when compared

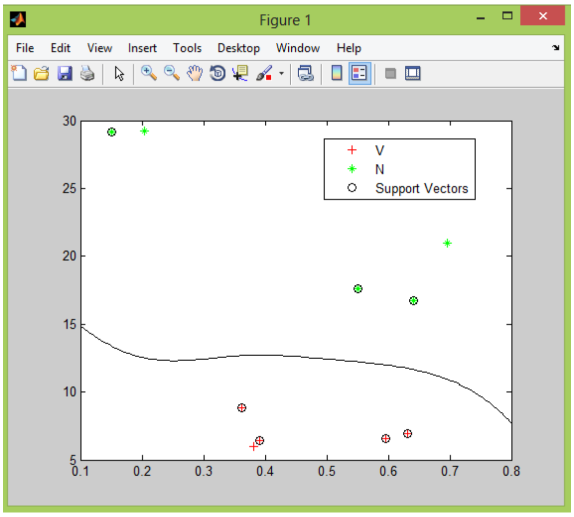


Figure 5.11: Classification performed by NeuronDyn on manually marked objects in training stage. The x axis represents eccentricity and the y axis major axis length.

with the naive nearest neighbor association, and considers some dynamic properties of the vesicles: birth and death, gaps and clusters. However, the algorithm does not consider split and merge events, not associating those vesicles. In the case of gaps, the algorithm searches for the next four frames to fill its misdetection, continuing the track. Figure 5.12 shows the result of detected and tracked vesicles.

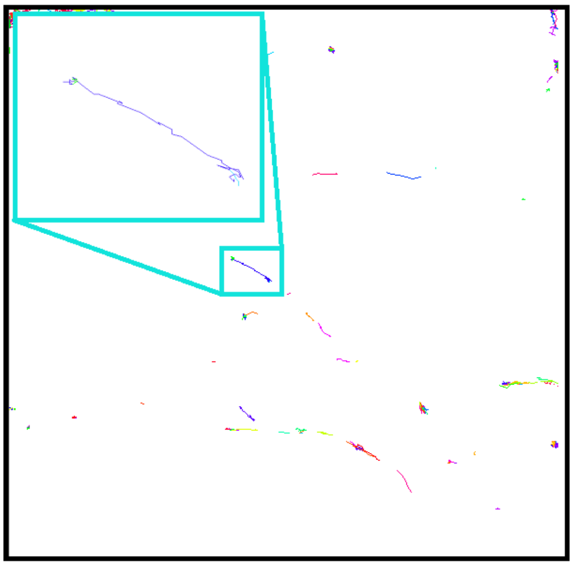
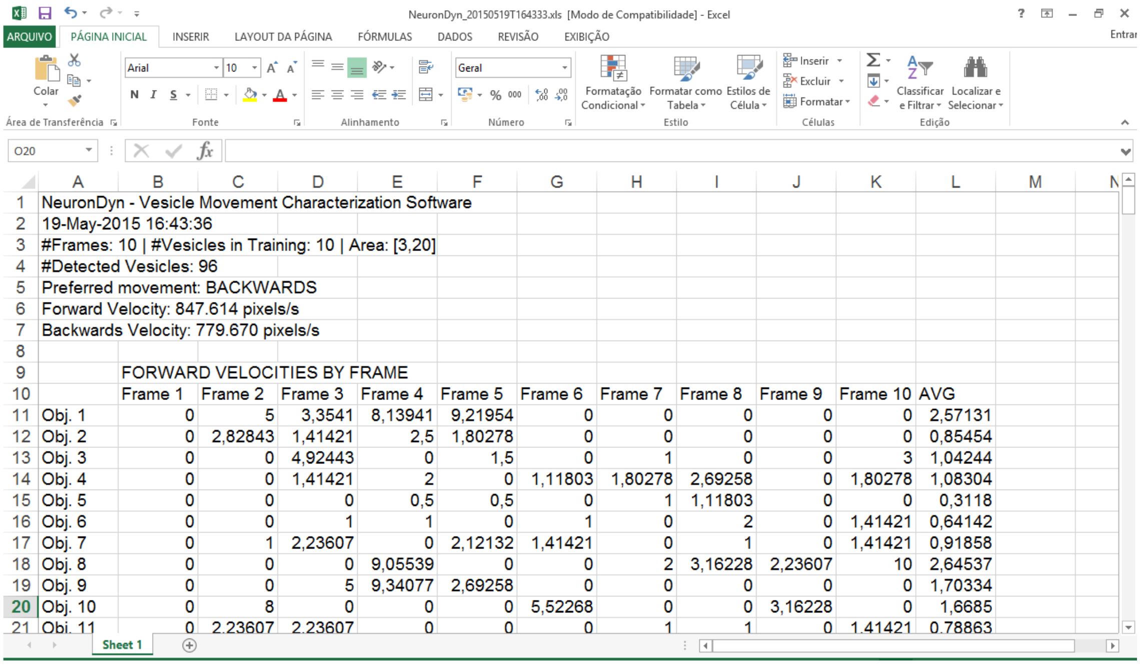


Figure 5.12: Vesicles path obtained after applying the tracking algorithm. Coloured lines represent the path of each vesicle. Highlighted on the figure is a zoomed view of one of the detected tracks.

The results are then exported to an Excel file (Figure 5.13), according to the user selected parameters. It is possible to visualize the date, selected parameters, preferred movement, forward and backwards velocities for each vesicle and the global values.



NeuronDyn - Vesicle Movement Characterization Software
19-May-2015 16:43:36
#Frames: 10 | #Vesicles in Training: 10 | Area: [3,20]
#Detected Vesicles: 96
Preferred movement: BACKWARDS
Forward Velocity: 847.614 pixels/s
Backwards Velocity: 779.670 pixels/s

FORWARD VELOCITIES BY FRAME

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	Frame 6	Frame 7	Frame 8	Frame 9	Frame 10	AVG
Obj. 1	0	5	3,3541	8,13941	9,21954	0	0	0	0	0	2,57131
Obj. 2	0	2,82843	1,41421	2,5	1,80278	0	0	0	0	0	0,85454
Obj. 3	0	0	4,92443	0	1,5	0	1	0	0	3	1,04244
Obj. 4	0	0	1,41421	2	0	1,11803	1,80278	2,69258	0	1,80278	1,08304
Obj. 5	0	0	0	0,5	0,5	0	1	1,11803	0	0	0,3118
Obj. 6	0	0	1	1	0	1	0	2	0	1,41421	0,64142
Obj. 7	0	1	2,23607	0	2,12132	1,41421	0	1	0	1,41421	0,91858
Obj. 8	0	0	0	9,05539	0	0	2	3,16228	2,23607	10	2,64537
Obj. 9	0	0	5	9,34077	2,69258	0	0	0	0	0	1,70334
Obj. 10	0	8	0	0	0	5,52268	0	0	3,16228	0	1,6685
Obj. 11	0	2,23607	2,23607	0	0	0	1	1	0	1,41421	0,78863

Figure 5.13: Example of excel file with calculated vesicles' movement measures.

The classification algorithm was evaluated based on some standard parameters and the results were obtained by comparing masks created by the algorithm with the ones resultant from manual marking. In a first phase of evaluation, it was used a limited number of datasets in order to allow the comparison with a similar method from the literature (FluoTracker). The results of this step are presented in Table 5.2.

Table 5.2: Classifier Preliminary Evaluation

	FluoTracker [82]	**NeuronDynamics [92]**	**NeuronDyn (1st version) [4]**	**NeuronDyn (final version)**
Accuracy	41%	85%	99%	74%
Sensitivity	43%	85%	92%	89%
Specificity	40%	96%	99%	97%
Precision	29%	98%	77%	85%
Computational Cost (s/frame)	125.2	0.183	20.18	0.245

By analyzing table 5.2 we easily observe that the results obtained with NeuronDynamics were considerably better than the ones obtained by a similar method on the literature (Fluotracker). From Neurondynamics to the first version of Neurondyn there was also an improvement on the results, although they were obtained at a much higher computational cost, mainly due to the utilization of an ANN based classifier. This way, one of the main points to improve on NeuronDynV1 was exactly to reduce computational cost without compromising the results. Once more by examining the table it is clear that the new version of the algorithm was able to maintain good results at lower cost. However, the dataset used in this phase was still small and further testing was needed to support this preliminary results.

Bearing this in mind, the plan was to use datasets 1 to 10 to perform a more reliable evaluation but researchers suggested to exclude datasets 5 to 8 due to presence of noise and doubt about the relevant information that could be extracted, even using manual marking. This way, it was used a total of 6 datasets (number 1,2,3,4,9 and 10 from Table 5.1) with a total of 548 analyzed frames. As the datasets used were provided by IBMC exclusively for our project, we were not able to continue comparing the tools developed by our group with FluoTracker.

Table 5.3 shows the results obtained by comparing all versions of our tool using the above mentioned dataset.

Table 5.3: Classifier Final Evaluation

	NeuronDynamics [92]	**NeuronDyn (1^{st} version) [4]**	**NeuronDyn (final)**
Accuracy	69%	81%	70%
Sensitivity	74%	79%	67%
Specificity	79%	87%	82%
Precision	70%	80%	73%
Computational Cost (s/frame)	0.25	30.46	0.31

After analyzing these final results we conclude that NeuronDynamics' results dropped considerably and it can be explained by several factors, including the utilization of a linear classifier which is more susceptible to errors when adapting to new data or the different characteristics between the videos used on the first evaluation and on the last one (NeuronDynamics was optimized based on the available videos at that time).

Both versions of NeuronDyn revealed more universality, once they were able to maintain a good performance even with the inclusion of more datasets with higher resolution. This adaptability is crucial because the videos we want to analyze in the future may present slight differences from the ones available when the tool was developed. The most relevant distinction between the first and second version is still centered on the amount of time consumed by the algorithm. This aspect becomes even more important after multiplying the elapsed time in each frame by the total number of frames in a video.

In general, the results show the classifier performed well although there are always improvements to make. The low sensitivity value does not represent a big concern once it is mainly caused by a high number of false negatives. This fact was expected once the algorithm was designed bearing in mind researchers' advice that it is preferable not to detect every vesicle than to include non-vesicles in the results. Moreover, researchers generally only need to analyse some vesicles of each video, enough to be a representation of general vesicles' behaviour.

Chapter 6

Conclusions

The advances in neuroscience and the increasing challenges in the neuronal transport allowed the development of solutions to better analyse and characterize particle's movement, however it is still done manually. In the present work, a semi-automatic vesicle movement characterization algorithm was presented. It is known that manual vesicle counting and tracking is a tedious, obsolete and time-consuming task. For these reasons, it is important to keep developing existent tools in order to get acceptance from researchers and to implement the utilization of these tools as a standard routine in researchers work. Semi-automatic algorithms combine the quickness of automatic analysis and the know-how of experienced users. However, some issues are associated to the tracking procedure, especially due to some vesicles events, misdetections and collisions of objects.

Due to the large number of different conditions in image acquisition and cell/neuron features and behaviours, designing a universal tracking system is impractical. However, NeuronDyn can be improved in the future, mainly considering more robust tracking algorithms and implementing it in a new programming language. The GUI can always be improved in order to make it more user friendly.

Some of the latest NeuronDyn improvements include:

- Simpler candidate detection method using an adaptive threshold value;
- Region-growing method for candidates' segmentation was replaced by an active contours-based approach, which solved the problem of infinite growing and consequent freezing of Matlab.
- More robust classification approach based on SVM, which also performed with lower computational cost (the computational cost using ANN was 20.17 s/frame and with SVM was 0.24s/frame);
- Possibility to use an already trained classifier rather than train one every time we run the algorithm;
- Vesicles' path can be reviewed along time instead of presenting it on a static image;

- Some more punctual improvements in terms of algorithm's quickness and stability and in GUI aspect (including the the alterations mentioned above);
- Evaluation was done using more datasets which ensures the reliability of the results obtained.

We are currently gathering users' feedback related to practical utilization of the tool but so far the reactions are very positive.

Appendix A

EMBC 2015 - Neurotransmitter Vesicle Movement Dynamics in Living Neurons (Accepted for publication)

Neurotransmitter Vesicle Movement Dynamics in Living Neurons*

Hélder T. Moreira, *Student Member, IEEE, EMBS*, Ivo M. Silva, Mónica Sousa, Paula Sampaio and João Paulo Silva Cunha, *Senior Member, IEEE, EMBS*

Abstract— **The communication between two neurons is established by endogenous chemical particles aggregated in vesicles that move along the axons. It is known that an abnormal transport of these vesicles is correlated with neurodegenerative diseases. The quantification of the dynamics of vesicles movement can therefore be a window to study early detection of such diseases. Nevertheless, most of the studies in the literature rely on manual tracking techniques. In this paper we present a novel methodology for quantifying neurotransmitter vesicle dynamics by using a combination of image tracking and classification algorithms. We use confocal microscopy videos of living neurons to detect and classify vesicles using support vector machine (SVM), while motion is extracted via global nearest neighbor (GNN) tracking approach. Results of the classification algorithm are presented and compared to a ground truth dataset defined by experts. Sensitivity of 90% and specificity of 97% were obtained at a much lower computational cost than an established method from the literature (0.24s/frame vs. 125s/frame). These preliminary results suggest the great potential of the method and tool we have been developing for single particle movement dynamics measure in living cells.**

I. Introduction

Communication between and inside cells is a vital function for human organism. In fact, intracellular transport of organelles is a fundamental process and despite being intensively studied, it is still an open challenge for researchers.

Neurons are highly differentiated cells composed of a cell body, dendrites and axon, and are responsible for transporting information from and to the brain. The communication between two neurons is established by endogenous chemical particles called neurotransmitters, which travel along neurons axon aggregated in vesicles [1]. It is known that an abnormal transport of these vesicles is correlated with neurodegenerative diseases such as spastic paraplegia, Charcot Marie Tooth, amyotrophic lateral sclerosis (ALS), Alzheimers, Huntingtons and Parkinsons. In fact studies show that altered vesicles dynamics occur even before the first common symptoms are detected [2], [3].

With the rapid evolution of microscopy techniques and computer science, it is now possible to acquire dynamic images of moving cells, including neurons, and automatically process them in order to extract a set of features that may considerably facilitate researchers work.

To date, the characterization of vesicle movement has remained a manual process where the researcher has to individually mark every vesicle on each frame of the video. Such a laborious process becomes impractical once the number of vesicles per frame is usually large. In our work, we leverage recent advancements made in the area of object tracking to be able to automatically detect, classify and extract relevant measures (such as backward and forward velocity and morphology features) from vesicles [4], [5]. There are already some tools for cell and particle tracking in the literature, developed for different platforms (Windows, Mac, ImageJ, Matlab, between others) [6]. FluoTracker is one of the most relevant contributions in this area. It consists on a recursive Bayesian estimation algorithm that exploits intrinsic information contained in an image sequence. The algorithm is sequential and uses information extracted from previous frames to predict the most likely object position. The objects are detected and tracked robustly despite complicating factors inherent to biological samples. In order to track a variable number of objects with different movement characteristics, it uses multi-hypothesis tracking to render the approach computationally feasible [7].

However, most of the algorithms are fully automatic and are not easily adapted to one specific problem. This could constitute a disadvantage due to the wide variety of specifications that different videos can have. Besides, automation can make them unattractive for researchers, which usually like to have some control over the process.

Resulting from cooperation between BRAINlab engineering group from INESC-TEC Porto and IBMC - Instituto de Biologia Molecular e Celular - Advanced Light Microscopy unit and Nerve Regeneration group, where the need of the project emerged, the main goal of this project is the creation of a neuroscientist friendly computational tool to help researchers achieving better and faster results in cell particle tracking experiments.

Our approach consists in a semi-automatic tool which is based on the combination of three methods: adaptive threshold algorithm to detect candidates to vesicles; SVM algorithm which classifies candidates into vesicles or non-vesicles based on their characteristics; and global nearest neighbor algorithm that captures vesicles trajectories over time. Combined in an intuitive GUI, these three technologies

*This work was supported by the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project UID/EEA/50014/2013 granted to INESC TEC.

Hélder Moreira is with FEUP, University of Porto, Porto, Portugal (email: up201306956@fe.up.pt).

Ivo Silva is with FEUP, University of Porto, Porto, Portugal (e-mail: ivofmsilva@gmail.com).

Mónica Sousa is with IBMC, Porto, Portugal (email: msousa@ibmc.up.pt).

Paula Sampaio is with IBMC, Porto, Portugal (email: sampaio@ibmc.up.pt).

João Paulo Silva Cunha (corresponding author) is with INESC-TEC Porto, and FEUP, University of Porto, Porto, Portugal (e-mail: jcunha@ieee.org).

provide a unique ability to identify vesicles path and obtain dynamic measures. The present work is an evolution of a previous contribution of our group [8].

II. Methods

A. Dataset

The dataset available for this project is composed of a total of 13 videos of time-lapse confocal microscopy images of living neurons with characteristics presented on Table I. The first 10 videos were provided by IBMC while the last 3 were obtained from the Department of Functional Genomics, Vrije University, Amsterdam, with neurons conducting marked vesicles (NPY-EGFP). The number of true vesicles in each frame differs from one video to the other ranging from 15 to 25. Datasets 9, 10 and 11 are manually labeled by experts and are our gold standard. The other datasets will be manually labelled by experts in a second phase of the project.

TABLE I
Dataset Characteristics

Dataset	Resolution	Frame Rate (fps)	Number of Frames
1 to 8	1024x1024	11	107
9	512x512	1	60
10	1024x1024	1	60
11	399x201	1	61
12	298x187	1	61
13	298x211	1	61

B. Candidate detection & classification

Firstly, an adaptive algorithm was used to select an optimal segmentation threshold to separate the candidates to vesicles from the background. The segmentation threshold is selected through the following iterative procedure: Let Ti be the segmentation threshold at step i, obtained by calculating the mean intensity value between the two highest peaks on images histogram. To choose a new segmentation threshold, we apply T^i to the image to separate object and background pixels. Let mu_b and mu_n be the mean gray-level of the object pixels and background pixels after segmentation with T^i. Then the new threshold for step $i+1$ is $T^{i+1} = \frac{\mu_b+\mu_n}{2}$.

This iterative threshold update procedure is repeated until there is no change in the threshold, i.e., $T^{i+1} = T^i$. T^{i+1} is then selected as the optimal threshold value for that dataset [9].

Before the tracking step, we need to find out which of the candidates are, indeed, vesicles and which of them are not. Given a set of labelled training examples, an SVM training algorithm constructs a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [10], [11]. Two general attributes define the SVM algorithm: C, a hyper-parameter which controls the trade-off between margin maximization and error minimization; and kernel, a function that maps training data into high-dimensional features spaces. The kernel function is used to train SVMs classifiers. The type of kernel function used is a key factor on the performance of SVM classification algorithm. The types which are more commonly used are the linear (Linear SVM) and the gaussian (Radial Basis Function, RBF) - RBF SVM [12]. In this phase of the project we are using a linear kernel SVM classifier.

C. Tracking & obtained Measures

The GNN tracking algorithm not only relates the nearest object from one frame to other, but it also relates the information of a defined number of frames. This method can deal with gaps, which happen when one particle that is detected in one frame is not detected in the subsequent one, appearing in a further frame. This algorithm can efficiently identify and associate targets in complex state, such as targets with parallel movement, targets with intersecting movement, and targets with turning movement, but it only associates at most one target point, ignoring divisions. In the case of gaps, the algorithm searches for the next four frames to fill its misdetection, continuing the track [13], [14].

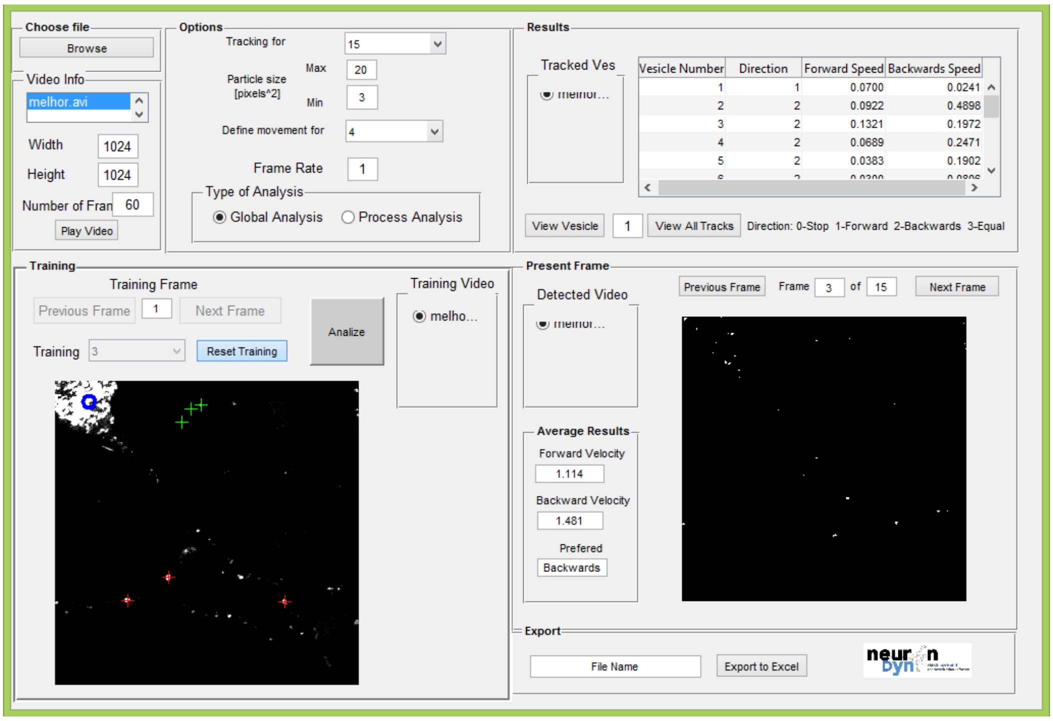
Each vesicle from each frame is compared to all the candidates of the next frame and the new position is determined using the Euclidean distance between two consecutive points ($D_{v,k\rightarrow k+1}$) where the new coordinates are those who have the shortest distance to the previous position. This approach is derived from the nearest neighbor tracking method with a notorious improvement: no fixed velocity is required. The vesicle speed ($S_{v,k\rightarrow k+1}$) was estimated (in pixels/second) by $D_{v,k\rightarrow k+1} = \frac{S_{v,k\rightarrow k+1}}{VideoRate}$

A vesicle is considered to be moving if it changes its position for at least a user-defined number of consecutive frames (typically three). If the new position is farther away to the cellular body than the last one, the vesicle is labelled as moving forward. If opposite, backwards, but can also be bidirectional or even stopped. The general preferred movement is determined as the most common vesicle movement in a global context, considering all the moving vesicles along all the frames. This result along with the global average velocity of each movement, average velocity, most common movement of each vesicle and vesicle velocity in each frame are outputs of the method.

D. Graphical User Interface

In order to enhance the usability and facilitate the interaction between the user and the algorithm, we also developed a GUI, shown in Figure 1.The GUI utilization intends to be as simple and intuitive as possible to be biologists friendly.

In a first step, the user selects the video/dataset he/she wants to analyze, being able to select up to three videos, saving time in the training step (once the features extracted may be similar from one video to another). After this, some parameters can be defined in order to optimize the whole process, including: number of frames to perform the tracking, maximum and minimum particle size, maximum gap size (in



melhor.avi
Width 1024
Height 1024
Number of Fran 60
Tracking for 15
Particle size [pixels^2] Max 20 Min 3
Define movement for 4
Frame Rate 1
Type of Analysis: ● Global Analysis ○ Process Analysis
Training 1 3

Vesicle Number	Direction	Forward Speed	Backwards Speed
1	1	0.0700	0.0241
2	2	0.0922	0.4898
3	2	0.1321	0.1972
4	2	0.0689	0.2471
5	2	0.0383	0.1902

View Vesicle 1
Direction: 0-Stop 1-Forward 2-Backwards 3-Equal
Frame 3 of 15
Average Results
Forward Velocity 1.114
Backward Velocity 1.481
Prefered Backwards
File Name

Fig. 1. GUI developed in this project.

frames) in which an object can disappear from the image and still be considered a true vesicle, frame rate of the video, and whether the user wants to perform global image analysis or only in one specific process of the neuron.

In the training step there are two options available, either the user does the training from zero or uploads the classifier already trained from a previous utilization of the algorithm.

III. Results

Visual results of the several steps of the algorithm are shown in Figure 2. It shows an example of the dataset used in this project, which was obtained from a living neurons culture by confocal microscopy, and the preliminary results of the segmentation step using an optimal threshold value to encounter candidates to vesicles.

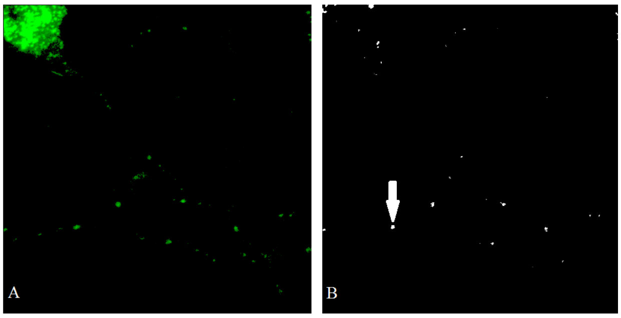


Fig. 2. A) Example of available dataset used in this project; B) Candidates to vesicle detected by optimal threshold value (examples from dataset 5).

After this segmentation, candidates are analyzed with SVM classification algorithm and grouped in two clusters whether they represent true vesicles or artifacts which were not excluded on the first step due to their similar intensity. After the classification stage, vesicles path is tracked along the previously defined frames and the results are presented in a table similar to Table II.

TABLE II
Obtained Measures

Vesicle Number	Preferred Direction[a]	Forward Speed (pixels/s)	Backward Speed (pixels/s)
1	1	0.0700	0.0241
2	2	0.0922	0.4898
3	2	0.1321	0.1972
...			

[a] 0-stop; 1-forward; 2-backward; 3-equal;

It is also possible to visualize the detected vesicles path on a static image or to follow its evolution along frames. The results shown above can further be exported to an excel document for posterior analysis. Figure 3 is an example of the resulting image after the tracking algorithm is applied. One of the vesicles path detected is zoomed and highlighted in red.

In order to evaluate the performance of SVM algorithm, we calculated some standard parameters such as: accuracy, sensitivity, specificity, precision and computational cost [15], [16]. Currently, the user is required to define the training frame in which he/she marks x true vesicles and a similar number of ambiguous vesicles (x is a number also defined by the user). The remaining frames are then used as testing data. In this phase of the project, we are using datasets 9, 10 and 11 to perform the evaluation of the classification. The training

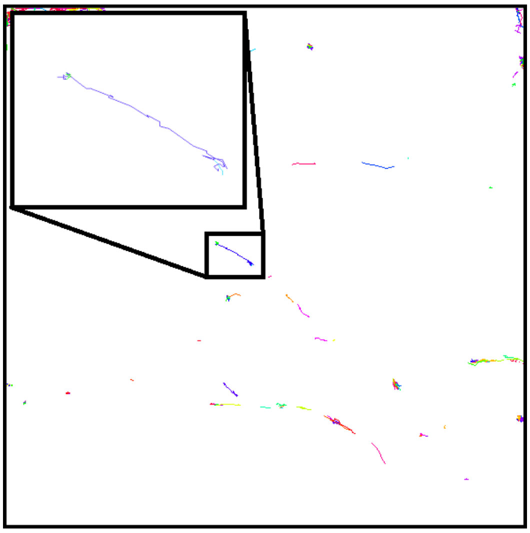


Fig. 3. Vesicles path obtained after applying the tracking algorithm. Coloured lines represent the path of each vesicle. Highlighted on the figure is a zoomed view of one of the detected tracks.

stage is currently performed every time the program runs due to the still limited ground truth provided by researchers.

The results are presented in Table III were obtained by marking 10 true vesicles and 10 ambiguous vesicles as training data.

TABLE III
CLASSIFIER EVALUATION

	Fluotracker [7]	**NeuronDyn**
Accuracy	41%	74%
Sensitivity	43%	90%
Specificity	40%	97%
Precision	29%	85%
Computational Cost (s/frame)	125.2	0.24

IV. Discussion and Conclusion

Preliminary results indicate that the classification algorithm performs well and with low computational cost, nevertheless the dataset is still small. As this is a work in progress, the tracking algorithm is still to be improved with a larger dataset and by comparing obtained measures with those obtained by manual tracking. With this contribution, we achieved good results in terms of vesicle classification and considerably faster than the traditional way and than similar tools such as Fluotracker [7]. The project will now progress to enlarge the ground truth dataset and evolve the computational tool so that it can be integrated in the daily usage of the neurobiology lab and contribute to speed-up their data analysis procedures.

V. Acknowledgment

The authors wish to thank the Department of Functional Genomics, Vrije University, Amsterdam for providing datasets 11 to 13 and the members of Nerve Regeneration group of IBMC, Porto, Portugal for the remaining datasets and for the permanent support and feedback.

References

[1] G. A. Morfini, M. R. Burns, D. L. Stenoien, and S. T. Brady, "Axonal transport," *Basic Neurochemistry: Principles of Molecular, Cellular, and Medical Neurobiology*, p. 146, 2011.

[2] S. Millecamps and J.-P. Julien, "Axonal transport deficits and neurodegenerative diseases," *Nature Reviews Neuroscience*, vol. 14, no. 3, pp. 161–176, 2013.

[3] M. Potokar, N. Vardjan, M. Stenovec, M. Gabrijel, S. Trkov, J. Jorgačevski, M. Kreft, and R. Zorec, "Astrocytic vesicle mobility in health and disease," *International journal of molecular sciences*, vol. 14, no. 6, pp. 11 238–11 258, 2013.

[4] J. Deschamps, V. Kantsler, E. Segre, and V. Steinberg, "Dynamics of a vesicle in general flow," *Proceedings of the National Academy of Sciences*, vol. 106, no. 28, pp. 11 444–11 447, 2009.

[5] W. W. Ahmed, B. J. Williams, A. M. Silver, and T. A. Saif, "Measuring nonequilibrium vesicle dynamics in neurons under tension," *Lab on a Chip*, vol. 13, no. 4, pp. 570–578, 2013.

[6] E. Meijering, O. Dzyubachyk, I. Smal *et al.*, "Methods for cell and particle tracking," *Methods Enzymol*, vol. 504, no. 9, pp. 183–200, 2012.

[7] J. H. Broeke, H. Ge, I. M. Dijkstra, A. T. Cemgil, J. A. Riedl, L. N. Cornelisse, R. F. Toonen, M. Verhage, and W. J. Fitzgerald, "Automated quantification of cellular traffic in living cells," *Journal of neuroscience methods*, vol. 178, no. 2, pp. 378–384, 2009.

[8] F. A. Carpinteiro, P. M. Costa, M. S. Espinoza, I. M. Silva, and J. P. Cunha, "Neurondynamics: A method for neurotransmitter vesicle movement characterization in neurons," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*. IEEE, 2014, pp. 481–484.

[9] S. Hu, E. A. Hoffman, and J. M. Reinhardt, "Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 6, pp. 490–498, 2001.

[10] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] Y. Liu and Y. F. Zheng, "Fs_sfs: A novel feature selection method for support vector machines," *Pattern recognition*, vol. 39, no. 7, pp. 1333–1345, 2006.

[13] B. Banitalebi and H. Amiri, "An improved nearest neighbor data association method for underwater multi-target tracking."

[14] B. Mourllion, D. Gruyer, C. Royere, and S. Théroude, "Multi-hypotheses tracking algorithm based on the belief theory," in *Information Fusion, 2005 8th International Conference on*, vol. 2. IEEE, 2005, pp. 8–pp.

[15] C. Smochină, "Image processing techniques and segmentation evaluation," 2011.

[16] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, 2013.

Appendix B

1st DCE FEUP - Neurotransmitter Vesicle Movement Dynamics in Living Neurons

Live Neurotransmitter Vesicle Movement Dynamics in Living Neurons

Hélder T. Moreira[1,2], Ivo M. Silva[1], Mónica Sousa[3], Paula Sampaio[3], João P. S. Cunha[1,2]

[1]Faculty of Engineering, University of Porto, Porto, Portugal;

[2] C-BER, INESC-TEC Porto, Portugal;

[3] IBMC, Porto, Portugal;

Abstract

In this paper we present a novel methodology for quantifying neurotransmitter vesicle dynamics by using a combination of image tracking and classification algorithms. We use confocal microscopy videos of living neurons to detect and classify vesicles using support vector machine (SVM), while motion is extracted via global nearest neighbor (GNN) tracking. We present results of the classification algorithm when compared to a ground truth defined by experts.

1. Introduction

In recent years, there has been growing interest in the study of neurotransmitter vesicle dynamics as it is proved the relation between abnormalities in these dynamics and early stages of neurodegenerative diseases (Millecamps and Julien 2013). To date, the characterization of vesicle movement has remained a manual process where the researcher has to individually mark every vesicle on each frame of the video. Such a laborious process becomes impractical once the number of vesicles per frame is usually large. In our work, we leverage recent advancements made in the area of object tracking to be able to automatically detect, classify and extract relevant measures (such as backward and forward velocity and morphology features) from vesicles. Combined in an intuitive GUI, our algorithm provides a unique ability to identify vesicles' path and obtain dynamic measures.

2. Vesicle Tracking Approach

Our approach is based on the combination of three methods:

A. ***Candidate Detection using Adaptive Threshold Value:*** Firstly, we will use adaptive threshold to select a segmentation threshold to separate the candidates to vesicles from the background. The segmentation threshold is selected through an iterative procedure. Let T^i be the segmentation threshold at step *i*. To choose a new segmentation threshold, we apply T^i to the image to separate object and background pixels. Let μ_b and μ_n be the mean gray-level of the object pixels and background pixels after segmentation with T^i. Then the new threshold for step *i+1* is

$$T^{i+1} = \frac{\mu_b + \mu_n}{2} \tag{1}$$

This iterative threshold update procedure is repeated until there is no change in the threshold, i.e., $T^{i+1} = T^i$ (Hu, Hoffman, and Reinhardt 2001).

B. ***Support Vector Machine for Candidate Classification:*** Given a set of labelled (vesicle or non-vesicle) training examples, an SVM training algorithm constructs a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible (Steinwart and Christmann 2008).

C. Global Nearest Neighbor Tracking Algorithm: The GNN algorithm not only associates each vesicle from one with its nearest neighbor from the next frame, but it also relates with information of a defined of number of frames. This method can deal with gaps, which happen when one vesicle that is detected in one frame is not detected in the subsequent one, appearing in a further frame (Banitalebi and Amiri 2008).

3. Results

The images obtained by confocal microscopy are converted to grayscale and an optimal threshold value is applied. Morphology features such as eccentricity, major axis length and area are extracted and candidates are classified by SVM algorithm. The vesicles' path is defined by linking them over time using GNN approach. These steps are illustrated in Figure 1.

Figure 1: Example of available dataset used in our project (in the left); candidates detected by optimal threshold value (in the center); example of one of the detected vesicle paths (in the right).

The SVM classifier was evaluated by comparison with groundtruth manually established by experts. The results are shown in Table 1.

Accuracy	Sensitivity	Specificity	Precision	Computational Time (s/frame)
74%	90%	97%	85%	0.24

Table 1: Classification algorithm evaluation based on some standard parameters.

4. Discussion and Conclusions

Preliminary results indicate that the classification algorithm performs well and with low computational cost, nevertheless the dataset is still small. As this is a work in progress, the tracking algorithm is still to be tested and improved with a larger dataset and by comparing obtained measures with those obtained by manual tracking.

References

Banitalebi, B, and H Amiri. 2008. "An Improved Nearest Neighbor Data Association Method for Underwater Multi-Target Tracking."

Hu, S, E a Hoffman, and J M Reinhardt. 2001. "Automatic Lung Segmentation for Accurate Quantitation of Volumetric X-Ray CT Images." *IEEE Transactions on Medical Imaging* 20 (6): 490–98.

Millecamps, Stéphanie, and Jean-Pierre Julien. 2013. "Axonal Transport Deficits and Neurodegenerative Diseases." *Nature Reviews. Neuroscience* 14 (3). Nature Publishing Group: 161–76.

Steinwart, I, and A Christmann. 2008. *Support Vector Machines*. Information Science and Statistics. Springer.

References

[1] Scott Brady, George Siegel, R Wayne Albers, and Donald Price. *Basic neurochemistry: molecular, cellular and medical aspects*. Academic Press, 2005.

[2] Stéphanie Millecamps and Jean-Pierre Julien. Axonal transport deficits and neurodegenerative diseases. *Nature Reviews Neuroscience*, 14(3):161–176, 2013.

[3] Maja Potokar, Nina Vardjan, Matjaž Stenovec, Mateja Gabrijel, Saša Trkov, Jernej Jorgačevski, Marko Kreft, and Robert Zorec. Astrocytic vesicle mobility in health and disease. *International journal of molecular sciences*, 14(6):11238–11258, 2013.

[4] Ivo Silva. Neurondyn: Live neurotransmitter vesicle movement dynamics in living neurons. 2014.

[5] Isaac Asimov and Anthony Ravielli. *The human brain: Its capacities and functions*. New American Library, 1965.

[6] Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience*, volume 2. Lippincott Williams & Wilkins, 2007.

[7] John T Cacioppo and Gary G Berntson. *Handbook of neuroscience for the behavioral sciences*. John Wiley & Sons, 2009.

[8] Larry R Squire. *Fundamental neuroscience*. Academic Press, 2013.

[9] Lauralee Sherwood. *Human physiology: from cells to systems*. Cengage Learning, 2015.

[10] Greg Stuart, Nelson Spruston, and Michael Häusser. *Dendrites*. Oxford University Press New York, 2008.

[11] Eric R Kandel, James H Schwartz, Thomas M Jessell, et al. *Principles of neural science*, volume 4. McGraw-Hill New York, 2000.

[12] Edward H Koo, Sangram S Sisodia, David R Archer, Lee J Martin, Andreas Weidemann, Konrad Beyreuther, Peter Fischer, Colin L Masters, and Donald L Price. Precursor of amyloid protein in alzheimer disease undergoes fast anterograde axonal transport. *Proceedings of the National Academy of Sciences*, 87(4):1561–1565, 1990.

[13] Eran Perlson, Sandra Maday, Meng-meng Fu, Armen J Moughamian, and Erika LF Holzbaur. Retrograde axonal transport: pathways to cell death? *Trends in neurosciences*, 33(7):335–344, 2010.

[14] Gordon M Shepherd et al. *The synaptic organization of the brain*. Oxford University Press New York, 2004.

[15] 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):391–405, 2006.

[16] James H Schwartz. Neurotransmitters. *eLS*, 2000.

[17] Thomas C Südhof. The synaptic vesicle cycle. *Annu. Rev. Neurosci.*, 27:509–547, 2004.

[18] Andrea Raimondi, Shawn M Ferguson, Xuelin Lou, Moritz Armbruster, Summer Paradise, Silvia Giovedi, Mirko Messa, Nao Kono, Junko Takasaki, Valentina Cappello, et al. Overlapping role of dynamin isoforms in synaptic vesicle endocytosis. *Neuron*, 70(6):1100–1114, 2011.

[19] Nobutaka Hirokawa, Shinsuke Niwa, and Yosuke Tanaka. Molecular motors in neurons: transport mechanisms and roles in brain function, development, and disease. *Neuron*, 68(4):610–638, 2010.

[20] Max A Schlager and Casper C Hoogenraad. Basic mechanisms for recognition and transport of synaptic cargos. *Mol Brain*, 2(1):25, 2009.

[21] Peter J Hollenbeck and William M Saxton. The axonal transport of mitochondria. *Journal of cell science*, 118(23):5411–5419, 2005.

[22] Michael A Welte. Bidirectional transport along microtubules. *Current Biology*, 14(13):R525–R537, 2004.

[23] William O Hancock. Bidirectional cargo transport: moving beyond tug of war. *Nature Reviews Molecular Cell Biology*, 15(9):615–628, 2014.

[24] Nobutaka Hirokawa, Yasuko Noda, Yosuke Tanaka, and Shinsuke Niwa. Kinesin superfamily motor proteins and intracellular transport. *Nature reviews Molecular cell biology*, 10(10):682–696, 2009.

[25] Xinnan Wang and Thomas L Schwarz. Imaging axonal transport of mitochondria. *Methods in enzymology*, 457:319–333, 2009.

[26] Thomas L Schwarz. Mitochondrial trafficking in neurons. *Cold Spring Harbor perspectives in biology*, 5(6):a011304, 2013.

[27] Erica Chevalier-Larsen and Erika LF Holzbaur. Axonal transport and neurodegenerative disease. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1762(11):1094–1108, 2006.

[28] Bernadette H LaMonte, Karen E Wallace, Beth A Holloway, Spencer S Shelly, Jennifer Ascaño, Mariko Tokito, Thomas Van Winkle, David S Howland, and Erika LF Holzbaur. Disruption of dynein/dynactin inhibits axonal transport in motor neurons causing late-onset progressive degeneration. *Neuron*, 34(5):715–727, 2002.

[29] K Schindowski, K Belarbi, and L Buee. Neurotrophic factors in alzheimer's disease: role of axonal transport. *Genes, Brain and Behavior*, 7(s1):43–56, 2008.

[30] Xin-An Liu, Valerio Rizzo, and Sathyanarayanan Puthanveettil. Pathologies of axonal transport in neurodegenerative diseases. *Translational neuroscience*, 3(4):355–372, 2012.

[31] Gorazd B Stokin, Concepción Lillo, Tomás L Falzone, Richard G Brusch, Edward Rockenstein, Stephanie L Mount, Rema Raman, Peter Davies, Eliezer Masliah, David S Williams, et al. Axonopathy and transport deficits early in the pathogenesis of alzheimer's disease. *Science*, 307(5713):1282–1288, 2005.

[32] Nathan P Staff, Eduardo E Benarroch, and Christopher J Klein. Neuronal intracellular transport and neurodegenerative disease. *Neurology*, 76(11):1015–1020, 2011.

[33] Lynsey G Bilsland, Erik Sahai, Gavin Kelly, Matthew Golding, Linda Greensmith, and Giampietro Schiavo. Deficits in axonal transport precede als symptoms in vivo. *Proceedings of the National Academy of Sciences*, 107(47):20523–20528, 2010.

[34] M Borrell-Pages, D Zala, S Humbert, and F Saudou. Huntington's disease: from huntingtin function and dysfunction to therapeutic strategies. *Cellular and Molecular Life Sciences CMLS*, 63(22):2642–2660, 2006.

[35] P Hemachandra Reddy and Ulziibat P Shirendeb. Mutant huntingtin, abnormal mitochondrial dynamics, defective axonal transport of mitochondria, and selective synaptic degeneration in huntington's disease. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1822(2):101–110, 2012.

[36] Yaping Chu, Gerardo A Morfini, Lori B Langhamer, Yinzhen He, Scott T Brady, and Jeffrey H Kordower. Alterations in axonal transport motor proteins in sporadic and experimental parkinson's disease. *Brain*, page aws133, 2012.

[37] Aaron G Filler. The history, development and impact of computed imaging in neurological diagnosis and neurosurgery: Ct, mri, and dti. *Nature Precedings*, 7(1):1–69, 2009.

[38] Frederik Barkhof, Nick C Fox, António J Bastos-Leite, and Philip Scheltens. *Neuroimaging in dementia*. Springer Science & Business Media, 2011.

[39] Amiram Grinvald and Rina Hildesheim. Vsdi: a new era in functional imaging of cortical dynamics. *Nature Reviews Neuroscience*, 5(11):874–885, 2004.

[40] Marvin Minsky. Memoir on inventing the confocal scanning microscope. *Scanning*, 10(4):128–138, 1988.

[41] Denis Semwogerere and Eric R Weeks. Confocal microscopy. *Encyclopedia of Biomaterials and Biomedical Engineering*, pages 1–10, 2005.

[42] Carolyn L Smith. Basic confocal microscopy. *Current Protocols in Neuroscience*, pages 2–2, 2001.

[43] Akihiko Nakano. Spinning-disk confocal microscopy. a cutting-edge tool for imaging of membrane traffic. *Cell structure and function*, 27(5):349–355, 2002.

[44] Ralph Gräf, Jens Rietdorf, and Timo Zimmermann. Live cell spinning disk microscopy. In *Microscopy Techniques*, pages 57–75. Springer, 2005.

[45] Tony Wilson. Spinning-disk microscopy systems. *Cold Spring Harbor Protocols*, 2010(11):pdb–top88, 2010.

[46] Roger Y Tsien. Imagining imaging's future, imaging in cell biology, 2003.

[47] Chandrika Saxena and Deepak Kourav. Noises and image denoising techniques: A brief survey. 2014.

[48] Reza Ahmadi, Javad Kangarani Farahani, Farbod Sotudeh, Ashkan Zhaleh, and Saeid Garshasbi. Survey of image denoising techniques. *Life Science Journal*, 10(1), 2013.

[49] Qiang Wu, Fatima Merchant, and Kenneth Castleman. *Microscope image processing*. Academic press, 2010.

[50] Lin Yin, Ruikang Yang, Moncef Gabbouj, and Yrjo Neuvo. Weighted median filters: a tutorial. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 43(3):157–192, 1996.

[51] Raman Maini and Himanshu Aggarwal. A comprehensive review of image enhancement techniques. *arXiv preprint arXiv:1003.4053*, 2010.

[52] SS Bedi and Rati Khandelwal. Various image enhancement techniques-a critical review. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(3), 2013.

[53] Navneet Kaur and Er Kanwalpreet Singh. Image enhancement techniques: A selected review. In *International Journal of Engineering Research and Technology*, volume 2. ESRSA Publications, 2013.

[54] Bhabatosh Chanda and Dwijesh Dutta Majumder. *Digital image processing and analysis*. PHI Learning Pvt. Ltd., 2004.

[55] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.

[56] Dongju Liu and Jian Yu. Otsu method and k-means. In *Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on*, volume 1, pages 344–349. IEEE, 2009.

[57] Allan Hanbury. Image segmentation by region based and watershed algorithms. *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

[58] HP Narkhede. Review of image segmentation techniques. *International Journal of Science and Modern Engineering (IJISME) ISSN*, pages 2319–6386, 2013.

[59] J Wu, E Lewis, G Ferns, and J Giles. Automated coronary calcium scoring using predictive active contour segmentation. In *Nuclear Science Symposium Conference Record (NSS/MIC), 2009 IEEE*, pages 3970–3974. IEEE, 2009.

[60] Mark Nixon, Mark S Nixon, and Alberto S Aguado. *Feature extraction and image processing for Computer Vision*. Academic Press, 2012.

[61] Mingqiang Yang, Kidiyo Kpalma, and Joseph Ronsin. A survey of shape feature extraction techniques. *Pattern recognition*, pages 43–90, 2008.

[62] Dong ping Tian. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.

[63] Dengsheng Zhang and Guojun Lu. Shape-based image retrieval using generic fourier descriptor. *Signal Processing: Image Communication*, 17(10):825–848, 2002.

[64] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[65] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[66] Magali RG Meireles, Paulo EM Almeida, and Marcelo Godoy Simões. A comprehensive review for industrial applicability of artificial neural networks. *Industrial Electronics, IEEE Transactions on*, 50(3):585–601, 2003.

[67] Jianmin Jiang, P Trundle, and Jinchang Ren. Medical image analysis with artificial neural networks. *Computerized Medical Imaging and Graphics*, 34(8):617–631, 2010.

[68] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.

[69] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

[70] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[71] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[72] Yi Liu and Yuan F Zheng. Fs_sfs: A novel feature selection method for support vector machines. *Pattern recognition*, 39(7):1333–1345, 2006.

[73] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[74] Phil Blunsom. Hidden markov models. *Lecture notes, August*, 15:18–19, 2004.

[75] Erik Meijering, Oleh Dzyubachyk, Ihor Smal, et al. Methods for cell and particle tracking. *Methods Enzymol*, 504(9):183–200, 2012.

[76] Anton Milan. Energy minimization for multiple object tracking. 2013.

[77] Sandeep Kumar Patel and Agya Mishra. Moving object tracking techniques: A critical review. *Indian Journal of Computer Science and Engineering*, 4(2):95–102, 2013.

[78] Bahadır Karasulu. Review and evaluation of well-known methods for moving object detection and tracking in videos. *Journal of aeronautics and space technologies*, 4(4):11–22, 2010.

[79] Duc Phu Chau, François Bremond, and Monique Thonnat. Object tracking in videos: Approaches and issues. *arXiv preprint arXiv:1304.5212*, 2013.

[80] Peter S Maybeck. The kalman filter: An introduction to concepts. In *Autonomous Robot Vehicles*, pages 194–204. Springer, 1990.

[81] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[82] Jurjen HP Broeke, Haifang Ge, Ineke M Dijkstra, Ali Taylan Cemgil, Jürgen A Riedl, L Niels Cornelisse, Ruud F Toonen, Matthijs Verhage, and William J Fitzgerald. Automated quantification of cellular traffic in living cells. *Journal of neuroscience methods*, 178(2):378–384, 2009.

[83] B Banitalebi and H Amiri. An improved nearest neighbor data association method for underwater multi-target tracking.

[84] Benjamin Mourllion, Dominique Gruyer, Cyril Royere, and Sébastien Théroude. Multi-hypotheses tracking algorithm based on the belief theory. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8–pp. IEEE, 2005.

[85] Yoshio Goshima, Tomonobu Hida, and Toshiyuki Gotoh. Computational analysis of axonal transport: a novel assessment of neurotoxicity, neuronal development and functions. *International journal of molecular sciences*, 13(3):3414–3430, 2012.

[86] Robert D Lindsay. *Computer analysis of neuronal structures*. Plenum Publishing Corporation, 1977.

[87] Wylie W Ahmed, Brian J Williams, Aaron M Silver, and Taher A Saif. Measuring nonequilibrium vesicle dynamics in neurons under tension. *Lab on a Chip*, 13(4):570–578, 2013.

[88] C Smochină. Image processing techniques and segmentation evaluation. 2011.

[89] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[90] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[91] Shiying Hu, Eric A Hoffman, and Joseph M Reinhardt. Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images. *Medical Imaging, IEEE Transactions on*, 20(6):490–498, 2001.

[92] Frederico A Carpinteiro, Pedro M Costa, Mario Saenz Espinoza, Ivo M Silva, and Joao PS Cunha. Neurondynamics: A method for neurotransmitter vesicle movement characterization in neurons. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 481–484. IEEE, 2014.